SYLLABUS
OF
BSC COMPUTER SCIENCE

ACADEMIC YEAR 2022-2023

University of Mumbai

No. AAMS(UG)/88 of 2021-22

CIRCULAR:-

Attention of the Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology is invited to this office circular No. UG/18 of 2016-17. dated 27th June, 2016 relating to the revised syllabus as per the (CBSGS) of F.Y:B.Sc. (Computer Science) (Sem. I & II).

They are hereby informed that the recommendations made by the Ad-hoc Board of Studies in Computer Science at its meeting held on 21st June , 2021 and subsequently passed by the Board of Deans at its meeting held on 28th June, 2021 vide item No. 6.38 (R) have been accepted by the Academic Council at its meeting held on 29th June, 2021 vide item No.6.38 (R) and that in accordance therewith, the revised syllabus as per the (CBSGS) for the F.Y.B.Sc. Computer Science (Sem. I & II) has been brought into force with effect from the academic year 2021-22 accordingly. (The same is available on the University's website www.mu.ac.in).

MUMBAI – 400 032
30th September, 2021

(Dr. B.N.Gaikwad)
I/c REGISTRAR

To

The Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology.

A.C/6.38(R) 29/06/2021

****

No. AAMS(UG)/88 -A of 2021-22 MUMBAI-400 032 30th September, 2021

Copy forwarded with Compliments for information to:-

1) The Dean, Faculty of Science & Technology,
2) The Chairman, Ad-hoc Board of Studies in Computer Science,
3) The Director, Board of Examinations and Evaluation,
4) The Director, Board of Students Development,
5) The Co-ordinator, University Computerization Centre,

(Dr. B.N.Gaikwad)
I/c REGISTRAR

Copy to :-

1. The Deputy Registrar, Academic Authorities Meetings and Services (AAMS),
2. The Deputy Registrar, College Affiliations & Development Department (CAD),
3. The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Department (AEM),
4. The Deputy Registrar, Research Administration & Promotion Cell (RAPC),
5. The Deputy Registrar, Executive Authorities Section (EA),
6. The Deputy Registrar, PRO, Fort, (Publication Section),
7. The Deputy Registrar, (Special Cell),
8. The Deputy Registrar, Fort/ Vidyanagari Administration Department (FAD) (VAD), Record Section,
9. The Director, Institute of Distance and Open Learning (IDOL Admin), Vidyanagari,

They are requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to in the above circular and that on separate Action Taken Report will be sent in this connection.

1. P.A to Hon'ble Vice-Chancellor,
2. P.A Pro-Vice-Chancellor,
3. P.A to Registrar,
4. All Deans of all Faculties,
5. P.A to Finance & Account Officers, (F.& A.O),
6. P.A to Director, Board of Examinations and Evaluation,
7. P.A to Director, Innovation, Incubation and Linkages,
8. P.A to Director, Board of Lifelong Learning and Extension (BLLE),
9. The Director, Dept. of Information and Communication Technology (DICT) (CCF & UCC), Vidyanagari,
10. The Director of Board of Student Development,
11. The Director, Department of Students Walfare (DSD),
12. All Deputy Registrar, Examination House,
13. The Deputy Registrars, Finance & Accounts Section,
14. The Assistant Registrar, Administrative sub-Campus Thane,
15. The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan,
16. The Assistant Registrar, Ratnagiri sub-centre, Ratnagiri,
17. The Assistant Registrar, Constituent Colleges Unit,
18. BUCTU,
19. The Receptionist,
20. The Telephone Operator,
21. The Secretary MUASA

for information.

AC – 29/06/2021
Item No: 6.38

UNIVERSITY OF MUMBAI

Syllabus

For the

Program: F.Y.B.Sc. Sem -I &II CBCS

Course: Computer Science

(Choice Based and Credit System with effect from the academic year 2021-22)

AC – 29/06/2021

Item No: 6.38

UNIVERSITY OF MUMBAI

Syllabus for Approval

Sr. No.	Heading	Particulars
1.	Title of the Course	**F.Y.B.Sc. Sem. I & II (Computer Science)**
2.	Eligibility for Admission	Ordinance no. O.5719 Circular no. UG/284 of 2007 dated 16th June 2007
3.	Passing Marks	40%
4.	Ordinances / Regulations (if, any)	As applicable for all B.Sc. Courses
5.	Number of years / Semesters	Three years – Six Semesters
6.	Level	~~P.G.~~/ U.G. /~~Diploma / Certificate~~ (Strike out which is not applicable)
7.	Pattern	~~Yearly /~~ Semester, Choice Based (Strike out which is not applicable)
8.	Status	~~New~~ /Revised
9.	To be implemented from Academic year	From the Academic Year **2021 – 2022**

Date: 28/06/2021

Dr. Jagdish Bakal
BoS Chairperson in Computer Science

Dr. Anuradha Majumdar
Dean, Science and Technology

Preamble

The rise of Information and Communication Technology (ICT) has profoundly affected modern society. Increasing applications of computers in almost all areas of human endeavor has led to vibrant industries with concurrent rapid change in technology.

As the computing field advances at a rapid pace, the students must possess a solid foundation that allows and encourages them to maintain relevant skills as the field evolves. Specific languages and technology platforms change over time. Thus students must continue to learn and adapt their skills throughout their careers. To develop this ability, students will be exposed to multiple programming languages, tools, paradigms and technologies as well as the fundamental underlying principles throughout this programme.

The programme offers required courses such as programming languages, data structures, computer architecture and organization, algorithms, database systems, operating systems, and software engineering; as well as specialized courses in artificial intelligence, computer-based communication networks, distributed computing, information security, graphics, human-computer interaction, multimedia, scientific computing, web technology, and other current topics in computer science.

The core philosophy of this programme is to –

- Form strong foundations of Computer Science
- Nurture programming, analytical & design skills for the real world problems.
- Introduce emerging trends to the students in gradual way.
- Groom the students for the challenges of ICT industry

The students these days not only aspire for a career in the industry but also look for research opportunities. The main aim of this programme is to deliver a modern curriculum that will equip graduates with strong theoretical and practical backgrounds to enable them to excel in the workplace and to be lifelong learners. Not only does it prepare the students for a career in Software industry, it also motivates them towards further studies and research opportunities. Graduating students, can thus take up postgraduate programmes in CS leading to research as well as R&D, can be employable at IT industries, or can adopt a business management career.

In the first year i.e. for semester I & II, basic foundation of important skills required for software development is laid. The syllabus proposes to have four core subjects of Computer science and two core courses of Mathematics-Statistics. All core subjects are proposed to have theory as well as practical tracks. While the Computer Science courses will form fundamental skills for solving computational problems, the Mathematics & Statistics course will inculcate research-oriented acumen. Ability Enhancement Courses on Soft Skill Development will ensure an overall and holistic development of the students. The syllabus design for further semesters encompasses more advanced and specialized courses of Computer Science.

We sincerely believe that any student taking this programme will get very strong foundation and exposure to basics, advanced and emerging trends of the subject. We hope that the students‟ community and teachers‟ fraternity will appreciate the treatment given to the courses in the syllabus.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents; we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly, helped designing certain specialized courses and the syllabus as a whole.

Programme Structure for B.Sc. Computer Science

Programme Duration	06 Semesters *spread across 3 years*
Total Credits required for successful completion of the Course	120
Credits required from the Core Courses	76
Credits required for the Ability Enhancement Courses	04
Credits required for Skills Enhancement Courses	32
Credits for General Elective Courses	08
Minimum Attendance per Semester	75%

Progamme Objectives

The objectives of the 3 year B.Sc. Computer Science programme are as follows:

□ To develop an understanding and knowledge of the basic theory of Computer Science with good foundation on theory, systems and applications.

□ To fosternecessary skills and analytical abilities for developing computer based solutions of real-life problems.

□ To provide training in emergent computing technologies which lead to innovative solutions for industry and academia.

□ To develop the necessary study skills and knowledge to pursue further post-graduate study in computer science or other related fields.

□ To develop the professional skillset required for a career in an information technology oriented business or industry.

□ To enable students to work independently and collaboratively, communicate effectively, and become responsible, competent, confident, insightful, and creative users of computing technology

Progamme Learning Outcomes

At the end of three year Bachelor of Computer Science the students will be able:

- To formulate, to model, to design solutions, procedure and to use software tools to solve real world problems.

- To design and develop computer programs/computer -based systems in the areas such as networking, web design, security, cloud computing, IoT, data science and other emerging technologies.

- To familiarize with the modern-day trends in industry and research based settings and thereby innovate novel solutions to existing problems.

- To apply concepts, principles, and theories relating to computer science to new situations.

- To use current techniques, skills, and tools necessary for computing practice

- To apply standard Software Engineering practices and strategies in real-time software project development

- To pursue higher studies of specialization and to take up technical employment.

- To work independently or collaboratively as an effective tame member on a substantial software project.

- To communicate and present their work effectively and coherently.

- To display ethical code of conduct in usage of Internet and Cyber systems.

- To engage in independent and life-long learning in the background of rapid changing IT industry.

Academic year 2021-2022

Semester – I				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures/Week**
USCS101	Core Subject	Digital Systems & Architecture	2	3
USCSP101	Core Subject Practical	Digital Systems & Architecture – Practical	1	3
USCS102	Core Subject	Introduction to Programming with Python	2	3
USCSP102	Core Subject Practical	Introduction to Programming with Python – Practical	1	3
USCS103	Core Subject	LINUX Operating System	2	3
USCSP103	Core Subject Practical	LINUX Operating System – Practical	1	3
USCS104	Core Subject	Open Source Technologies	2	3
USCSP104	Core Subject Practical	Open Source Technologies – Practical	1	3
USCS105	Core Subject	Discrete Mathematics	2	3
USCSP105	Core Subject Practical	Discrete Mathematics – Practical	1	3
USCS106	Core Subject	Descriptive Statistics	2	3
USCSP106	Core Subject Practical	Descriptive Statistics – Practical	1	3
USCS107	Ability Enhancement Course	Soft Skills	2	3

F.Y.B.Sc. Computer Science Syllabus

Choice Based Credit System (CBCS)

with effect from

Academic year 2021-2022

Semester – II				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures/Week**
USCS201	Core Subject	Design & Analysis of Algorithms	2	3
USCSP201	Core Subject Practical	Design & Analysis of Algorithms – Practical	1	3
USCS202	Core Subject	Advanced Python Programming	2	3
USCSP202	Core Subject Practical	Advanced Python Programming – Practical	1	3
USCS203	Core Subject	Introduction to OOPs using C++	2	3
USCSP203	Core Subject Practical	Introduction to OOPs using C++ – Practical	1	3
USCS204	Core Subject	Database Systems	2	3
USCSP204	Core Subject Practical	Database Systems – Practical	1	3
USCS205	Core Subject	Calculus	2	3
USCSP205	Core Subject Practical	Calculus – Practical	1	3
USCS206	Core Subject	Statistical Methods	2	3
USCSP206	Core Subject Practical	Statistical Methods – Practical	1	3
USCS207	Ability Enhancement Course	E-Commerce & Digital Marketing	2	3

Semester I

Course Code	Course Title	Credits	Lectures /Week
USCS101	**Digital Systems & Architecture**	**2**	**3**

About the Course:
This course introduces the principles of computer organization and the basic architecture concepts.The course emphasizes performance and cost analysis, instruction set design, pipelining, memory technology, memory hierarchy, virtual memory management, and I/O systems.

Course Objectives:

□ To have an understanding of Digital systems and operation of a digital computer.
□ To learn different architectures & organizations of memory systems, processor organization and control unit.
□ To understand the working principles of multiprocessor and parallel organization‟s as advanced computer architectures

Learning Outcomes:
After successful completion of this course, students would be able to

□ To learn about how computer systems work and underlying principles
□ To understand the basics of digital electronics needed for computers
□ To understand the basics of instruction set architecture for reduced and complex instruction sets
□ To understand the basics of processor structure and operation
□ To understand how data is transferred between the processor and I/O devices

Unit	Topics	No of Lectures
I	**Fundamentals of Digital Logic:** Boolean algebra, Logic Gates, Simplification of Logic Circuits: Algebraic Simplification, Karnaugh Maps. Combinational Circuits: Adders, Mux, De-Mux, Sequential Circuits: Flip-Flops (SR, JK & D), Counters: synchronous and asynchronous Counter **Computer System:** Comparison of Computer Organization & Architecture, Computer Components and Functions, Interconnection Structures. Bus Interconnections, Input / Output: I/O Module, Programmed I/O, Interrupt Driven I/O, Direct Memory Access	**15**
II	**Memory System Organization:** Classification and design parameters, Memory Hierarchy, Internal Memory: RAM, SRAM and DRAM, Interleaved and Associative Memory. Cache Memory: Design Principles, Memory mappings, Replacement Algorithms, Cache performance, Cache Coherence. Virtual Memory, External Memory: Magnetic Discs, Optical Memory, Flash Memories, RAID Levels **Processor Organization:** Instruction Formats, Instruction Sets, Addressing Modes, Addressing Modes Examples with Assembly Language [8085/8086 CPU], Processor Organization, Structure and Function. Register	**15**

	Organization, Basic Microprocessor operations: Data Transfer (Register / Memory) Operations, Arithmetic & Logical Operations, Instruction Cycle, Instruction Pipelining. Introduction to RISC and CISC Architecture, Instruction Level Parallelism and Superscalar Processors: Design Issues	
III	**Control Unit:** Micro-Operations, Functional Requirements, Processor Control, Hardwired Implementation, Micro-programmed Control**.** **Fundamentals of Advanced Computer Architecture:** Parallel Architecture: Classification of Parallel Systems, Flynn‟s Taxonomy, Array Processors, Clusters, and NUMA Computers. Multiprocessor Systems: Structure & Interconnection Networks, Multi-Core Computers: Introduction, Organization and Performance.	**15**

Textbooks:

1. M. Mano, Computer System Architecture 3rd edition, Pearson
2. Carl Hamacher et al., Computer Organization and Embedded Systems, 6 ed., McGraw-Hill 2012
3. R P Jain, Modern Digital Electronics, Tata McGraw Hill Education Pvt. Ltd. , 4th Edition, 2010

Additional References:

1. William Stallings (2010), Computer Organization and Architecture- designing for performance,8th edition, Prentice Hall, New Jersy.
2. Anrew S. Tanenbaum (2006), Structured Computer Organization, 5th edition, PearsonEducation Inc,
3. John P. Hayes (1998), Computer Architecture and Organization, 3rd edition, Tata McGrawHill

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP101	**Digital Systems & Architecture – Practical**	**1**	**3**

1	Study and verify the truth table of various logic gates (NOT, AND, OR, NAND, NOR, EX-OR, and EX-NOR).
2	Simplify given Boolean expression and realize it.
3	Design and verify a half/full adder
4	Design and verify half/full subtractor
5	Design a 4 bit magnitude comparator using combinational circuits.
6	Design and verify the operation of flip-flops using logic gates.
7	Verify the operation of a counter.
8	Verify the operation of a 4 bit shift register
9	Design and implement expression using multiplexers / demultiplexers.
10	Design and implement 3-bit binary ripple counter using JK flip flops.
11	Simple microprocessor programs for data transfer operations
12	Simple microprocessor programs for arithmetic & logical transfer operations
Note	Practical 1 – 10 can be performed using any open source simulator (like Logisim) (Download it from https://sourceforge.net/projects/circuit/) Practical 11 – 12 can be performed on any simulation software like Jubin‟s 8085 simulator

Course Code	Course Title	Credits	Lectures /Week
USCS102	**Introduction to Programming with Python**	**2**	**3**

About the Course:
This course is aims at introducing one of the fastest growing programming language of current time and enables learners to understand the fundamentals of programming with Python. Learners will be able to write programs to solve real-world problems, and produce quality code. It will help to develop strong skills of programming for implementing applications for emerging fields including data science and machine learning.

Course Objectives:

- To learn how to design and program Python applications.
- To explore the innards of Python Programming and understand components of Python Program
- To define the structure and components of a Python program.
- To learn how to write loops and decision statements in Python
- To learn about inbuilt input/output operations and compound data types in Python

Learning Outcomes:
After successful completion of this course, students would be able to:

- Ability to store, manipulate and access data in Python
- Ability to implement basic Input / Output operations in Python
- Ability to define the structure and components of a Python program.
- Ability to learn how to write loops and decision statements in Python.
- Ability to learn how to write functions and pass arguments in Python.
- Ability to create and use Compound data types in Python

Unit	Topics	No of Lectures
I	**Overview of Python:** History & Versions, Features of Python, Execution of a Python Program, Flavours of Python, Innards of Python, Python Interpreter, Memory Management in Python, Garbage Collection in Python, Comparison of Python with C and Java, Installing Python, Writing and Executing First Python Program, Getting Help, IDLE **Data Types, Variables and Other Basic Elements:** Comments, Docstrings, Data types- Numeric Data type, Compound Data Type, Boolean Data type, Dictionary, Sets, Mapping, Basic Elements of Python, Variables **Input and Output Operations:** Input Function, Output Statements, The print() function, The print(“string”) function, The print(variables list) function, , The print(object) function, The print(formatted string) function, Command Line Arguments **Control Statements:** The if statement, The if … else Statement, The „if …	**15**

	elif … else‟ Statement, Loop Statement- while loop, for loop, Infinite loop, Nested loop, The else suite, break statement, continue statement, pass statement, assert statement, return statement	
II	**Operators:** Arithmetic operators, Assignment operators, Unary minus operator, Relational operators, Logical operators, Bitwise operators, Membership operators, Identity operators, Precedence of Operators, Associativity of Operators **Arrays:** Creating Arrays, Indexing and Slicing of Arrays, Basic Array Operations, Arrays Processing, Mathematical Operations on Array, Aliasing Arrays, Slicing and Indexing in NumPy Arrays, Basic slicing, Advanced Indexing, Dimensions of Arrays, Attributes of an Array, The ndim Attribute, The shape Attribute, The size Attribute, The itemsize Attribute **Functions:** Function definition and call, Returning Results, Returning Multiple Values from a Function, Built-in Functions, Difference between a Function and a Method, Pass Value by Object Reference, Parameters and Arguments, Formal and Actual Arguments, Positional Arguments, Keyword Arguments, Default Arguments, Arbitrary Arguments, Recursive Functions, Anonymous or Lambda Functions, Using Lambda with the filter() Function, Using Lambda with the map() Function, Using Lambda with the reduce() Function **Modules**:Introduction to Modules in Python	**15**
III	**Strings:** Creating Strings, Functions of Strings, Working with Strings, Length of a String, Indexing and Slicing, Repeating and Concatenating Strings, Checking Membership, Comparing Strings, Removing Spaces, Finding Substrings, Counting Substrings, Immutability, Splitting and Joining Strings, Changing Case, Checking Starting and Ending of a String, Sorting Strings, Searching in the Strings, Testing Methods, Formatting Strings, Finding the Number of Characters and Words, Inserting Substrings into a String **List and Tuples:** Lists, List Functions and Methods, List Operations, List Slices, Nested Lists, Tuples, Functions in Tuple **Dictionaries:** Creating a Dictionary, Operators in Dictionary, Dictionary Methods, Using for Loop with Dictionaries, Operations on Dictionaries, Converting Lists into Dictionary, Converting Strings into Dictionary, Passing Dictionaries to Functions, Sorting the Elements of a Dictionary using Lambda, Ordered Dictionaries	**15**

Textbooks:

1. Practical Programming: An Introduction to Computer Science Using Python 3, Paul Gries , Jennifer Campbell, Jason Montojo, Pragmatic Bookshelf, 2nd Edition, 2014
2. Programming through Python, M. T Savaliya, R. K. Maurya& G M Magar, Sybgen Learning India, 2020

Additional References:

1. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
2. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
3. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018
4. Python Programming: Using Problem Solving Approach, ReemaThareja, Oxford Univeristy Press, 2017
5. Let Us Python, Yashwant. B. Kanetkar, BPB Publication, 2019

Course Code	Course Title	Credits	Lectures /Week
USCSP102	**Introduction to Programming with Python – Practical**	**1**	**3**
1	Write a program to design and develop python program to implement various control statement using suitable examples		
2	Write program in Python to define and call functions for suitable problem.		
3	Write Python program to demonstrate different types of function arguments.		
4	Write a Python program to demonstrate the precedence and associativity of operators.		
5	Write suitable Python program to implement recursion for problems such as Fibonacci series, Factorial, Tower of Hanoi etc.		
6	Write Python program to implement and use lambda function in python		
7	Write a python program to create and manipulate arrays in Python. Also demonstrate use of slicing and indexing for accessing elements from the array.		
8	Write a program to implement list in Python for suitable problem. Demonstrate various operations on it.		
9	Write a program to implement tuple in Python for suitable problem. Demonstrate various operations on it.		
10	Write a program to implement dictionary in Python for suitable problem. Demonstrate various operations on it.		

Course Code	Course Title	Credits	Lectures /Week
USCS103	**LINUX Operating System**	**2**	**3**

About the Course:
This syllabus will help to train students in fundamental skills and build-up sustainable interest in Linux Operating System. It will improve necessary knowledge base to understand Linux Operating System and its practical implementation, it will also help to develop Linux based solutions for real life problems.

Course Objectives:

- To learn basic concepts of Linux in terms of operating system
- To learn use of various shell commands with regular expressions
- To set Linux Environment variables and learn setting file permissions to maintain Linux security implementation
- To learn various editors available in Linux OS
- To learn shell scripting.
- To learn installation of compilers and programming using C and Python languages on Linux platform

Learning Outcomes:
After successful completion of this course, students would be able to

- Work with Linux file system structure, Linux Environment
- Handle shell commands for scripting, with features of regular expressions, redirections
- Implement file security permissions
- Work with vi, sed and awk editors for shell scripting using various control structures
- Install softwares like compilers and develop programs in C and Python programming languages on Linux Platform

Unit	Topics	No of Lectures
I	**Linux operating system and Basics :** History, GNU Info and Utilities, Various Linux Distributions, The Unix/Linux architecture, Features of Unix/Linux, Starting the shell, Shell prompt, Command structure, File Systems and Directory Structure, man pages, more documentation pages **Basic Bash shell commands:** General purpose utility Commands, basic commands, Various file types, attributes and File handling Commands, Handling Ordinary Files. More file attributes **Advanced Bash shell commands:**Simple Filters, Filters using regular expressions. **The Linux environment variable:** Setting, Locating and removing environment variables like PATH etc, Default shell environment variables, Using command aliases.	**15**

II	**Understanding Linux file permission:** Linux security, Using Linux groups, Decoding file permissions, Changing security setting, Sharing files. **Linux Security:** Understanding Linux Security, uses of root, sudo command, working with passwords, Understanding ssh. **Networking:** TCP/IP Basics, TCP/IP Model, Resolving IP addresses, Applications, ping, telnet, ftp, DNS **Working withEditors:** awk, sed and Introduction to vi	15
III	**Basic script building:** Using multiple commands, Creating script files, Displaying messages, Using variables, Redirecting Input and Output, Pipes performing math, Exiting the script. **Using structured commands:** Working with if-then, if-then-else and nested if statements, test command, Compound condition testing, while command, until command, case command. **Script and Process control :** Handling signals, Running scripts in background mode, Running scripts without a console, Job control, Job scheduling commands: ps, nice, renice, at, batch, cron table, Running the script at boot	15

Textbooks:

1. "Linux Command line and Shell Scripting Bible", Richard Blum, Wiley India.
2. "Unix: Concepts and Applications", Sumitabha Das, 4th Edition, McGraw Hill.
3. "Official Ubuntu Book", Matthew Helmke& Elizabeth K. Joseph with Jose Antonio Rey and Philips Ballew, 8th Ed.

Additional References:

1. "Linux Administration: A Beginner's Guide", Fifth Edition, Wale Soyinka, Tata McGraw-Hill, 2008.
2. "Linux: Complete Reference", Richard Petersen, 6th Edition, Tata McGraw-Hill
3. "Beginning Linux Programming", Neil Mathew, 4th Edition, Wiley Publishing, 2008.

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP103	**LINUX Operating System – Practical**	**1**	**3**
1	**Installation of Ubuntu Linux operating system.** a) Booting and Installing from (USB/DVD) b) Using Ubuntu Software center / Using Synaptic c) Explore useful software packages.		
2	**Becoming an Ubuntu power user** a) Administering system and User setting b) Learning Unity keyboard c) Using the Terminal d) Working with windows programs		

3	**File System Commands:** touch, help, man, more, less, pwd, cd, mkdir, rmdir, ls, find, ls, etc **File handling Commands:** cat, cp, rm, mv, more, file, wc, od, cmp, diff, comm, chmod, chown, chgrp, gzip and gunzip, zip and unzip, tar, ln, umask,, chmod, chgrp, chown, etc
4	**General purpose utility Commands:**cal, date, echo, man, printf, passwd, script, who, uname, tty, stty, etc **Simple Filters and I/O redirection:** head, tail, cut paste, sort, grep family, tee, uniq, tr, etc. **Networking Commands:** who, whoami, ping, telnet, ftp, ssh, etc
5	**Editors:** vi, sed, awk
6	**Working and Managing with processes-** sh, ps, kill, nice, at and batch etc.
7	**Shell scripting I:** Defining variables, reading user input, exit and exit status commands, , expr, test, [], if conditional, logical operators
8	**Shell scripting II:** Conditions (for loop, until loop and while loop) arithmetic operations, examples
9	**Shell scripting III:** Redirecting Input / Output in scripts, creating your own Redirection
10	Installation of C/C++/Java/Python Compiler and creating an environment for app development. Basic programming using C and Python Languages.

Course Code	Course Title	Credits	Lectures /Week
USCS104	**Open Source Technologies**	**2**	**3**

About the Course:

Open Source Software is becoming an important resource for development, especially in developing countries. A working understanding of the economic and technical background of the Free / Open Source Software movement (FOSS) is essential for its effective use. The course takes students through the history and current status of the FOSS world, and starts them exploring it, by connecting their personal experiences with corresponding FOSS projects. Students will experience finding and using Open Source Software projects.

Course Objectives:

- Understand the difference between open-source software and commercial software.
- Understand the policies, licensing procedures and ethics of FOSS.
- Understand open-source philosophy, methodology and ecosystem.
- Awareness with Open-Source Technologies.

Learning Outcomes:

- Differentiate between Open Source and Proprietary software and Licensing.
- Recognize the applications, benefits and features of Open-Source Technologies
- Gain knowledge to start, manage open-source projects.

Unit	Topics	No of Lectures
I	**Introduction to Open-Source:** Open Source, Need and Principles of OSS, Open-Source Standards, Requirements for Software, OSS success, Free Software, Examples, Licensing, Free Vs. Proprietary Software, Free Software Vs. Open-Source Software, Public Domain. History of free software, Proprietary Vs Open-Source Licensing Model, use of Open-Source Software, FOSS does not mean no cost. History: BSD, The Free Software Foundation and the GNU Project. **Open-Source Principles and Methodology:** Open-Source History, Open-Source Initiatives, Open Standards Principles, Methodologies, Philosophy, Software freedom, Open-Source Software Development, Licenses, Copyright vs. Copy left, Patents, Zero marginal cost, Income-generation Opportunities, Internationalization. **Licensing:** What Is A License, How to create your own Licenses, Important FOSS Licenses (Apache, BSD, PL, LGPL), copyrights and copy lefts, Patent.	15
II	**Open-Source projects:** Starting and maintaining own Open-Source Project, Open-Source Hardware, Open-Source Design, Open-source Teaching, Open-source media. **Collaboration:** Community and Communication, Contributing to Open-Source Projects Introduction to GitHub, interacting with the community on GitHub, Communication and etiquette, testing open-source code, reporting	15

	issues, contributing code. Introduction to Wikipedia, contributing to Wikipedia or contributing to any prominent open-source project of student"s choice. **Open-Source Ethics and Social Impact:** Open source vs. closed source, Open-source Government, Ethics of Open-source, Social and Financial impacts of open-source technology, Shared software, Shared source, Open Source as a Business Strategy	
III	**Understanding Open-Source Ecosystem**: Open-Source Operating Systems: GNU/Linux, Android, Free BSD, Open Solaris. Open-Source Hardware, Virtualization Technologies, Containerization Technologies: Docker, Development tools, IDEs, Debuggers, Programming languages, LAMP, Open-Source Database technologies **Case Studies:** Example Projects: Apache Web server, BSD, GNU/Linux, Android, Mozilla (Firefox), Wikipedia, Drupal, WordPress, Git, GCC, GDB, GitHub, Open Office, LibreOffice Study: Understanding the developmental models, licensing, mode of funding, commercial/non-commercial use.	**15**

Textbooks:

1. "Open-Source Technology", Kailash Vadera&Bhavyesh Gandhi, University Science Press, Laxmi Publications, 2009
2. "Open-Source Technology and Policy", Fadi P. Deek and James A. M. McHugh, Cambridge University Press, 2008.

Additional References:

1. "Perspectives on Free and Open-Source Software", Clay Shirky and Michael Cusumano, MIT press.
2. "Understanding Open Source and Free Software Licensing", Andrew M. St. Laurent, O"Reilly Media.
3. "Open Source for the Enterprise", Dan Woods, GautamGuliani, O"Reilly Media
4. Linux kernel Home: http://kernel.org4
5. Open-Source Initiative: https://opensource.org/5
6. The Linux Foundation: http://www.linuxfoundation.org/
7. The Linux Documentation Project: http://www.tldp.org/2
8. Docker Project Home: http://www.docker.com3.
9. Linux Documentation Project: http://www.tldp.org/6
10. Wikipedia: https://en.wikipedia.org/7.https://en.wikipedia.org/wiki/Wikipedia:Contributing_to_Wikipedia8
11. GitHub: https://help.github.com/9.
12. The Linux Foundation: http://www.linuxfoundation.org/

Course Code	Course Title	Credits	Lectures /Week
USCSP104	**Open Source Technologies– Practical**	**1**	**3**
1	**Open Source Operating Systems** □ Learn the following open source operating system of your choice: Linux, Android, FreeBSD, Open Solaris etc. □ Learn the installation. □ Identify the unique features of these OS.		
2	Hands on with **LibreOffice** □ Learn it from practical view-point □ Give a brief presentation about it to the class		
3	Hands on with **GIMP** Photo Editing Tool □ Learn it from practical view-point □ Give a brief presentation about it to the class		
4	Hands on with **Shotcut** Video Editing Tool □ Learn it from practical view-point □ Give a brief presentation about it to the class		
5	Hands on with **Blender** Graphics and Animation Tool □ Learn it from practical view-point □ Give a brief presentation about it to the class		
6	Hands on with **Apache** Web Server □ Learn it from practical view-point □ Give a brief presentation about it to the class		
7	Hands on with **WordPress** CMS □ Learn it from practical view-point □ Give a brief presentation about it to the class		
8	Contributing to **Wikipedia**: □ Introduction to wikipedia: operating model, license, how to contribute? □ Create your user account on wikipedia □ c. Identify any topic of your choice and contribute the missing information		
9	**Github** □ Create and publish your own open source project: Write any simple program using your choice of programming language. □ Create a repository on github and save versions of your project. You‟ll learn about the staging area, committing your code, branching, and merging, □ Using GitHub to Collaborate: Get practice using GitHub or other remote repositories to share your changes with others and collaborate on multi-developer projects. You‟ll learn how to make and review a pull request on GitHub. □ d. Contribute to a Live Project: Students will publish a repository containing their reflections from the course and submit a pull request.		

10	**Virtualization:** Open Source virtualization technologies: □ Install and configure the following: VirtualBox, Zen, KVM □ Create and use virtual machines
11	**Containerization:** □ Install and configure the following containerization technologies: docker, rocket, LXD □ Create and use containers using it

Course Code	Course Title	Credits	Lectures /Week
USCS105	**Discrete Mathematics**	**2**	**3**

About the Course:
Discrete Mathematics provides an essential foundation for virtually every area of Computer Science. The problem-solving techniques honed in Discrete Mathematics are necessary for writing complicated software. Discrete mathematics also builds the gateway to advanced courses in Mathematical Sciences, Data Science, Machine Learning, Software Engineering, etc.

Course Objectives:

- The purpose of the course is to familiarize the prospective learners with mathematical structures that are fundamentally discrete.
- This course will enhance prospective learners to reason and ability to articulate mathematical problems.
- This course will introduce functions, forming and solving recurrence relations and different counting principles. These concepts will be useful to study or describe objects or problems in computer algorithms and programming languages and these concepts can be used effectively in other courses.

Learning Outcomes:
After successful completion of this course, learners would be able to:

- Define mathematical structures (relations, functions, graphs) and use them to model real life situations.
- Understand, construct and solve simple mathematical problems.
- Solve puzzles based on counting principles.
- Provide basic knowledge about models of automata theory and the corresponding formal languages.
- Develop an attitude to solve problems based on graphs and trees, which are widely used in software.

Unit	Topics	No of Lectures
I	**Functions:** Definition of function; Domain, co-domain, range of a function; Examples of standard functions such as identity and constant functions, absolute value function, logarithmic and exponential functions, flooring and ceiling functions; Injective, surjective and bijective functions; Composite and inverse functions. **Relations:** Definition and examples of relation; Properties of relations, Representation of relations using diagraphs and matrices; Equivalence relation; Partial Order relation, Hasse Diagrams, maximal, minimal, greatest, least element, Lattices.	**15**

	Recurrence Relations: Definition and Formulation of recurrence relations; Solution of a recurrence relation; Solving recurrence relations- Back tracking method, Linear homogeneous recurrence relations with constant coefficients; Homogeneous solution of linear homogeneous recurrence relation with constant coefficients; Particular solution of non-linear homogeneous recurrence relation with constant coefficients; General solution of non-linear homogeneous recurrence relation with constant coefficients; Applications- Formulate and solve recurrence relation for Fibonacci numbers, Tower of Hanoi, Intersection of lines in a plane, Sorting Algorithms.	
II	**Counting Principles:** Basic Counting Principles (Sum and Product Rule); Pigeonhole Principle (without proof) - Simple examples; Inclusion Exclusion Principle (Sieve formula) (without proof); Counting using Tree diagrams. **Permutations and Combinations:** Permutation without and with repetition; Combination without and with repetition; Binomial numbers and identities: Pascal Identity, Vandermonde‟s Identity, Pascal triangle, Binomial theorem (without proof) and applications; Multionomial numbers, Multinomial theorem (without proof) and applications. **Languages, Grammars and Machines:** Languages and Grammars – Introduction, Phase structure grammar, Types of grammar, derivation trees; Finite-State Machines with Output; Finite-State Machines with No Output; Regular Expression and Regular Language.	**15**
III	**Graphs:** Graphs and Graph Models; Graph terminologies and Special types of graphs; Definition and elementary results; Representing graphs, Linked representation of a graph; Graph Isomorphism; Connectivity in graphs – path, trail, walk; Euler and Hamilton paths; Planar graphs, Graph coloring and chromatic number. **Trees:** Definition, Tree terminologies and elementary results; Linked representation of binary trees; Ordered rooted tree, Binary trees, Complete and extended binary trees, Expression trees, Binary Search tree, Algorithms for searching and inserting in binary search trees, Algorithms for deleting in a binary search tree; Traversing binary trees	**15**

Textbooks:

1. Discrete Mathematics and Its Applications, Seventh Edition by Kenneth H. Rosen, McGraw Hill Education (India) Private Limited. (2011)
2. Discrete Mathematics: SemyourLipschutz, Marc Lipson, Schaum‟s out lines, McGraw- Hill Inc.

3rd Edition

3. Data Structures Seymour Lipschutz, Schaum"s out lines, McGraw- Hill Inc. 2017
4. Norman L. Biggs, Discrete Mathematics, Revised Edition, Clarendon Press, Oxford 1989.

Additional References:

1. Elements of Discrete Mathematics: C.L. Liu, Tata McGraw- Hill Edition.
2. Concrete Mathematics (Foundation for Computer Science): Graham, Knuth, Patashnik Second Edition, Pearson Education.
3. Discrete Mathematics: SemyourLipschutz, Marc Lipson, Schaum"s out lines, McGraw- Hill Inc.
4. Foundations in Discrete Mathematics: K.D. Joshi, New Age Publication, New Delhi.

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP105	**Discrete Mathematics – Practical**	**1**	**3**
1	**Functions –** a. Identify if the given mapping is a function b. Finding domain and range of a given function c. Check if the given function is injective/surjective/bijective d. Find the inverse of a given function e. Operations on functions f. Graphs of functions using any online tool		
2	**Relations –** a. Representation of relations b. Determine if the given relation satisfies equivalence relation/partial order relation c. Draw Hasse diagrams d. Find maximal, minimal, greatest, least element in a poset e. Determine if a given poset is a lattice		
3	**Recurrence Relation –** a. Solve recurrence relation using backtracking method b. Solve linear homogeneous recurrence relations with constant coefficients c. Find homogeneous, particular, general solution of a recurrence relation d. Formulate and solving recurrence relation		
4	**Counting Principles –** a. Sum and product rule b. Pigeonhole Principle c. Inclusion Exclusion Principle d. Counting using Tree diagrams		
5	**Permutations and Combinations –** a. Permutations b. Permutations with repetitions c. Combinations d. Combinations with repetitions e. Binomial numbers and Identities		

	f. Applications on Binomial theorem g. Applications on Multinomial theorem
6	**Languages and Grammars –** a. Find the language generated by given grammar b. Check if a given string belongs or not to a given language/grammar c. Operations on languages d. Identify the type of grammar
7	**Finite State Machines –** a. Check if a given string is accepted or rejected by FSM without output b. Find the output for a FSM with output c. Describe a machine (diagram/table)
8	**Regular Expression and Regular Language –** a. Describe the regular expressions represented by given language b. Describe the language represented by given regular expression
9	**Graphs –** a. Types of graph b. Properties of graph c. Representation of graph d. Graph Isomorphism e. Connectivity in graphs – path, trail, walk f. Euler and Hamilton graphs g. Planar graphs h. Graph coloring and chromatic number
10	**Trees –** a. Tree terminologies b. Types of tree c. Properties of tree d. Representation of tree e. Expression tree f. Binary Search tree g. Tree traversal

Course Code	Course Title	Credits	Lectures /Week
USCS106	**Descriptive Statistics**	**2**	**3**

About the Course:
This course is designed to provide learners with an understanding of the data and to develop an understanding of the quantitative techniques from Statistics. It also provides the knowledge of different statistical tools used for primary statistical analysis of data.

Course Objectives:

1. To develop the learners ability to deal with different types of data.
2. To enable the use of different measures of central tendency and dispersion wherever relevant.
3. To make learner aware about the techniques to check the Skewness and Kurtosis of data.
4. To make learner enable to find the correlation between different variables and further apply the regression analysis to find the exact relation between them.
5. To develop ability to analyze statistical data through R software.

Learning Outcomes:
After successful completion of this course, learners would be able to

1. Organize, manage and present data.
2. Analyze Statistical data using measures of central tendency and dispersion.
3. Analyze Statistical data using basics techniques of R.
4. Study the relationship between variables using techniques of correlation and regression.

Unit	Topics	No of Lectures
I	**Data Types and Data Presentation: Data types:** Attribute, Variable, Discrete and Continuous variable, Univariate and Bivariate distribution. Types of Characteristics, Different types of scales: nominal, ordinal, interval and ratio. **Data presentation:** Frequency distribution, Histogram, Ogive curves. **Introduction to R:** Data input, Arithmetic Operators, Vector Operations, Matrix Operations, Data Frames, Built-in Functions. Frequency Distribution, Grouped Frequency Distribution, Diagrams and Graphs, Summary statistics for raw data and grouped frequency distribution. **Measures of Central tendency:** Concept of average/central tendency, characteristics of good measure of central tendency. Arithmetic Mean (A.M.), Median, Mode - Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, merits and demerits. Combined arithmetic mean. Partition Values: Quartiles, Deciles and Percentiles - examples for ungrouped and grouped data	**15**

II	**Measures dispersion:** Concept of dispersion, Absolute and Relative measure of dispersion, characteristics of good measure of dispersion. Range, Semi-interquartile range, Quartile deviation, Standard deviation - Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, merits and demerits. Combined standard deviation, Variance. Coefficient of range, Coefficient of quartile deviation and Coefficient of variation (C.V.) **Moments:** Concept of Moments, Raw moments, Central moments, Relation between raw and central moments. **Measures of Skewness and Kurtosis:** Concept of Skewness and Kurtosis, measures based on moments, quartiles.	15
III	**Correlation:** Concept of correlation, Types and interpretation, Measure of Correlation: Scatter diagram and interpretation; Karl Pearson‟s coefficient of correlation (r): Definition, examples for ungrouped and grouped data, effect of shift of origin and change of scale, properties; Spearman‟s rank correlation coefficient: Definition, examples of with and without repetition. Concept of Multiple correlation. **Regression:** Concept of dependent (response) and independent (predictor) variables, concept of regression, Types and prediction, difference between correlation and regression, Relation between correlation and regression. Linear Regression - Definition, examples using least square method and regression coefficient, coefficient of determination, properties. Concept of Multiple regression and Logistic regression.	15

Textbooks:

1. Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, The World Press Pvt. Ltd., Calcutta.
2. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi

Additional References:

1. Sarma, K. V. S. (2001). Statistics Made it Simple: Do it yourself on PC. Prentce Hall of India, NewDelhi.
2. Agarwal, B. L. (2003). Programmed Statistics, Second Edition, New Age International Publishers, NewDelhi.
3. Purohit, S. G., Gore S. D., Deshmukh S. R. (2008). Statistics Using R, Narosa Publishing House, NewDelhi.
4. Schaum‟s Outline Of Theory And Problems Of Beginning Statistics, Larry J. Stephens, Schaum‟s Outline Series Mcgraw-Hill

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP106	**Descriptive Statistics – Practical**	**1**	**3**
Problem solving and implementation using R programming			
1	**Basics of R-** a. Data input, Arithmetic Operators b. Vector Operations, Matrix Operations c. Data Frames, Built-in Functions d. Frequency Distribution, Grouped Frequency Distribution e. Diagrams and Graphs		
2	**Frequency distribution and data presentation-** a. Frequency Distribution (Univariate data/ Bivariate data) b. Diagrams c. Graphs		
3	**Measures of Central Tendency-** a. Arithmetic Mean b. Median c. Mode d. Partition Values		
4	**Measures dispersion-** a. Range and Coefficient of range b. Quartile deviation and Coefficient of quartile deviation c. Standard deviation, Variance and Coefficient of variation (C.V.)		
5	**Moments-** a. Raw moments b. Central moments		
6	**Measures of Skewness -** a. Karl Pearson‟s measure of Skewness b. Bowley‟s measure of Skewness c. Moment coefficient of Skewness		
7	**Measures of Kurtosis-** a. Moment coefficient of Kurtosis (Absolute measure) b. Moment coefficient of Kurtosis (Relative measure)		
8	**Correlation-** a. Karl Pearson‟s correlation coefficient b. Spearman‟s Rank correlation		
9	**Regression-** a. Method of least squares b. Using regression coefficients c. Properties of regression lines & regression coefficients		
10	**Summary Statistics using R-** a. Summary statistics for raw data b. Summary statistics for grouped frequency distribution c. Simple Correlation & Regression using R		

Course Code	Course Title	Credits	Lectures /Week
USCS107	**Soft Skills**	**2**	**3**

About the Course:
To help learners develop their soft skills and develop their personality along with technical skills. Focus on various communication enhancement along with academic and professional ethics.

Course Objectives:

- Understand the significance and essence of a wide range of soft skills.
- Learn how to apply soft skills in a wide range of routine social and professional settings
- Learn how to employ soft skills to improve interpersonal relationships
- Learn how to employ soft skills to enhance employability and ensure workplace and career success

Learning Outcomes:

- Learners will be able to understand the importance and types soft skills
- Learners will develop skills for Academic and Professional Presentations.
- Learners will able to understand Leadership Qualities and Ethics.
- Ability to understand the importance of stress management in their academic & professional life.

Unit	Topics	No of Lectures
I	**Introduction to Soft Skills** **Soft Skills: An Introduction –** Definition and Significance of Soft Skills; Process, Importance and Measurement of Soft Skill Development. **Personality Development:** Knowing Yourself, Positive Thinking, Johari‟s Window, Physical Fitness **Emotional Intelligence:** Meaning and Definition, Need for Emotional Intelligence, Intelligence Quotient versus Emotional Intelligence Quotient, Components of Emotional Intelligence, Competencies of Emotional Intelligence, Skills to Develop Emotional Intelligence **Positivity and Motivation:** Developing Positive Thinking and Attitude; Driving out Negativity; Meaning and Theories of Motivation; Enhancing Motivation Levels **Etiquette and Mannerism:** Introduction, Professional Etiquette, Technology Etiquette **Ethical Values:** Ethics and Society, Theories of Ethics, Correlation between Values and Behavior, Nurturing Ethics, Importance of Work Ethics, Problems in the Absence of Work Ethics	**15**

II	**Basic Skills in Communication:** **Components of effective communication:** Communication process and handling them, Composing effective messages, Non – Verbal Communication: its importance and nuances: Facial Expression, Posture, Gesture, Eye contact, appearance (dress code). **Communication Skills:** Spoken English, Phonetics, Accent, Intonation **Employment Communication:** Introduction, Resume, Curriculum Vitae, Scannable Resume, Developing an Impressive Resume, Formats of Resume, Job Application or Cover Letter **Job Interviews:** Introduction, Importance of Resume, Definition of Interview, Background Information, Types of Interviews, Preparatory Steps for Job Interviews, Interview Skill Tips, Changes in the Interview Process, FAQ During Interviews **Group Discussion:** Introduction, Ambience/Seating Arrangement for Group Discussion, Importance of Group Discussions, Difference between Group Discussion, Panel Discussion and Debate, Traits, Types of Group Discussions, topic based and Case based Group Discussion, Individual Traits	**15**
III	**Academic and Professional Skills:** **Professional Presentation:** Nature of Oral Presentation, planning a Presentation, Preparing the Presentation, Delivering the Presentation **Creativity at Workplace:** Introduction, Current Workplaces, Creativity, Motivation, Nurturing Hobbies at Work, The Six Thinking Hat Method. **Capacity Building: Learn, Unlearn and Relearn**: Capacity Building, Elements of Capacity Building, Zones of Learning, Ideas for Learning, Strategies for Capacity Building **Leadership and Team Building:** Leader and Leadership, Leadership Traits, Culture and Leadership, Leadership Styles and Trends, Team Building, Types of Teams. **Decision Making and Negotiation:** Introduction to Decision Making, Steps for Decision Making, Decision Making Techniques, Negotiation Fundamentals, Negotiation Styles, Major Negotiation Concepts **Stress and Time Management:** Stress, Sources of Stress, Ways to Cope with Stress	**15**

Textbooks:

1. Managing Soft Skills for Personality Development – edited by B.N.Ghosh, McGraw Hill India, 2017.
2. Soft Skills: An Integrated Approach to Maximize Personality, Gajendra S. Chauhan, Sangeeta Sharma, Wiley India

Additional References:

1. Personality Development and Soft Skills, Barun K. Mitra, Oxford Press
2. Business Communication, ShaliniKalia, Shailja Agrawal, Wiley India
3. Cornerstone: Developing Soft Skills, Sherfield, Pearson India

Semester II

Course Code	Course Title	Credits	Lectures /Week
USCS201	**Design & Analysis of Algorithms**	**2**	**3**

About the Course:
The course covers the concepts of - (i) calculating complexity of algorithms, (ii) the essential operations like searching, sorting, selection, pattern matching & recursion, and (iii) various algorithmic strategies like greedy, divide-n-conquer, dynamic programming, backtracking and implementations of all these on basic data structures like array, list and stack.

Course Objectives:
The objectives of this course are:

- To make students understand the basic principles of algorithm design
- To give idea to students about the theoretical background of the basic data structures
- To familiarize the students with fundamental problem-solving strategies like searching, sorting, selection, recursion and help them to evaluate efficiencies of various algorithms.
- To teach students the important algorithm design paradigms and how they can be used to solve various real world problems.

Learning Outcomes:
After successful completion of this course, students would be able to

- Students should be able to understand and evaluate efficiency of the programs that they write based on performance of the algorithms used.
- Students should be able to appreciate the use of various data structures as per need
- To select, decide and apply appropriate design principle by understanding the requirements of any real life problems

Unit	Topics	No of Lectures
I	**Introduction to algorithms** - What is algorithm, analysis of algorithm, Types of complexity, Running time analysis, How to Compare Algorithms, Rate of Growth, Types of Analysis, Asymptotic Notation, Big-O Notation, Omega-Ω Notation, Theta-Θ Notation, Asymptotic Analysis, Performance characteristics of algorithms, Estimating running time / number of steps of executions on paper, Idea of Computability **Introduction to Data Structures -** What is data structure, types, Introduction to Array(1-d & 2-d), Stack and List data structures, operations on these data structures, advantages disadvantages and applications of these data structures like solving linear equations, Polynomial Representation, Infix-to-Postfix conversion	**15**
II	**Recursion -** What is recursion, Recursion vs Iteration, recursion applications like Factorial of a number, Fibonacci series & their	**15**

	comparative analysis with respect to iterative version, Tower of hanoi problem **Basic Sorting Techniques -** Bubble, Selection and Insertion Sort & their comparative analysis **Searching Techniques -** Linear Search and its types, Binary Search and their comparative analysis **Selection Techniques -** Selection by Sorting, Partition-based Selection Algorithm, Finding the Kth Smallest Elements in Sorted Order & their comparative analysis **String Algorithms** - Pattern matching in strings, Brute Force Method & their comparative analysis	
III	**Algorithm Design Techniques -** Introduction to various types of classifications/design criteria and design techniques **Greedy Technique -** Concept, Advantages & Disadvantages, Applications, Implementation using problems like - file merging problem **Divide-n-Conquer -** Concept, Advantages & Disadvantages, Applications, Implementation using problems like - merge sort, Strassen's Matrix Multiplication **Dynamic Programming** - Concept, Advantages & Disadvantages, Applications, Implementation using problems like - Fibonacci series, Factorial of a number, Longest Common subsequence **Backtracking Programming -** Concept, Advantages & Disadvantages, Applications, Implementation using problems like N-Queen Problem	**15**

Textbooks:

1. “Data Structure and AlgorithmUsing Python”, Rance D. Necaise, Wiley India Edition, 2016.
2. “Data Structures and Algorithms Made Easy”, NarasimhaKarumanchi, CareerMonk Publications, 2016.
3. “Introduction to Algorithms”, Thomas H. Cormen, 3rd Edition, PHI.

Additional References:

1. “Introduction to the Design and Analysis of Algorithms”, Anany Levitin, Pearson, 3rd Edition, 2011.
2. “Design and Analysis of Algorithms”, S. Sridhar, Oxford University Press, 2014.

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP201	**Design & Analysis of Algorithms – Practical**	**1**	**3**
1	Programs on 1-d arrays like - sum of elements of array, searching an element in array, finding minimum and maximum element in array, count the number of even and odd numbers in array. For all such programs, also find the time complexity, compare if there are multiple methods		
2	Programs on 2-d arrays like row-sum, column-sum, sum of diagonal elements, addition of two matrices , multiplication of two matrices. For all such programs, also find the time complexity, compare if there are multiple methods		
3	Program to create a list-based stack and perform various stack operations.		
4	Program to perform linear search and binary search on list of elements. Compare the algorithms by calculating time required in milliseconds using readymade libraries.		
5	Programs to sort elements of list by using various algorithms like bubble, selection sort, and insertion sort. Compare the efficiency of algorithms.		
6	Programs to select the Nth Max/Min element in a list by using various algorithms. Compare the efficiency of algorithms.		
7	Programs to find a pattern in a given string - general way and brute force technique. Compare the efficiency of algorithms.		
8	Programs on recursion like factorial, fibonacci, tower of hanoi. Compare algorithms to find factorial/fibonacci using iterative and recursive approaches.		
9	Program to implement file merging, coin change problems using Greedy Algorithm and to understand time complexity.		
10	Program to implement merge sort, Straseen‟s Matrix Multiplication using D-n-C Algorithm and to understand time complexity.		
11	Program to implement fibonacci series, Longest Common Subsequence using dynamic programming and to understand time complexity. Compare it with the general recursive algorithm.		
12	Program to implement N-Queen Problem, Binary String generation using Backtracking Strategy and to understand time complexity.		

Course Code	Course Title	Credits	Lectures /Week
USCS202	**Advanced Python Programming**	**2**	**3**

About the Course:
This course aims to explore and enable learners to master the skills of advanced topics in Python Programming. It helps learners develops advanced skills such as working with databases, matching patterns, implementing threads and exception handling and GUI in Python. It also highlights and why Python is a useful scripting language for all developers.

Course Objectives:

- To learn how to design object-oriented programs with Python classes.
- To learn about reading, writing and implementing other operation on files in Python.
- To implement threading concept and multithreading on Python
- To design GUI Programs and implement database interaction using Python.
- To know about use of regular expression and handling exceptions for writing robust python programs.

Learning Outcomes:
After successful completion of this course, students would be able to

- Ability to implement OOP concepts in Python including Inheritance and Polymorphism
- Ability to work with files and perform operations on it using Python.
- Ability to implement regular expression and concept of threads for developing efficient program
- Ability to implement exception handling in Python applications for error handling.
- Knowledge of working with databases, designing GUI in Python and implement networking in Python

Unit	Topics	No of Lectures
I	**Working with files:** Files, opening and closing a file, working with text files containing strings, knowing whether a file exists or not, working with binary files, the „with‟ statement, the seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories, running other programs from python program **Regular expressions:** What is a regular expression?, sequence characters in regular expressions, quantifiers in regular expressions, special characters in regular expressions, using regular expression on files, retrieving information from an html file, **Threads in python:** Difference between process and thread, types of threads, benefits of threads, creating threads, single tasking and multitasking, thread synchronization, deadlock in threads, daemon threads **Date and time in python:** Date and time now, combining date and time,	**15**

	formatting dates and times, finding durations using "time delta", comparing two dates, sorting dates, stopping execution temporarily, knowing the time taken by a program, calendar module	
II	**Database in python:** Using SQL with python, retrieving rows from a table, inserting rows into a table, deleting rows from a table, updating rows in a table, creating database tables through python, Exception handling in databases. **Exceptions in python:** Errors in a python program, compile & run-time errors, logical error, exceptions-exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions, logging the exceptions **Networking:** Protocols,server-client architecture, tcp/ip and udp communication **Graphical user interface:** Creating a GUI in python, Widget classes, Working with Fonts and Colours, working with Frames, Layout manager, Event handling	**15**
III	**OOPs in python:** Features of Object Oriented Programming system (oops)-classes and objects, encapsulation, abstraction, inheritance, polymorphism, constructors and destructors **Classes and objects:** Creating a class, the self-variable, types of variables, namespaces, types of methods, instance methods, class methods, static methods, passing members of one class to another class, inner classes **Inheritance and polymorphism:** Inheritance in python, types of inheritance- single inheritance, multilevel inheritance, hierarchical inheritance, multiple inheritance, constructors in inheritance, overriding super class constructors and methods, the super() method, method resolution order (mro), polymorphism, duck typing, operator overloading, method overloading, method overriding, **Abstract classes and interfaces:** Abstract class, abstract method,interfaces in python, abstract classes vs. Interfaces	**15**

Textbooks:

1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 3rd Edition, 2018
2. Programming through Python, M. T Savaliya, R. K. Maurya, G M Magar, Revised Edition, Sybgen Learning India, 2020

Additional References:

1. Advanced Python Programming, Dr. Gabriele Lanaro, Quan Nguyen, SakisKasampalis, Packt Publishing, 2019
2. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018
3. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018
4. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017
5. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP202	**Advanced Python Programming – Practical**	**1**	**3**
1	Write a program to Python program to implement various file operations.		
2	Write a program to Python program to demonstrate use of regular expression for suitable application.		
3	Write a Program to demonstrate concept of threading and multitasking in Python.		
4	Write a Python Program to work with databases in Python to perform operations such as a. Connecting to database b. Creating and dropping tables c. Inserting and updating into tables.		
5	Write a Python Program to demonstrate different types of exception handing.		
6	Write a GUI Program in Python to design application that demonstrates a. Different fonts and colors b. Different Layout Managers c. Event Handling		
7	Write Python Program to create application which uses date and time in Python.		
8	Write a Python program to create server-client and exchange basic information		
9	Write a program to Python program to implement concepts of OOP such as a. Types of Methods b. Inheritance c. Polymorphism		
10	Write a program to Python program to implement concepts of OOP such as a. Abstract methods and classes b. Interfaces		

Course Code	Course Title	Credits	Lectures /Week
USCS203	**Introduction to OOPs using C++**	**2**	**3**

About the Course:

The course aims to introduce a new programming paradigm called Object Oriented Programming. This will be covered using C++ programming language. C++ is a versatile programming language, which supports a variety of programming styles, including procedural, object-oriented, and functional programming. This makes C++ powerful as well as flexible. It can be used to develop software such as operating systems, databases, and compilers.

Course Objectives:

Learning Outcomes:

After successful completion of this course, students would be able to

- Work with numeric, character and textual data and arrays.
- Understand the importance of OOP approach over procedural language.
- Understand how to model classes and relationships using UML.
- Apply the concepts of OOPS like encapsulation, inheritance and polymorphism.
- Handle basic file operations.

Unit	Topics	No of Lectures
I	**Introduction to Programming Concepts:** Object oriented programming paradigm, basic concepts of object oriented programming, benefits of object oriented programming, object oriented languages, applications of object oriented programming. Tokens-keywords, identifiers, constants-integer, real, character and string constants, backslash constants, features of C++ and its basic structure, simple C++ program without class, compiling and running C++ program. **Data Types, Data Input Output and Operators:** Basic data types, variables, rules for naming variables, programming constants, the type cast operator, implicit and explicit type casting, cout and cin statements, operators, precedence of operators. **Decision Making, Loops, Arrays and Strings:** Conditional statements-if, if...else, switch loops- while, do...while, for, types of arrays and string and string manipulations **Unified Modeling Language (UML):** Introduction to UML & class diagrams. **Classes, Abstraction & Encapsulation:** Classes and objects, Dot Operator, data members, member functions, passing data to functions, scope and visibility of variables in function.	**15**

II	**Constructors and Destructors:** Default constructor, parameterized constructor, copy constructor, private constructor, destructors. **Working with objects:**Accessor - mutator methods, static data and static function, access specifiers, array of objects. **Polymorphism -** Binding-static binding & overloading, constructor overloading function overloading, operator overloading, overloading unary and binary operators. **Modelling Relationships in Class Diagrams:** Association, Aggregation-Composition and examples covering these principles	**15**
III	**Inheritance:** Defining base class and its derived class, access specifiers, types of inheritance-single, multiple, hierarchical, multilevel, hybrid inheritance, friend function and friend class, constructors in derived classes. **Modelling Relationships**: Generalization-Specialization and examples covering these principles **Run time Polymorphism -** Dynamic Binding, Function overriding, virtual function, pure virtual function, virtual base class, abstract class. **Pointers:** Introduction to pointers, * and & operators, assigning addresses to pointer variables, accessing values using pointers, pointers to objects & this pointer, pointers to derived classes **File Handling:** File Stream classes, opening and closing file-file opening modes, text file handling, binary file handling. **Applying OOP to solve real life applications:** To cover case studies like library management, order management etc. to design classes covering all relationships	**15**

Textbooks:

1. Object Oriented Programming with C++, Balagurusamy E., 8th Edition, McGraw Hill Education India.
2. UML & C++: A Practical Guide to Object Oriented Development, Lee/Tepfenhart, Pearson Education, 2nd Edition2015

Additional References:

1. Mastering C++ by Venugopal, Publisher: McGraw-Hill Education, 2017
2. Let Us C++ by KanetkarYashwant, Publisher: BPB Publications, 2020
3. Object Oriented Analysis and Design by Timothy Budd TMH, 2001

Course Code	Course Title	Credits	Lectures /Week
USCSP203	**Introduction to OOPs using C++ - Practical**	**1**	**3**
1	Program todemonstrate use of data members & member functions.		
2	Programs based on branching and looping statements using classes.		
3	Program to demonstrate one and two dimensional arrays using classes		
4	Program to use scope resolution operator. Display the various values of the same variables declared at different scope levels.		
5	Programs to demonstrate various types of constructors and destructors.		
6	Programs to demonstrate use of public, protected & private scope specifiers.		
7	Programs to demonstrate single and multilevel inheritance		
8	Programs to demonstrate multiple inheritance and hierarchical inheritance		
9	Programs to demonstrate inheritance and derived class constructors		
10	Programs to demonstrate friend function, inline function, this pointer		
11	Programs to demonstrate function overloading and overriding.		
12	Programs to demonstrate use of pointers		
13	Programs to demonstrate text and binary file handling		

Course Code	Course Title	Credits	Lectures /Week
USCS204	**Database Systems**	**2**	**3**

About the Course:
The course introduces the core principles and techniques required in the design and implementation of database systems. It includes ER Model, Normalization, Relational Model, and Relational Algebra. It also provides students with theoretical knowledge and practical skills of creating and manipulating data with an interactive query language (MySQL). It also provide student knowledge and importance of data protection.

Course Objectives:

- To make students aware fundamentals of database system.
- To give idea how ERD components helpful in database design and implementation.
- To experience the students working with database using MySQL.
- To familiarize the student with normalization, database protection anddifferent DCL Statements.
- To make students aware about importance of protecting data from unauthorized users.
- To make students aware of granting and revoking rights of data manipulation.

Learning Outcomes:
After successful completion of this course, students would be able to

- To appreciate the importance of database design.
- Analyze database requirements and determine the entities involved in the system and their relationship to one another.
- Write simple queries to MySQL related to String, Maths and Date Functions.
- Create tables and insert/update/delete data, and query data in a relational DBMS using MySQL commands.
- Understand the normalization and its role in the database design process.
- Handle data permissions.
- Create indexes and understands the role of Indexes in optimization search.

Unit	Topics	No of Lectures
I	**Introduction to DBMS –** Database, DBMS – Definition, Overview of DBMS, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture **Data models -** Client/Server Architecture, Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network) **Entity Relationship Model -** Entities, attributes, entity sets, relations, relationship sets, Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER)	**15**

	ER to Table- Entity to Table, Relationship to tables with and without key constraints. **DDL Statements -** Creating Databases, Using Databases, datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables **DML Statements** – Viewing the structure of a table insert, update, delete, Select all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause	
II	**Relational data model**– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint **Relational Algebra** operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division) **Functions** – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse) **Joining Tables** – inner join, outer join (left outer, right outer, full outer) **Subqueries** – subqueries with IN, EXISTS, subqueries restrictions, Nested subqueries, ANY/ALL clause, correlated subqueries	**15**
III	**Schema refinement and Normal forms:** Functional dependencies, first, second, third, and BCNF normal forms based on primary keys, lossless join decomposition. **Database Protection:** Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control, Backing Up and Restoring databases **Views** (creating, altering dropping, renaming and manipulating views) **DCL Statements** (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges), Transaction control commands – Commit, Rollback **Index Structures of Files:** Introduction, Primary index, Clustering Index, Multilevel indexes	**15**

Textbooks:

1. “Fundamentals of Database System”, ElmasriRamez, NavatheShamkant, Pearson Education, Seventh edition, 2017
2. “Database Management Systems”, Raghu Ramakrishnan and Johannes Gehrke, 3rd Edition, 2014
3. “Murach's MySQL”, Joel Murach, 3rd Edition, 3rd Edition, 2019

Additional References:

1. “Database System Concepts”, Abraham Silberschatz,HenryF.Korth,S.Sudarshan, McGraw Hill, 2017
2. “MySQL: The Complete Reference”, VikramVaswani , McGraw Hill, 2017
3. “Learn SQL with MySQL: Retrieve and Manipulate Data Using SQL Commands with Ease”, AshwinPajankar, BPB Publications, 2020

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP204	**Database Systems – Practical**	**1**	**3**
1.	Conceptual Designing using ER Diagrams (Identifying entities, attributes, keys and relationships between entities, cardinalities, generalization, specialization etc.)		
2.	Perform the following: □ Viewing all databases □ Creating a Database □ Viewing all Tables in a Database □ Creating Tables (With and Without Constraints) □ Inserting/Updating/Deleting Records in a Table		
3.	Perform the following: □ Altering a Table □ Dropping/Truncating/Renaming Tables □ Backing up / Restoring a Database		
4.	Perform the following: □ Simple Queries □ Simple Queries with Aggregate functions		
5.	Queries involving □ Date Functions □ String Functions □ Math Functions		
6.	Join Queries □ Inner Join □ Outer Join		

7.	Subqueries □ With IN clause □ With EXISTS clause
8.	Converting ER Model to Relational Model and apply Normalization on database. (Represent entities and relationships in Tabular form, Represent attributes as columns, identifying keys and normalization up to 3rd Normal Form).
9.	Views □ Creating Views (with and without check option) □ Dropping views □ Selecting from a view
10.	DCL statements □ Granting and revoking permissions □ Saving (Commit) and Undoing (rollback)
11.	Creating Indexes on data tables.

Course Code	Course Title	Credits	Lectures /Week
USCS205	Calculus	2	3

About the Course:
Calculus is a branch of mathematics that involves the study of rates of change. In Computer Science, Calculus is used in Machine Learning, Data Mining, Scientific Computing, Image Processing, and creating the graphics and physics engines for video games, including the 3D visuals for simulations.

Course Objectives:

- The primary objective of this course is to introduce the basic tools of Calculus which are helpful in understanding their applications to the real world problems.
- The course is designed to have a grasp of important concepts of Calculus in a scientific way.
- It covers topics from as basic as definition of functions to partial derivatives of functions in a gradual and logical way.
- The learner is expected to solve as many examples as possible to a get compete clarity and understanding of the topics covered.

Learning Outcomes:
After successful completion of this course, learners would be able to:

- Develop mathematical skills and enhance thinking power of learners.
- Understand mathematical concepts like limit, continuity, derivative, integration of functions, partial derivatives.
- Appreciate real world applications which use the learned concepts.
- Skill to formulate a problem through Mathematical modelling and simulation.

Unit	Topics	No of Lectures
I	**DERIVATIVES AND ITS APPLICATIONS:** **Review of Basic Concepts:** Functions, limit of a function, continuity of a function, derivative function. **Derivative In Graphing And Applications:** Increase, Decrease, Concavity, Relative Extreme; Graphing Polynomials, Rational Functions, Cusps and Vertical Tangents. Absolute Maxima and Minima, Applied Maximum and Minimum Problems, Newton‟s Method.	15
II	**INTEGRATION AND ITS APPLICATIONS:** **Integration:** An Overview of the Area Problem, Indefinite Integral, Definition of Area as a Limit; Sigma Notation, Definite Integral, Evaluating Definite Integrals by Substitution, Numerical Integration: Simpson‟s Rule. **Applications of Integration:** Area between two curves, Length of a plane curve. **Mathematical Modeling with Differential Equations**: Modeling with	15

	Differential Equations, Separation of Variables, Slope Fields, Euler‟s Method, First-Order Differential Equations and Applications.	
III	**PARTIAL DERIVATIVES AND ITS APPLICATIONS:** **Functions of Several Variables**: Functions of two or more variables, Limits and Continuity of functions of two or three variables. **Partial Derivatives:** Partial Derivatives, Differentiability, Differentials, and Local Linearity, Chain Rule, Implicit Differentiation, Directional Derivatives and Gradients, **Applications ofPartial Derivatives:** Tangent Planes and Normal Vectors, Maxima and Minima of Functions of Two Variables.	**15**

Textbooks:

1. Calculus: Early transcendental (10th Edition): Howard Anton, IrlBivens, Stephen Davis, John Wiley & sons, 2012.

Additional References:

1. Calculus and analytic geometry (9th edition): George B Thomas, Ross L Finney, Addison Wesley, 1995
2. Calculus: Early Transcendentals (8th Edition): James Stewart, Brooks Cole, 2015.
3. Calculus (10th Edition): Ron Larson, Bruce H. Edwards, Cengage Learning, 2013.
4. Thomas' Calculus (13th Edition): George B. Thomas, Maurice D. Weir, Joel R. Hass, Pearson, 2014.

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP205	**Calculus – Practical**	**1**	**3**
1	**Review of Basic Concepts –** a. Functions of one variable, its domain and range, Operations on functions b. Limits of functions of one variable c. Continuity of functions of one variable d. Derivatives of functions of one variable		
2	**Applications of Derivatives I –** a. Increasing and Decreasing functions b. Concavity and inflection points c. Relative Extrema d. Absolute Extrema		
3	**Applications of Derivatives II –** a. Analysis of polynomials b. Graphing rational functions c. Graphs With Vertical Tangents And Cusps d. Newton‟s method to find approximate solution of an equation		

4	**Integration –** a. Finding area using rectangle method and antiderivative method b. Indefinite and definite integrals c. Properties of integrals d. Numerical integration using Simpson‟s rule.
5	**Applications of Integration –** a. Area between two curves b. Length of a plane curve
6	**Differential Equations –** a. Solution of a first order first degree differential equation using variable separable method b. Solution of a first order linear differential equation using integrating factor c. Numerical solution of first-order equations using Euler‟s method d. Modeling using differential equation
7	**Functions of Several Variables –** a. Functions of two or more variables, its domain and range, Operations on functions, level curves b. Limits of functions of two or three variables c. Continuity of functions of two or three variables
8	**Partial Derivatives I –** a. Partial derivatives of functions, First and Second order partial derivatives, Mixed derivative theorem, Higher order partial derivatives b. Differential for functions of two or three variables c. Local linear approximation for functions of two or three variables
9	**Partial Derivatives II –** a. Chain rule for functions of two or three variables b. Implicit differentiation c. Directional derivatives and gradient
10	**Applications of Partial Derivatives–** a. Tangent Planes and Normal Vectors for functions of two or three variables b. Maxima and Minima of Functions of Two Variables
NOTE	**Above Practical's can also to be implemented using Sage Math/ Geogebra.**

Course Code	Course Title	Credits	Lectures /Week
USCS206	**Statistical Methods**	**2**	**3**

About the Course:
This course introduces the key concepts in probability, conditional probabilities and distribution theory, including probability laws, random variables, expectation and variance, functions of random variables and its probability distributions. Emphasis is placed on theoretical understanding combined with problem solving using various statistical inferential techniques.

Course Objectives:

- To make learner aware about basic probability axioms and rules and its application.
- To understand the concept of conditional probability and Independence of events.
- To make learner familiar with discrete and continuous random variables as well as standard discrete and continuous distributions.
- To learn computational skills to implement various statistical inferential approaches.

Learning Outcomes:
After successful completion of this course, learners would be able to

- Calculate probability, conditional probability and independence.
- Apply the given discrete and continuous distributions whenever necessary.
- Define null hypothesis, alternative hypothesis, level of significance, test statistic and p value.
- Perform Test of Hypothesis as well as calculate confidence interval for a population parameter for single sample and two sample cases.
- Apply non-parametric test whenever necessary.
- Conduct and interpret one-way and two-way ANOVA.

Unit	Topics	No of Lectures
I	**Probability:** Random experiment, sample space, events types and operations of events, Probability definition: classical, axiomatic, Elementary Theorems of probability (without proof). Conditional probability, „Bayes‟ theorem, independence, Examples on Probability. **Random Variables:** Concept and definition of a discrete random variable and continuous random variable. Probability mass function, Probability density function and cumulative distribution function of discrete and continuous random variable, Properties of cumulative distribution function.	**15**
II	**Mathematical Expectation and Variance:** Expectation of a function, Variance and S.D of a random variable, properties. **Standard Probability distributions:** Introduction, properties, examples and applications of each of the following distributions: Binomial distribution, Normal distribution, Chi-square distribution, t distribution, F distribution	**15**

III	**Hypothesis testing:** One sided, Two sided hypothesis, critical region, p-value, tests based on t, Normal and F, confidence intervals. **Analysis of Variance:** One-way, two-way analysis of variance. **Non-parametric tests:** Need of non-parametric tests, Sign test, Wilicoxon‟s signed rank test, run test, Kruskal-Walis tests, Chi square test.	**15**

Textbooks:

1. Gupta, S.C. and Kapoor, V.K. (1987): Fundamentals of Mathematical Statistics, S. Chand and Sons, New Delhi
2. Goon, A. M., Gupta, M. K. and Dasgupta, B. (1983). Fundamentals of Statistics, Vol. 1, Sixth Revised Edition, The World Press Pvt. Ltd., Calcutta.

Additional References:

1. Mood, A. M. and Graybill, F. A. and Boes D.C. (1974). Introduction to the Theory of Statistics, Ed. 3, McGraw Hill Book Company.
2. Hoel P. G. (1971). Introduction to Mathematical Statistics, John Wiley and Sons, New York.
3. Hogg, R.V. and Craig R.G. (1989). Introduction to Mathematical Statistics, Ed. MacMillan Publishing Co., New York.
4. Walpole R. E., Myers R. H. and Myers S. L. (1985), Probability and Statistics for Engineers and Scientists
5. Agarwal, B. L. (2003). Programmed Statistics, Second Edition, New Age International Publishers, New Delhi.

Course Code	Course Title	Credits	Lectures /Week
USCSP206	**Statistical Methods – Practical**	**1**	**3**
1	**Probability-** a. Examples based on Probability definition: classical, axiomatic b. Examples based on elementary Theorems of probability		
2	**Conditional probability and independence-** a. Examples based on Conditional probability b. Examples based on „Bayes‟ theorem c. Examples based on independence		
3	**Discrete random variable-** a. Probability distribution of discrete random variable b. Probability mass function		
4	**Continuous random variable-** a. Probability distribution of continuous random variable b. Probability density function		

5	**Mathematical Expectation and Variance-** a. Mean of discrete and continuous Probability distribution b. S.D. and variance of discrete and continuous Probability distribution
6	**Standard probability distributions-** a. Calculation of probability, mean and variance based on Binomial distribution b. Calculation of probability based on Normal distribution
7	**Large Sample tests based on Normal (Z) -** a. Test of significance for proportion (Single proportion Ho: P = Po) b. Test of significance for difference between two proportions (Double proportion Ho: P1 = P2) c. Test of significance for mean (Single mean Ho: µ = µ0) d. Test of significance for difference between two means. (Double mean Ho: µ1 = µ2)
8	**Small sample tests based on t and F-** a. t-test for significance of single mean, population variance being unknown (Single mean Ho : µ = µ0) b. t-test for significance of the difference between two sample means (Independent samples) c. t-test for significance of the difference between two sample means (Related samples) d. F-Test to Compare Two Variances
9	**Analysis of variance -** a. Perform One-way ANOVA b. Perform Two-way ANOVA
10	**Non-parametric tests-** a. Sign test and Wilcoxon Sign rank test b. Run test c. Kruskal-Wallis (H) test d. Chi-square test
Note: Practical no. 6, 7, 8, 9 can also to be implemented using R programming.	

Course Code	Course Title	Credits	Lectures /Week
USCS207	**E-Commerce & Digital Marketing**	**2**	**3**

About the Course:
This course introduces the fundamental concepts of e-commerce, its types, the various legal and ethical issues of e-commerce and different e-commerce applications. The course also aims to introduce basic principles and types of digital marketing and web and Google analytics

Course Objectives:

- To understand increasing significance of E-Commerce and its applications in Business and Various Sectors
- To provide an insight on Digital Marketing activities on various Social Media platforms and its emerging significance in Business
- To understand Latest Trends and Practices in E-Commerce and Digital Marketing, along with its Challenges and Opportunities for an Organization

Learning Outcomes:
After successful completion of this course, students would be able to

- Understand the core concepts of E-Commerce.
- Understand the various online payment techniques
- Understand the core concepts of digital marketing and the role of digital marketing in business.
- Apply digital marketing strategies to increase sales and growth of business
- Apply digital marketing through different channels and platforms
- Understand the significance of Web Analytics and Google Analytics and apply the same.

Unit	Topics	No of Lectures
I	**Introduction to E-Commerce and E- Business:** Definition and competing in the digital economy, Impact of E-Commerce on Business Models, Factors Driving e-commerce and e-Business Models, Economics and social impact of e-Business, opportunities and Challenges, e-Commerce vs m- Commerce, Different e-Commerce Models (B2B, B2C, C2B, C2C, B2E), e-Commerce Applications: e-Trading, e-Learning, e-Shopping, Virtual Reality & Consumer Experience, Legal and Ethical issues in e-Commerce. **Overview of Electronic Payment system**s: Types of Electronic payment schemes (Credit cards, Debit cards, Smartcards, Internet banking), E-checks, E-Cash Concepts and applications of EDI and Limitation **Introduction & origin of Digital Marketing**: Traditional v/s Digital Marketing. Digital Marketing Strategy, The P-O-E-M Framework, Segmenting & Customizing Messages, The Digital landscape, Digital Advertising Market in India. Skills required in Digital Marketing. Digital Marketing Plan.	**15**
II	**Social Media Marketing:** Meaning, Purpose, types of social media websites, Social Media Engagement, Target audience, Facebook Marketing: Business through Facebook Marketing, Creating Advertising Campaigns,	**15**

	Adverts, Facebook Marketing Tools, LinkedIn Marketing: Importance of LinkedIn Marketing, Framing LinkedIn Strategy, Lead Generation through LinkedIn, Content Strategy, Analytics and Targeting, Twitter Marketing: Framing content strategy, Twitter Advertising Campaigns, YouTube Marketing: Video optimization, Promoting on YouTube, Monetization, YouTube Analytics **Email Marketing:** Types of Emails, Mailing List, Email Marketing tools, Email Deliverability & Email Marketing automation **Mobile Marketing**: Introduction, Mobile Usage, Mobile Advertising, Mobile Marketing Types, Mobile Marketing Features, Mobile Campaign Development, Mobile Advertising Analytics **Content Marketing:** Introduction, Content marketing statistics, Types of Content, Types of Blog posts, Content Creation, Content optimization, Content Management & Distribution, Content Marketing Strategy, Content creation tools and apps, Challenges of Content Marketing.	
III	**Search Engine Optimization:** Meaning, Common SEO techniques, Understanding Search Engines, basics of Keyword search, Google rankings, Link Building, Steps to optimize website, On-page and off-page optimization **Search Engine Marketing:** Introduction to SEM, Introduction to Ad Words - Google Ad Words, Ad Words fundamentals, Ad Placement, Ad Ranks, Creating Ad Campaigns, Campaign Report Generation, Display marketing, Buying Models: Cost per Click (CPC), Cost per Milli (CPM), Cost per Lead (CPL), Cost per Acquisition (CPA). **Web Analytics:** Purpose, History, Goals & objectives, Web Analytic tools & Methods. Web Analytics Mistakes and Pitfalls. **Google Analytics:** Basics of Google Analytics, Installing Google Analytics in website, Parameters of Google Analytics, Reporting and Analysis	**15**

Textbooks:

1. "E-Commerce Strategy, Technologies and Applications", Whitley, David, Tata McGraw Hill, 2017
2. Digital Marketing, Seema Gupta, McGraw Hill Education, 2nd Edition

Additional References:

1. E-Commerce by S. Pankaj, A.P.H. Publication, New Delhi
2. Fundamentals of Digital Marketing, Punit Singh Bhatia, Pearson, 2nd Edition
3. "Understanding Digital Marketing: Marketing Strategies for Engaging the Digital Generation", Damian Ryan, Calvin Jone. Kogan Page, 4th Edition

Evaluation Scheme

I. Internal Evaluation for Theory Courses – 25 Marks

(i) Mid-Term Class Test– 15Marks

- □ It should be conducted using any **learning management system** such as **Moodle** (Modular object-oriented dynamic learning environment)
- □ The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes.**

(ii) Assignment/ Case study/ Presentations– 10 Marks

- □ Assignment / Case Study Report / Presentation can be uploaded on any **learning management system**.

II. External Examination for Theory Courses – 75 Marks

□ Duration: **2.5 Hours**

□ Theory question paper pattern:

	All questions are compulsory.		
Question	**Based on**	**Options**	**Marks**
Q.1	Unit I	*Any 4 out of 6*	20
Q.2	Unit II	*Any 4 out of 6*	20
Q.3	Unit III	*Any 4 out of 6*	20
Q.4	Unit I,II and III	*Any 5 out of 6*	15

□ All questions shall be compulsory with internal choice within the questions.

□ Each Question maybe sub-divided into subquestions as a, b, c, d, etc. & the allocation of Marks depends on the weightage of the topic.

III. Practical Examination

□ Each core subjectcarries50 Marks

40 marks + 05 marks (journal) + 05 marks (viva)

□ Duration: **2 Hours** for each practical course.

□ Minimum **80% practical** from each core subjects are required to be completed.

□ **Certified Journal is compulsory for appearing at the time of Practical Exam**

□ The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

1-11-22
Return

University of Mumbai

No. AAMS_UGS/ICC/2022-23/ 106

CIRCULAR:-

Attention of the Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology is invited to this office circular No. UG/107 of 2017 dated 27th July, 2017 relating to the revised syllabus S.Y.B.Sc.(Computer Science) (Sem . III & IV) (CBCS).

They are hereby informed that the recommendations made by the Ad-hoc Board of Studies in **Computer Science** at its meeting held on 02nd June, 2022 and subsequently passed in the Faculty and then by the Board of Deans at its meeting held on 5th July, 2022 **vide** item No. 6.6 (R) have been accepted by the Academic Council at its meeting held on 11th July, 2022 **vide** item No. 6.6 (R) and that in accordance therewith, the revised syllabus of S.Y.B.Sc.(Computer Science) (Sem.III & IV) (CBCS), has been brought into force with effect from the academic year 2022-23. (The same is available on the University's website www.mu.ac.in).

MUMBAI – 400 032
11th October, 2022

(Dr. Shailendra Deolankar)
I/c Registrar

To

The Principals of the Affiliated Colleges and Directors of the Recognized Institutions in Faculty of Science & Technology.

A.C/6.6(R)/11/07/2022

****

No. AAMS_UGS/ICC/ 2022-23/ 106

11th October, 2022

Copy forwarded with Compliments for information to:-

1) The Dean, Faculty of Science & Technology,
2) The Chairman, Ad-hoc Board of Studies Computer Science,
3) The Director, Board of Examinations and Evaluation,
4) The Director, Board of Students Development,
5) The Director, Department of Information & Communication Technology,
6) The Co-ordinator, MKCL.

(Dr. Shailendra Deolankar)
I/c Registrar

Desktop/Circular Faculty of Science/priya

Copy to :-

1. **The Deputy Registrar, Academic Authorities Meetings and Services (AAMS),**
2. **The Deputy Registrar, College Affiliations & Development Department (CAD),**
3. **The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Department (AEM),**
4. **The Deputy Registrar, Research Administration & Promotion Cell (RAPC),**
5. **The Deputy Registrar, Executive Authorities Section (EA),**
6. **The Deputy Registrar, PRO, Fort, (Publication Section),**
7. **The Deputy Registrar, (Special Cell),**
8. **The Deputy Registrar, Fort/ Vidyanagari Administration Department (FAD) (VAD), Record Section,**
9. **The Director, Institute of Distance and Open Learning (IDOL Admin), Vidyanagari,**

They are requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to in the above circular and that on separate Action Taken Report will be sent in this connection.

1. **P.A to Hon'ble Vice-Chancellor,**
2. **P.A Pro-Vice-Chancellor,**
3. **P.A to Registrar,**
4. **All Deans of all Faculties,**
5. **P.A to Finance & Account Officers, (F.& A.O),**
6. **P.A to Director, Board of Examinations and Evaluation,**
7. **P.A to Director, Innovation, Incubation and Linkages,**
8. **P.A to Director, Board of Lifelong Learning and Extension (BLLE),**
9. **The Director, Dept. of Information and Communication Technology (DICT) (CCF & UCC), Vidyanagari,**
10. **The Director of Board of Student Development,**
11. **The Director, Department of Students Walfare (DSD),**
12. **All Deputy Registrar, Examination House,**
13. **The Deputy Registrars, Finance & Accounts Section,**
14. **The Assistant Registrar, Administrative sub-Campus Thane,**
15. **The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan,**
16. **The Assistant Registrar, Ratnagiri sub-centre, Ratnagiri,**
17. **The Assistant Registrar, Constituent Colleges Unit,**
18. **BUCTU,**
19. **The Receptionist,**
20. **The Telephone Operator,**
21. **The Secretary MUASA**

for information.

AC – 11/7/2022
Item No. – 6.6 (R)

University of Mumbai

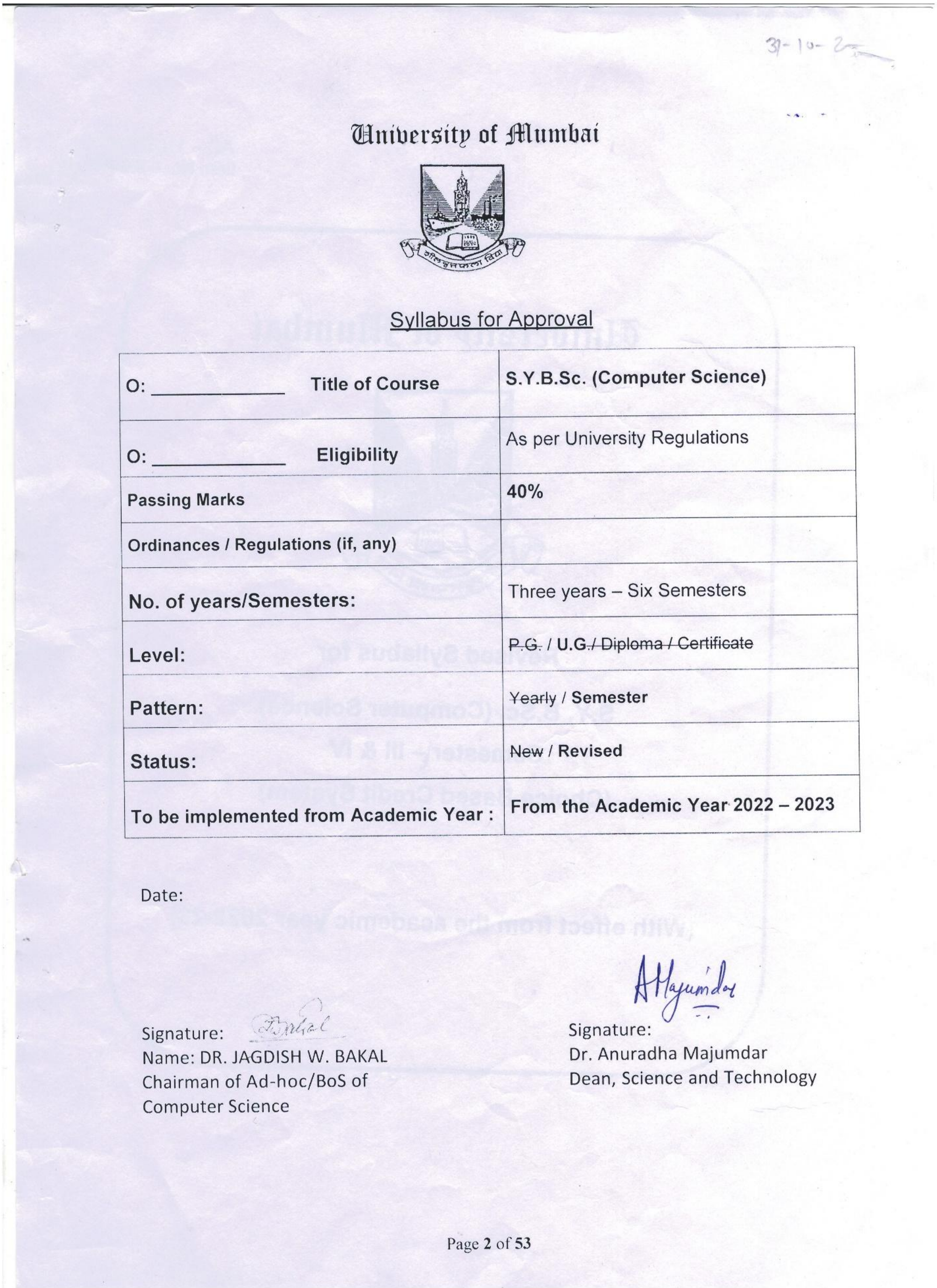
Revised Syllabus for

S.Y. B.Sc. (Computer Science)

Semester – III & IV

(Choice Based Credit System)

(With effect from the academic year 2022-23)



31-10-22

University of Mumbai

Syllabus for Approval

O: ______________ **Title of Course**	**S.Y.B.Sc. (Computer Science)**
O: ______________ **Eligibility**	As per University Regulations
Passing Marks	**40%**
Ordinances / Regulations (if, any)	
No. of years/Semesters:	Three years – Six Semesters
Level:	~~P.G.~~ / **U.G.**~~/ Diploma / Certificate~~
Pattern:	~~Yearly~~ / **Semester**
Status:	~~New~~ / **Revised**
To be implemented from Academic Year :	**From the Academic Year 2022 – 2023**

Date:

AMajumdar

Signature:
Name: DR. JAGDISH W. BAKAL
Chairman of Ad-hoc/BoS of
Computer Science

Signature:
Dr. Anuradha Majumdar
Dean, Science and Technology

Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. It also endeavors to align the programme structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

The Core Subjects offers to develop strong theoretical foundations in Computer Science to build computational thinking, analytical, and problem solving skills. Principles of Operating Systems course provides an overview of computer operating systems, their functionalities, processes, and computing resource management. Linear Algebra course covers concepts crucial to many areas of computer science, such as graphics, image processing, cryptography, machine learning, computer vision, optimization, graph algorithms, quantum computation, computational biology, information retrieval and web search. Data Structures course provides an understanding of different types of data structures and how to use them per the requirements of a given application. Advanced Database Concepts course touches the touches security, recovery, and transaction aspects of database. Theory of Computation course helps to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas. Computer Networks course include topics such as application layer protocols, Internet protocols, network interfaces, local and wide area networks, wireless networks, bridging and routing, among other current topics. Software Engineering course embodies an engineering approach to the development of software. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and estimations along with topics on software testing and quality assurance. The course on IoTTechnologies will definitelyopen future area as Embedded Engineer, involvement in IoT projects, Robotics and many more.

Skill Enhancement courses such as Java based Application Development, Web Technologies, Android Application Development and Advanced Application Developmentcater to present day needs of web and mobile based platforms and applications. These courses aims to produce skilled graduates with a creative mind-set who can recognize a computational problem either in IT industry or society, and develop effective solutions.

The General Elective courses offers the students the option to explore disciplines of interest beyond the choicesthey make in Core and Discipline Specific Elective papers. The course on Creative Content Writing prepare students to comprehend, refine, and enhance their writing abilities and enter the industry with enhanced skill and substantial competence. The course on Green Technologies emphasizes the use of principles and practices of green services and regulatory standards for addressing the carbon issues and related concerns. The Research Methodology instills basic research skills for students who wish to pursue a research or an academic career.Management& Entrepreneurship course aims to focus on giving students the business management and innovation skills required to succeed in a startup.

We sincerely believe that any student taking this programme will get very strong foundation and exposure to basics, advanced and emerging trends of the subject.

We wholeheartedly thank all experts who shared their valuable feedbacks and suggestions in order to improvise the contents, we have sincerely attempted to incorporate each of them. We further thank Chairperson and members of Board of Studies for their confidence in us.

Special thanks to University Department of Computer Science and colleagues from various colleges, who volunteered or have indirectly helped designing certain specialized courses and the syllabus as a whole.

S.Y.B.Sc. Computer Science Syllabus

Choice Based Credit System (CBCS)

with effect from

Academic year 2022-2023

Semester – III				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures/Week**
USCS301	Core Subject	Principles of Operating Systems	2	3
USCSP301	Core Subject Practical	Principles of Operating Systems – Practical	1	3
USCS302	Core Subject	Linear Algebra	2	3
USCSP302	Core Subject Practical	Linear Algebra – Practical	1	3
USCS303	Core Subject	Data Structures	2	3
USCSP303	Core Subject Practical	Data Structures – Practical	1	3
USCS304	Core Subject	Advanced Database Concepts	2	3
USCSP304	Core Subject Practical	Advanced Database Concepts – Practical	1	3
USCS305	Skill Enhancement Course (SEC)	Java based Application Development	2	3
USCSP305	Skill Enhancement Course (SEC) Practical	Java based Application Development – Practical	1	3
USCS306	Skill Enhancement Course (SEC)	Web Technologies	2	3
USCSP306	Skill Enhancement Course (SEC) Practical	Web Technologies – Practical	1	3
USCS3071	Generic Elective	Creative Content Writing	2	3
USCS3072	Generic Elective	Green Technologies	2	3

** Any one Generic Elective has to be selected by the student.*

S.Y.B.Sc. Computer Science Syllabus

Choice Based Credit System (CBCS)

with effect from

Academic year 2022-2023

Semester – IV				
Course Code	**Course Type**	**Course Title**	**Credits**	**Lectures/Week**
USCS401	Core Subject	Theory of Computation	2	3
USCSP401	Core Subject Practical	Theory of Computation – Practical	1	3
USCS402	Core Subject	Computer Networks	2	3
USCSP402	Core Subject Practical	Computer Networks – Practical	1	3
USCS403	Core Subject	Software Engineering	2	3
USCSP403	Core Subject Practical	Software Engineering – Practical	1	3
USCS404	Core Subject	IoT Technologies	2	3
USCSP404	Core Subject Practical	IoT Technologies – Practical	1	3
USCS405	Skill Enhancement Course (SEC)	Android Application Development	2	3
USCSP405	Skill Enhancement Course (SEC) Practical	Android Application Development – Practical	1	3
USCS406	Skill Enhancement Course (SEC)	Advanced Application Development	2	3
USCSP406	Skill Enhancement Course (SEC) Practical	Advanced Application Development – Practical	1	3
USCS4071	Generic Elective*	Research Methodology	2	3
USCS4072	Generic Elective*	Management & Entrepreneurship	2	3

** Any one Generic Elective has to be selected by the student.*

Semester III

Course Code	Course Title	Credits	Lectures /Week
USCS301	**Principles of Operating Systems**	**2**	**3**

About the Course: The purpose of this course is to provide an overview of computer operating systems, their functionalities, processes, and computing resource management. In particular, the course will cover processes and threads, mutual exclusion, CPU scheduling, deadlock, memory management, and file systems.

Course Objectives:

- To learn basic concepts and structure of operating systems
- To learn about process and synchronization in operating system level
- To learn CPU scheduling algorithms
- To learn Memory and File system management

Learning Outcomes:

After successful completion of this course, students would be able to

- Work with any type of operating system
- Handle threads, processes, process synchronization
- Implement CPU scheduling algorithms
- Understand the background role of memory management
- Design file system.

Unit	Topics	No of Lectures
I	**Introduction to Operating-Systems:** Definition of Operating System, Operating System's role, Operating-System Operations, Functions of Operating System, Computing Environments **Operating-System Structures:** Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, Operating-System Structure **Processes:** Process Concept, Process Scheduling, Operations on Processes, Interprocess Communication **Threads:** Overview, Multicore Programming, Multithreading Models	**15**
II	**Process Synchronization:** General structure of a typical process, race condition, The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Classic Problems of Synchronization, Monitors **CPU Scheduling:** Basic Concepts, Scheduling Criteria, Scheduling Algorithms (FCFS, SJF, SRTF, Priority, RR, Multilevel Queue Scheduling,	**15**

	Multilevel Feedback Queue Scheduling), Thread Scheduling **Deadlocks:** System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock	
III	**Main Memory:** Background, Logical address space, Physical address space, MMU, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table **Virtual Memory:** Background, Demand Paging, Copy-on-Write, Page Replacement, Allocation of Frames, Thrashing **Mass-Storage Structure:** Overview, Disk Structure, Disk Scheduling, Disk Management **File-System Interface:** File Concept, Access Methods, Directory and Disk Structure, File-System Mounting, File Sharing **File-System Implementation:** File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management	**15**

Textbook(s):

1. Abraham Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts, Wiley, 2021

Additional Reference(s):

1. Achyut S. Godbole, AtulKahate, Operating Systems, Tata McGraw Hill, 2017
2. Naresh Chauhan, Principles of Operating Systems, Oxford Press, 2014
3. Andrew S Tanenbaum, Herbert Bos, Modern Operating Systems, 4e Fourth Edition, Pearson Education, 2016

Course Code	Course Title	Credits	Lectures /Week
USCSP301	**Principles of Operating Systems – Practical**	**1**	**3**
1	Process Communication: a. Write a program to give a solution to the producer–consumer problem using shared memory. b. Write a program to give a solution to the producer–consumer problem using message passing.		
2	Threads: a. Write a program to work with a single thread. b. Write a program to work with multi threads. c. The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5. 8, ... Formally, it can be expressed as: fib0 = 0, fib1 = 1, fibn = fibn-1 + fibn-2. Write a multithreaded program that generates the Fibonacci sequence.		
3	Synchronization: a. Write a program to give a solution to the Bounded buffer problem. b. Write a program to give a solution to the readers–writers problem.		
4	Write a program that implements FCFS scheduling algorithm.		
5	Write a program that implements (with no premption) scheduling algorithm.		
6	Write a program that implements RR scheduling algorithm.		
7	Write a program that implements the banker’s algorithm		
8	Write a program that implements the FIFO page-replacement algorithm.		
9	Write a program that implements the LRU page-replacement algorithm.		
10	Write a program to design a File System.		

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCS302	**Linear Algebra**	**2**	**3**

About the Course:

Linear algebra, a branch of mathematics, provides concepts that are crucial to many areas of computer science, such as graphics, image processing, cryptography, machine learning, computer vision, optimization, graph algorithms, quantum computation, computational biology, information retrieval and web search. The course covers topics such as fields, vectors, matrices, eigenvalues and eigenvectors

Course Objectives:

- To offer the learner the relevant Linear Algebra concepts through Computer Science applications.
- To interpret existence and analyze the solution set of a system of linear equations.
- To formulate, solve, apply, and interpret properties of linear systems.
- To learn about the concept of linear independence of vectors over a field, and the dimension of a vector space.
- To interpret basic concepts of linear transformations, dimension, matrix representation of a linear transformation, and the change of coordinate matrix.

Learning Outcomes:

After successful completion of this course, students would be able to

- Appreciate the relevance and applications of Linear Algebra in the field of Computer Science.
- Understand the concepts through program implementation.
- Instill a computational thinking while learning linear algebra.
- Express clear understanding of the concept of a solution to a system of equations.
- Find eigenvalues and corresponding eigenvectors for a square matrix.

Unit	**Topics**	**No of Lectures**
I	**Field:** Introduction to complex numbers, complex numbers in Python, abstracting over fields, Playing with GF (2). **Vectors:** Vectors are functions, Vector addition, Scalar-vector multiplication, combining vector addition and scalar multiplication, Dictionary-based representations of vectors, Dot-product, Solving a triangular system of linear equations, Support Vector Machine – Introduction, Mechanism. **The Vector Space:** Linear combination, Span, The geometry of sets of vectors, Vector spaces, Linear systems, homogeneous and otherwise	**15**
II	**Matrix**: Matrices as vectors, Column space and row space, Matrix-vector and vector-matrix multiplication in terms of linear combinations, Matrix-	**15**

	vector multiplication in terms of dot-products, Null space, Computing sparse matrix-vector product, Linear functions, Matrix-matrix multiplication, Inner product and outer product, From function inverse to matrix inverse **Basis**: Coordinate systems, two greedy algorithms for finding a set of generators, Linear dependence, Basis, Unique representation, Change of basis, first look, Computational problems involving finding a basis **Dimension:** Dimension and rank, Direct sum, Dimension and linear functions, The annihilator **Gaussian elimination**: Echelon form, Gaussian elimination over GF(2), Solving a matrix-vector equation using Gaussian elimination.	
III	**Inner Product**: The inner product for vectors over the reals, Orthogonality. **Orthogonalization**: Projection orthogonal to multiple vectors, projecting orthogonal to mutually orthogonal vectors, Building an orthogonal set of generators, orthogonal complement. **Eigenvalues and Eigenvectors:** Characteristic Polynomials of degree 2 and 3, Eigenvalues and eigenvectors, Properties of eigenvalues and eigenvectors, Cayley–Hamilton Theorem, Minimal Polynomial. Coordinate representation in terms of eigenvectors, The Internet worm, Markov Chains, Google Page Rank algorithm.	**15**

Textbooks:

1. Coding the Matrix Linear Algebra through Applications to Computer Science, First Edition, Philip N. Klein, Newtonian Press 2013
2. Schaum's Outline of Linear Algebra, Sixth Edition by Seymour Lipschutz, Marc Lipson, McGraw Hill 2017

Additional References:

1. Linear Algebra and Probability for Computer Science Applications, First Edition, Ernest Davis, A K Peters/CRC Press, 2012.
2. Linear Algebra and Its Applications, Gilbert Strang, Cengage Learning, 4th Edition, 2007
3. Linear Algebra and Its Applications, David C Lay, Pearson Education India; 3rd Edition, 2002
4. Introduction to Information Retrieval, Christopher D. Manning, PrabhakarRaghavan and HinrichSchütze, Cambridge University Press, 2008.
5. Computer Networking With Internet Protocols and Technology, William Stallings, Pearson Education India, 2013.

Course Code	Course Title	Credits	Lectures /Week
USCSP302	**Linear Algebra – Practical**	**1**	**3**
1	Write a program which demonstrates the following: ● Addition of two complex numbers ● Displaying the conjugate of a complex number ● Plotting a set of complex numbers ● Creating a new plot by rotating the given number by a degree 90, 180, 270 degrees and also by scaling by a number a = 1/2, a = 1/3, a = 2 etc.		
2	Write a program to do the following: ● Enter a vector u as a n-list ● Enter another vector v as a n-list ● Find the vector au + bv for different values of a and b ● Find the dot product of u and v		
3	Vector Applications: Classify given data using support vector machines (SVM)		
4	Basic Matrix Operations: ● Matrix Addition, Subtraction, Multiplication ● Check if matrix is invertible. ● If yes then find Inverse		
5	Write a program to convert a matrix into its row echelon form. (Order 2). Write a program to find rank of a matrix.		
6	Basic Matrix Application – I Representation of Image in Matrix Format and Image Transformations		
7	Basic Matrix Application – II Perform Image addition, multiplication and subtraction		
8	Write a program to do the following: ● Enter a vector b and find the projection of b orthogonal to a given vector u. ● Find the projection of b orthogonal to a set of given vectors		
9	Write a program to calculate eigenvalue and eigenvector (Order 2 and 3)		
10	Implement Google’s Page rank algorithm.		

Course Code	Course Title	Credits	Lectures /Week
USCS303	**Data Structures**	**2**	**3**

About the Course:
The course focuses to give an understanding of different types of data structures that can be used to store data in memory, how to create-manipulate them and to use them in the best possible manner as per the requirements of the application.

Course Objectives:

- To introduce data abstraction and data representation in memory
- To describe, design and use of elementary data structures such as stack, queue, linked list, tree and graph
- How and why different data structures are used for different types of problems.

Learning Outcomes:
After successful completion of this course, students would be able to-

- Create different types of data structures.
- Understand which data structure to be used based on the type of the problem.
- Apply combined knowledge of algorithms and data structures to write highly effective programs in various domains.

Unit	Topics	No of Lectures
I	**Abstract Data Type:** Different Data Types, different types of data structures & their classifications, Introduction to ADT, Creating user-specific ADT **Linked Structures**: ADT for linked list, Advantages & Disadvantages, Singly Linked List-Traversing, Searching, Prepending and Removing Nodes, applications of linked list like polynomial equation **Stacks**: Stack ADT for Stack, Advantages & Disadvantages, Applications of stack like balanced delimiter, prefix to postfix notation **Queues**: Queue ADT, Advantages &Disadvantages, linked representations. Circular Queue operations, Dequeues, applications of queue like job scheduling queues	**15**
II	**Doubly Linked list:** ADT of doubly linked list, Advantages & Disadvantages, Insertion and deletion of nodes at various positions **Trees**: ADT for Tree Structure. Advantages &disadvantages, Binary Tree-Properties, Implementation and Traversals, Binary Search Tree, Balanced BST,Threaded Binary Trees, AVL Trees, Applications of Tree like Huffman Coding, **Priority Queues & Heaps:** Priority Queue, Priority Queue ADT,	

	Advantages and Disadvantages, Applications, Heaps, types of heaps, Heapifying the element,	
III	**Graph:** Introduction, Graph ADT, Advantages and Disadvantages, Graph Representation using adjacency matrix and adjacency list, Graph operations like insertion and deletion of nodes, Graph Traversals using BFS & DFS, Applications of Graphs like shortest path algorithms, **Hashing:** Hash Table ADT, Advantages & Disadvantages, Concept of hashing, hash table, hash functions, collision, collision avoidance techniques, Applications of hashing	

Textbooks:

1. Introduction to Algorithm, Thomas H Cormen, PHI
2. Data Structures And Algorithms Made Easy, NarasimhaKarumanchi, 2021

Additional References:

1. Fundamentals of Computer Algorithms, SartajSahni and SanguthevarRajasekaran Ellis Horowitz, Universities Press, 2018
2. Data Structures and Algorithms in Python, Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, Wiley, 2016

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP303	**Data Structures – Practical**	**1**	**3**
1	Write a program to implement Abstract Data Types (ADT)		
2	Write a program to implement Singly Linked list with insertion, deletion, traversal operations		
3	Write a program to implement Doubly Linked list with insertion, deletion, traversal operations		
4	Write a program to implement Stack with insertion, deletion, traversal operations		
5	Write a program to implement Queue with insertion, deletion, traversal operations		
6	Write a program to implement Priority Queue with insertion, deletion, traversal operations		
7	Write a program to implement Binary Tree with insertion, deletion, traversal operations		
8	Write a program to implement Huffman Coding		
9	Write a program to implement Graph with insertion, deletion, traversal operations		
10	Write a program to implement Travelling Salesman Problem		
11	Write a program to create basic Hash Table for insertion, deletion, traversal operations(assume that there are no collisions)		
12	Write a program to create hash table to handle collisions using overflow chaining		

Course Code	Course Title	Credits	Lectures /Week
USCS304	**Advanced Database Concepts**	**2**	**3**

About the Course:
This course deals with the basic understanding of programming in database. It touches security, recovery, and transaction aspects of database. The course will increase the confidence among the learner while dealing with database.

Course Objectives:

- To develop understanding of concepts and techniques for data management and learn about widely used systems for implementation and usage.
- To develop understanding of Transaction management and crash recovery.
- To develop concepts of programming concepts of database.

Learning Outcomes:
After successful completion of this course, students would be able to

- Master concepts of stored procedure, functions, cursors and triggers and its use.
- Learn about using PL/SQL for data management.
- Use efficiently Collections and records.
- Understand concepts and implementations of transaction management and crash recovery.

Unit	Topics	No of Lectures
I	**Overview of PL/SQL:** Advantages of PL/SQL, Main Features of PL/SQL, Architecture of PL/SQL **Fundamentals of PL/SQL:** Character Sets, Lexical Units, Declarations, References to Identifiers, Scope and Visibility of Identifiers, Assigning Values to Variables, Expressions, Error-Reporting Functions, Data Types. **Control Statements:** Conditional Selection Statements, LOOP Statements, Sequential Control Statements, GOTO, and NULL Statements. **Sequences:** creating sequences, referencing, altering, and dropping a sequence. **Stored Procedures and Functions:** Procedures: Types and benefits of stored procedures, creating stored procedures, executing stored procedures, altering stored procedures, viewing stored procedures. Functions: Calling function and recursion function.	**15**
II	**Collections and Records:** Associative Arrays, Varrays (Variable-Size Arrays), Nested Tables, Collection Constructors, Assigning Values to Collection Variables, Multidimensional Collections, Collection Comparisons, Collection Methods, Collection Types Defined in Package	**15**

	Specifications, Record Variables, Assigning Values to Record Variables. **Error Handling:** Compile-Time Warnings, Overview of Exception Handling, Internally Defined Exceptions, Predefined Exceptions, User-Defined Exceptions, Redeclared Predefined Exceptions, Raising Exceptions Explicitly, Exception Propagation, Unhandled Exceptions. **Cursors:** Overview of Cursor, Types of cursors, Invalid cursor Exception. **Static and Dynamic SQL:** Static SQL: Description of Static SQL, Cursors Overview, Processing Query Result Sets, Cursor Variables, CURSOR Expressions, Transaction Processing and Control, Autonomous Transactions. Dynamic SQL: Native Dynamic SQL, DBMS_SQL Package, SQL Injection.	
III	**Triggers:** Overview of Triggers, implementing triggers – creating triggers, Insert, delete, and update triggers, nested triggers, viewing, deleting, and modifying triggers, and enforcing data integrity through triggers. **Packages:** Overview of a Package. Need of Packages, Package Specification, Package Body, Package Instantiation and Initialization. **Transaction Management:** ACID Properties, Serializability, Two-phase Commit Protocol, Concurrency Control, Lock Management, Lost Update Problem, Inconsistent Read Problem, Read-Write Locks, Deadlocks Handling, Two Phase Locking protocol. **Crash Recovery:** ARIES algorithm. The log-based recovery, recovery related structures like transaction and dirty page table, Write-ahead log protocol, check points, recovery from a system crash, Redo and Undo phases	**15**

Textbooks:

1. Mastering PL/SQL Through Illustrations: From Learning Fundamentals to Developing Efficient PL/SQL Blocks, Dr. B. Chandra, BPB Publication, 2020
2. Oracle Pl/Sql Training Guide., Training guide, BPB Publications, 2016
3. Raghu Ramakrishnam, Gehrke, Database Management Systems, McGraw- Hill,3rd Edition, 2014
4. Abraham Silberschatz, Henry F. Korth,S.Sudarshan , Database System Concepts, 6th Edition 2019

Additional References:

1. Ivan Bayross, “SQL, PL/SQL -The Programming language of Oracle”, B.P.B. Publications 2009
2. RamezElmasri&ShamkantB.Navathe, Fundamentals of Database Systems, Pearson Education, 2008

Course Code	Course Title	Credits	Lectures /Week
USCSP304	**Advanced Database Concepts – Practical**	**1**	**3**
1	Writing PL/SQL Blocks with basic programming constructs by including following: a. Sequential Statements b. unconstrained loop		
2	Sequences: a. Creating simple Sequences with clauses like START WITH, INCREMENT BY, MAXVALUE, MINVALUE, CYCLE | NOCYCLE, CACHE | NOCACHE, ORDER | NOORECER. b. Creating and using Sequences for tables.		
3	Writing PL/SQL Blocks with basic programming constructs by including following: a. If...then...Else, IF...ELSIF...ELSE... END IF b. Case statement		
4	Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure: a. While-loop Statements b. For-loop Statements.		
5	Writing PL/SQL Blocks with basic programming constructs by including a GoTO to jump out of a loop and NULL as a statement inside IF.		
6	Writing Procedures in PL/SQL Block a. Create an empty procedure, replace a procedure and call procedure b. Create a stored procedure and call it c. Define procedure to insert data d. A forward declaration of procedure		
7	Writing Functions in PL/SQL Block. a. Define and call a function b. Define and use function in select clause, c. Call function in dbms_output.put_line d. Recursive function e. Count Employee from a function and return value back f. Call function and store the return value to a variable		
8	Creating and working with Insert/Update/Delete Trigger using Before/After clause.		
9	Write an Implicit and explicit cursor to complete the task.		
10	Create packages and use it in SQL black to complete the task.		
11	Write a SQL block to handle exception by writing: a. Predefined Exceptions, b. User-Defined Exceptions,		

	c. Redeclared Predefined Exceptions,
12	Create nested tables and work with nested tables.

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCS305	**Java based Application Development**	**2**	**3**

About the Course:

The objective of this course is to teach the learner how to use Object Oriented paradigm to develop code and understand the concepts of Core Java and explore advanced topic of Java programming for solving problems.

Course Objectives:

- To provide insight into java based applications using OOP concepts.
- To provide understanding of developing GUI based desktop applications in java.
- To provide knowledge of web based applications through servlet and jsp.
- To provide understanding and implementation of basic JSON

Learning Outcomes:

After successful completion of this course, students would be able to

- Design basic application in java using Graphical User Interface.
- The learner will be able to develop applications using swings
- The learner will be able to develop web based applications using servlet and jsp
- The learner will be able to connect databases with java through
- The learner will be able to perform programs using JSON objects

Unit	**Topics**	**No of Lectures**
I	**Introduction:** History, Features of Java, Java Development Kit, Java Application Programming Interface, Java Virtual Machine Java Program Structure, Java Tokens. **OOPS**: Introduction, Class, Object, Static Keywords, Constructors, this keyword, Inheritance,Inner class, Anonymous Inner class, super keyword, Polymorphism (overloading and overriding), Abstraction, Encapsulation, Abstract Classes, Interfaces **Packages:** Introduction to predefined packages, User Defined Packages, Access specifiers **Exception Handling:** Introduction, Pre-Defined Exceptions, try-catch-finally, throws, throw, User Defined Exceptions **Multithreading:** Thread Creations, Thread Life Cycle, Life Cycle Methods,Synchronization, wait() notify() notify all() methods	**15**
II	**Collection Framework**: Introduction, java.util Package interfaces, List,	**15**

	Set, Map,List interface & its classes, Set interface & its classes, Map interface & its classes. **Introduction to JFC and Swing-** Features of the Java Foundation Classes, Swing API Components, JComponent Class, Windows, Dialog Boxes, and Panels, Labels, Buttons, Check Boxes, Menus, Toolbars, Implementing Action interface, Pane, JScrollPane, Desktop pane, Scrollbars, Lists and Combo Boxes, Text-Entry Components, Colors and File Choosers, Tables and Trees, Printing with 2D API and Java Print Service API. **Event Handling:** Delegation Event Model, Events, Event classes, Event listener interfaces, Using delegation event model, adapter classes. **JDBC:**Introduction, JDBC Architecture, JDBC Drivers, JDBC Connectivity Model, java.sql package, Using Statement, PreparedStatement, CallableStatement, ResultSet, Scrollable and Updatable ResultSet, Navigating and manipulating data, ResultSetMetaData, Managing Transactions in JDBC, JDBC Exception classes, BLOB & CLOB	
III	**Servlets:** Introduction, Servlet Life Cycle, Types of Servlet, Servlet Configuration with Deployment Descriptor, Working with ServletContext and ServletConfig Object, Attributes in Servelt,, Response and Redirection using Request Dispacher and using sendRedirect Method, Filter API, Manipulating Responses using Filter API, Session Tracking: using Cookies, HTTPSession, Hidden Form Fields and URL Rewriting,Types of Servlet Event: ContextLevel and SessionLevel. **Java Server Pages (JSP)**: Introduction to JSP , Comparison with Servlet, JSP Architecture, JSP Life Cycle, JSP Scripting Elements, JSP Directives, JSP Action, JSP Implicit Objects, JSP Expression Language, JSP Standard Tag Libraries, JSP Custom Tag, JSP Session Management, JSP Exception Handling, JSP CRUD Applications **JSON:** Overview, Syntax, DataTypes, Objects, Schema, Comparison with XML, JSON with Java	**15**

Textbooks:

1. Herbert Schildt, Java The Complete Reference, Eleventh Edition, McGraw-Hill Education, 2020
2. Bryan Basham, Kathy Sierra, Bert Bates, Head First Servlets and JSP, O'reilly (SPD), 2018
3. Cay S. Horstmann, Gary Cornell, Core Java™ 2: Volume II–Advanced Features Prentice Hall PTR, 2004
4. Ivan Bayross, Web Enabled Commercial Applications Development Using Java 2, BPB Publications
5. Java XML and JSON: Document Processing for Java SE by Jeff Friesen January 2019, Apress

Additional References:

1. **E. Balagurusamy, Programming with Java- A Primer, Tata McGraw-Hill Education India, 2014**
2. **Programming in JAVA, 2nd Ed, Sachin Malhotra &SaurabhChoudhary, Oxford Press, 2018**

3. **Joe Wigglesworth and Paula McMillan, Java Programming: Advanced Topics, Thomson Course Technology (SPD)**
4. **Eric Jendrock, Jennifer Ball, D Carson and others, The Java EE 5 Tutorial, Pearson Education**
5. **The Java Tutorials: http://docs.oracle.com/javase/tutorial/**
6. **Java Parsing Collection XML JSON: Map List XML JSON Transform by Yang Hu, 2019**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP305	**Java based Application Development – Practical**	**1**	**3**
1	a. Write a program to create a class and implement the concepts of Constructor Overloading, Method Overloading, Static methods b. Write a program to implement the concept of Inheritance and Method Overriding		
2	a. Write a program to implement the concepts of Abstract classes and methods b. Write a program to implement the concept of interfaces		
3	Write a program to define user defined exceptions and raise them as per the requirements		
4	Write a program to demonstrate the methods of: a. List interface b. Set interface c. Map interface		
5	Write a program using various swing components design Java application to accept a student's resume. (Design form)		
6	a. Write a JDBC program that displays the data of a given table b. Write a JDBC program to return the data of a specified record from a given table c. Write a JDBC program to insert / update / delete records into a given table		
7	a. Construct a simple calculator using the JAVA Swings with minimum functionality. b. Construct a GUI using JAVA Swings to accept details of a record of a given table and submit it to the database using JDBC technology on the click of a button.		
8	a. Write a Servlet that accepts a User Name from a HTML form and stores it as a cookie. Write another Servlet that returns the value of this cookie and displays it. b. Write a Servlet that displays the names and values of the cookie stored on the client. c. Write a Servlet that accepts a User Name from a HTML form and stores it as a session variable. Write another Servlet that returns the value of this session variable and displays it.		

9	a. Write a registration Servlet that accepts the data for a given table and stores it in the database. b. Write a Servlet that displays all the records of a table.
10	a. Write a JSP that accepts a User Name from a HTML form and stores it as a cookie. Write another JSP that returns the value of this cookie and displays it. b. Write a JSP that displays the names and values of the cookie stored on the client. c. Write a JSP that accepts a User Name from a HTML form and stores it as a session variable. Write another JSP that returns the value of this session variable and displays it.
11	a. Write a JSP code that accepts username and password from HTML file and validates the user from the database b. Write a registration JSP that accept the data for a given table and stores it in the database. c. Write a JSP that displays all the records of a table
12	. Write Java application to encoding and decoding JSON in Java.

Course Code	Course Title	Credits	Lectures /Week
USCS306	**Web Technologies**	**2**	**3**

About the Course:
The course providesan insight into emerging technologies to design and develop state of the art web applications using client-side scripting, server-side scripting, and database connectivity

Course Objectives:

- To understand the concepts of Hyper Text Markup Language and Cascading Style Sheets.
- To learn JavaScript for creating dynamic websites.
- To learn various operations performed on data among web applications using XML
- To learn Server-Side Programming using PHP

Learning Outcomes:
After successful completion of this course, students would be able to

- Design valid, well-formed, scalable, and meaningful pages using emerging technologies.
- Understand the various platforms, devices, display resolutions, viewports, and browsers that render websites
- Develop and implement client-side and server-side scripting language programs.
- Develop and implement Database Driven Websites.
- Design and apply XML to create a markup language for data and document centric applications.

Unit	Topics	No of Lectures
I	**HTML5:** Fundamental Elements of HTML, Formatting Text in HTML, Organizing Text in HTML, Links and URLs in HTML, Tables in HTML, Images on a Web Page, Image Formats, Image Maps, Colors, FORMs in HTML, Interactive Elements, Working with Multimedia - Audio and Video File Formats, HTML elements for inserting Audio / Video on a web page **CSS:** Understanding the Syntax of CSS, CSS Selectors, Inserting CSS in an HTML Document, CSS properties to work with background of a Page, CSS properties to work with Fonts and Text Styles, CSS properties for positioning an element.	**15**
II	**JavaScript:** Using JavaScript in an HTML Document, Programming Fundamentals of JavaScript – Variables, Operators, Control Flow Statements, Popup Boxes, Functions – Defining and Invoking a Function, Defining Function arguments, defining a return Statement, Calling Functions with Timer, JavaScript Objects - String, RegExp, Math, Date, Browser Objects - Window, Navigator, History, Location, Document, Cookies, Document Object Model, Form Validation using JavaScript **XML:** Comparing XML with HTML, Advantages and Disadvantages of XML, Structure of an XML Document, XML Entity References, DTD, XSLT: XSLT Elements and Attributes - xsl:template, xsl:apply-templates,	**15**

	xsl:import, xsl:call-template, xsl:include, xsl:element, xsl:attribute,xsl:attribute-set, xsl:value-of	
III	**AJAX**: AJAX Web Application Model, How AJAX Works, XMLHttpRequest Object – Properties and Methods, handling asynchronous requests using AJAX **PHP**: Variables and Operators, Program Flow, Arrays, working with Files and Directories, working with Databases, Working with Cookies, Sessions and Headers **Introduction to jQuery**: Fundamentals, Selectors, methods to access HTML attributes, methods for traversing, manipulators, events, effects.	**15**

Textbooks:

1. HTML 5 Black Book, Covers CSS 3, JavaScript, XML, XHTML, AJAX, PHP and jQuery, 2ed, Dreamtech Press, 2016
2. Web Programming and Interactive Technologies, scriptDemics, StarEdu Solutions India, 2018
3. PHP: A Beginners Guide, VikramVaswani, TMH

Additional References:

1. **HTML, XHTML, and CSS Bible Fifth Edition, Steven M. Schafer, WILEY, 2011**
2. **Learning PHP, MySQL, JavaScript, CSS & HTML5, Robin Nixon, O’Reilly, 2018**
3. **PHP, MySQL, JavaScript & HTML5 All-in-one for Dummies, Steve Suehring, Janet Valade Wiley, 2018**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP306	**Web Technologies – Practical**	**1**	**3**
1	Design a webpage that makes use of a. Document Structure Tags b. Various Text Formatting Tags c. List Tags d. Image and Image Maps		
2	Design a webpage that makes use of a. Table tags b. Form Tags (forms with various form elements) c. Navigation across multiple pages d. Embedded Multimedia elements		
3	Design a webpage that make use of Cascading Style Sheets with a. CSS properties to change the background of a Page b. CSS properties to change Fonts and Text Styles c. CSS properties for positioning an element		
4	Write JavaScript code for a. Performing various mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number b. Validating the various Form Elements		
5	Write JavaScript code for a. Demonstrating different JavaScript Objects such as String, RegExp, Math, Date b. Demonstrating different JavaScript Objects such as Window, Navigator, History, Location, Document, c. Storing and Retrieving Cookies		
6	Create a XML file with Internal / External DTD and display it using a. CSS b. XSL		
7	Design a webpage to handle asynchronous requests using AJAX on a. Mouseover b. button click		
8	Write PHP scripts for a. Retrieving data from HTML forms b. Performing certain mathematical operations such as calculating factorial / finding Fibonacci Series / Displaying Prime Numbers in a given range / Evaluating Expressions / Calculating reverse of a number c. Working with Arrays		

	d. Working with Files (Reading / Writing)
9	Write PHP scripts for a. Working with Databases (Storing Records / Reprieving Records and Display them) b. Storing and Retrieving Cookies c. Storing and Retrieving Sessions
10	Design a webpage with some jQuery animation effects.

Course Code	Course Title	Credits	Lectures /Week
USCS3071	**Creative Content Writing**	**2**	**3**

About the Course:
With the advent of the internet, content writing has become a very lucrative and promising career. The course is designed to equip students to comprehend, refine, and enhance their writing abilities so that they may become proficient web content developers. The course aims to prepare students to enter the industry with enhanced skill and substantial competence.

Course Objectives:

- To introduce students to the concepts of content writing.
- To connect them with various writing and editing styles and techniques.
- To help them develop their creative abilities.
- To improve the learners' employability

Learning Outcomes:
After successful completion of this course, students would be able to

- Understand the fundamentals of content creation for Blog, Website etc.
- Acquire the ability to write and edit in a variety of styles and procedures
- To develop the creative abilities.
- To acquire essential language skills for editors.

Unit	Topics	No of Lectures
I	**Basics of Content writing:**Introduction to Content Writing, Learning Tone in Writing and Its Types, Comprehending style in writing and its Types, Common Grammatical Errors. **Best Practices for Writing for the Web:** Making our storyElegant, Professional, Write with an Attitude, Keep Verbs Active, List Items, Chunk Information, Title and Subtitle, Organize for Your Audience. **Things Marketers Write:** The Ideal Length for Blog Posts, Podcast, Facebook Posts, Tweets, and Other Marketing Content.	**15**
II	**Social Media Writing:** Writing for Twitter, writing with Hashtags, Writing Social Media with Humor, writing for Facebook, writing for LinkedIn, Writing Your LinkedIn Profile, writing for Email, Writing Landing Pages, Writing Headlines, writing a Home Page, Writing the About Us Page, Writing Better Blog Posts, Writing Annual Reports. **Infographics:** Visual Communication-What Are Infographics?, The Science of Visualization, Creating Infographics-Purpose, The Art of Observation, Processing Your Ideas, Designing Your Infographics, Publishing Your Infographics.	**15**

III	**Content Tools:** Research and Knowledge Management Tools, Writing Tools, Productivity Tools, Editing Tools, A Few Great Style Guides, Non-Text Writing Tools, Blog Idea Generators, Google Authorship, Image Sources,Tools for Content Writing. **Ethical and Legal aspects of content writing:** Learn Legal English, Learn Legal Vocabulary In Legal Writing, IPR Laws, and Copywriting, Plagiarism laws in Content Writing**.**	**15**

Textbooks:

1. Content Writing Handbook, Author:Kounal Gupta, 2020, Henry Harvin.
2. Feldar, Lynda. Writing for the Web: Creating Compelling Web Content Using Words, Pictures, and Sound. New Riders, CA, USA, 2011

Additional References:

1. **Everybody Writes: Your Go-To Guide to Creating Ridiculously Good Content Paperback Ann Handley Pan Macmillan India 2016**
2. **The Power of Infographics: Using Pictures to Communicate and Connect With Your Audiences Paperback – 15 June 2012 Mark Smiciklas**
3. **Law Relating to Intellectual Property Rights Book by V. K. Ahuja, 2017**

Web Resources:

1. https://www.locationrebel.com/b2b-writing/
2. https://www.mindler.com/blog/how-to-become-a-content-writer-in-india/
3. https://study.com/articles/What_is_a_Content_Writer.html
4. https://www.mondaq.com/india/contracts-and-commercial-law/445620/legal-contractsagreements-drafting-and-legal-vetting
5. https://www.crazyegg.com/blog/copywriting/

Course Code	Course Title	Credits	Lectures /Week
USCS3072	Green Technologies	2	3

About the Course:

This course focuses on familiarizing learners with the need and relevance of Green Computing, Technology, and its practices for creating a sustainable work and production environment for the IT-enabled sector. The course emphasizes the use of principles and practices of green services and regulatory standards for addressing the carbon issues and related concerns.

Course Objectives:

- Know about Green IT Fundamentals: Business, IT, and the Environment
- Green IT Strategies and Significance of Green IT Strategies
- Green Enterprise Architecture and Green Information Systems
- Sociocultural Aspects of Green IT and Green Compliance

Learning Outcomes:

After successful completion of this course, students would be able to

- Explain drivers and dimensions of change for Green Technology
- Appreciate Virtualization; smart meters and optimization in achieving green IT
- Gain knowledge about green assets, green processes, and green enterprise architecture
- ISO 14001 and related standards for Audit for Green Compliance

Unit	Topics	No of Lectures
I	**Green IT Fundamentals:** Information Technology and Environment, Business, Environment, and Green Enterprise Characteristics, Green Vision and Strategic Points, Green Value, Green IT Opportunity, Challenges of a Carbon Economy, Environmental Intelligence, Envisioning the Green Future **Green IT Strategies**: Green strategic alignment, Green IT Drivers-Cost, Regulatory and Legal, Sociocultural and Political, Business ecosystem, New market opportunities, Green IT Business Dimensions, KPIs in Green Strategies **Environmentally Responsible Business:** Developing ERBS, Policies, Practices, and Metrics, Mobility and Environment, Green It Metrics and Measurements, Green IT Readiness and CMM, Context Sensitivity and Automation in Green IT Measures **Green Assets:** Introduction, Green Assets, Green IT Hardware, Green Data Centers and ICT Equipment, Server and Data Strategy	15
II	**Green Assets and emerging Trends:** Data Servers Optimization and Virtualization, Physical Data Server Organization and Cooling, Cloud Computing and Data Centers, Networking and Communications	15

	Infrastructure, End-User Devices, Smart Meters in Real-Time, Managing Devices for Central Green Services, Devices and Organizational Boundaries for Measurements, Mobile Devices, and Sustainability **Green Business Process Management:** Introduction, Green Reengineering, Green Process, Green BPM and standards, Green Business Analysis, Green Requirements Modelling, Green IT Governance, Green Business Process and Applications, QoS, Achieving green BPM, Green Mobile Business Process, Digital Library **Green Enterprise Architecture:** Green IT and organizational Systems, Aspects of Green Solutions Architecture, Contents and Integration with Service-Oriented Architecture, Green Supply Chain Management, Green Portals in Green Enterprise Architecture, Environmental Intelligence	
III	**Green Information Systems(GIS): Design and Development Models:** Describing GIS, GIS Requirements **Sociocultural Aspects of Green IT:** Green IT's Social Impact, Learning Organization, Green Social Stakeholders, Role-Based View of Green IT, Green User Practices, Attitude and Subjectivity in Green IT, Green IT Ethics and Code of Conduct, Privacy and Security of Green Information, Green Washing, Communications in Green Transformation Projects, Green HR and Changing Organizational Structures, Green-Collar Workers: Roles and Skill Sets, Green Virtual Communities **Green Compliance: Protocols, Standards, and Audits:** Protocols and Standards, ISO 14000-2004 Standard, Various initiatives by stakeholders, Green Audits and types, Audit and use of Carbon emission management software **Emerging Carbon Issues:** Technologies and Future: Future Carbon Landscape, Green ICT and Technology Trends, Cloud Computing, Nanotechnology, Quantum computing, Renewable energies, eco-design, Collaborative environmental intelligence	**15**

Textbooks:

1. Green IT Strategies and Applications Using Environmental Intelligence, BhuvanUnhelkar, CRC Press, 2016
2. Green Information and Communication Systems for a Sustainable Future, Rajshree Srivastava, Sandeep Kautish, Rajeev Tiwari. CRC Press, 2020

Additional References:

1. **Emerging Green Technologies, Matthew N. O. Sadiku, Taylor and Francis (CRC Press), 2022**
2. **Sustainability Awareness and Green Information Technologies, TomayessIssa, Springer, 2021**
3. **Environmental Sustainability Role of Green Technologies, P. Thangavel, and G. Sridevi, Springer, 2016**

Semester IV

Course Code	Course Title	Credits	Lectures /Week
USCS401	**Theory of Computation**	2	3

About the Course:
The course providesa comprehensive insight into theory of computation by understanding grammar, languages and other elements of modern language design. It also helps to develop capabilities to design and develop formulations for computing models and identify its applications in diverse areas.

Course Objectives:

- To give an overview of the theoretical foundations of computer science from the perspective
- of formal languages
- To illustrate finite state machines to solve problems in computing
- To explain the hierarchy of problems arising in the computer sciences.
- To familiarize Regular grammars, context frees grammar.

Learning Outcomes:
After successful completion of this course, students would be able to

- Understand Grammar and Languages
- Learn about Automata theory and its application in Language Design
- Learn about Turing Machines and Pushdown Automata
- Understand Linear Bound Automata and its applications

Unit	Topics	No of Lectures
I	**Automata Theory:** Defining Automaton, Finite Automaton, Transitions and Its properties, Acceptability by Finite Automaton, Nondeterministic Finite State Machines, DFA and NDFA equivalence, Mealy and Moore Machines, Minimizing Automata. **Formal Languages:** Defining Grammar, Derivations, Languages generated by Grammar, Chomsky Classification of Grammar and Languages, Recursive Enumerable Sets, Operations on Languages, Languages and Automata	15
II	**Regular Sets and Regular Grammar:** Regular Grammar, Regular Expressions, Finite automata and Regular Expressions, Pumping Lemma and its Applications, Closure Properties, Regular Sets and Regular Grammar **Context Free Languages:** Context-free Languages, Derivation Tree, Ambiguity of Grammar, CFG simplification, Normal Forms, Pumping	15

	Lemma for CFG **Pushdown Automata:** Definitions, Acceptance by PDA, PDA and CFG	
III	**Linear Bound Automata:** The Linear Bound Automata Model, Linear Bound Automata and Languages. **Turing Machines:** Turing Machine Definition, Representations, Acceptability by Turing Machines, Designing and Description of Turing Machines, Turing Machine Construction, Variants of Turing Machine, **Undecidability:** The Church-Turing thesis, Universal Turing Machine, Halting Problem, Introduction to Unsolvable Problems	**15**

Textbooks:

1. Theory of Computer Science, K. L. P Mishra, Chandrasekharan, PHI,3rd Edition 2019
2. Introduction to Computer Theory, Daniel Cohen, Wiley,2nd Edition, 2007
3. Introductory Theory of Computer Science, E.V. Krishnamurthy, Affiliated East-West Press, 2009

Additional References:

1. **Theory of Computation, Kavi Mahesh, Wiley India, 2018**
2. **Elements of The Theory of Computation, Lewis, Papadimitriou, PHI, 2015**
3. **Introduction to Languages and the Theory of Computation, John E Martin, McGraw-Hill Education, 2010**
4. **Introduction to Theory of Computation, Michel Sipser, Thomson**
5. **Introduction to Automata Theory, Languages and Computation, John E. Hopcroft, Pearson Education, 2014**

Course Code	Course Title	Credits	Lectures /Week
USCSP401	Theory of Computation – Practical	1	3
1	Write a program for tokenization of given input		
2	Write a program for generating regular expressions for regular grammar		
3	Write a program for generating derivation sequence / language for the given sequence of productions		
4	Design a Program for creating machine that accepts three consecutive one.		
5	Design a Program for creating machine that accepts the string always ending with 101.		
6	Design a program for accepting decimal number divisible by 2.		
7	Design a program for creating a machine which accepts string having equal no. of 1’s and 0’s.		
8	Design a program for creating a machine which count number of 1’s and 0’s in a given string.		
9	Design a PDA to accept WCWR where w is any string and WR is reverse of that string and C is a Special symbol.		
10	Design a Turing machine that’s accepts the following language an b n c n where n>0		

Course Code	Course Title	Credits	Lectures /Week
USCS402	**Computer Networks**	**2**	**3**

About the Course:
This course introduces computer networks, with a special focus on the Internet architecture and protocols. The course includes topics such as network architectures, addressing, naming, forwarding, routing, communication reliability, the client-server model, web, email and other application layer protocols.

Course Objectives:

- To Understand Basic Concepts of Networking.
- To Understand Working of Network Layer Architecture.
- To Learn Practical Implementation of Basic Routing Algorithms.
- To Learn Different Networking Protocols.

Learning Outcomes:
After successful completion of this course, students would be able to

- Learn basic networking concepts and layered architecture.
- Understand the concepts of networking, which are important for them to be known as a ‘networking professionals’.

Unit	Topics	No of Lectures
I	**Introduction:** Networking standards and Administrations, networks, network types – LAN, MAN, WAN. **Network Models:** The OSI model, TCP/IP protocol suite, **Introduction to Physical layer:** Data and signals, periodic analog signals, digital signals, transmission impairment, data rate limits, performance. **Digital transmissions:** Digital-to-digital conversion, analog-to-digital conversion, transmission modes **Analog transmissions:** digital-to-analog conversion, analog-to-analog conversion. **Bandwidth Utilization – Multiplexing and Spectrum spreading:** Multiplexing, Spread Spectrum **Transmission media:** Guided Media, Unguided Media **Switching:** Introduction, Circuit Switched Network, Packet Switching.	**15**
II	**Introduction to Data Link Layer:** Link layer addressing, Data Link Layer Design Issues. **Error detection and correction**: -Block coding, cyclic codes, checksum,	**15**

	forward error correction, error correcting codes, error detecting codes. **Data Link Control:** DLC services, data link layer protocols, HDLC, Point-to-point protocol. **Media Access Control:** Random access, controlled access, channelization, **Wired LANs – Ethernet:** Ethernet Protocol, standard Ethernet, fast Ethernet, gigabit Ethernet, 10 gigabit Ethernet **Wired Network:** Telephone Network, Cable Network, SONET, ATM **Wireless LANs:** Introduction, IEEE 802.11 project, Bluetooth, WiMAX, Cellular telephony, Satellite networks. **Introduction to Network Layer:** Network layer services, packet switching, network layer performance, IPv4 addressing, forwarding of IP packets, **Network Layer Protocols** : Internet Protocol, ICMPv4, Mobile IP	
III	**Unicast Routing:** Introduction, routing algorithms, unicast routing protocols. **Next generation IP:** IPv6 addressing, IPv6 protocol, ICMPv6 protocol, transition from IPv4 to IPv6. **Introduction to the Transport Layer:** Transport Layer Protocol, User Datagram Protocol, Transmission Control Protocol, SCTP. **Introduction to Application Layer:** Client Server Programming, Iterative Programming. **Standard Client-Server Protocols:** WWW, HTTP, FTP, Electronic Mail, TELNET, Secure Cell, DNS, SNMP **Quality of Service:** Data Flow to improve QoS, Flow control to improve QoS, Integrated service (Intserv), Differentiated Service(Diffserv).	**15**

Textbooks:

1. Data Communications and Networking, Behrouz A. Forouzan, Fifth Edition, TMH, 2018.
2. Computer Network, Andrew S. Tanenbaum, David J. Wetherall, Fifth Edition, Pearson Education, 2018.

Additional References:

1. **Computer Network, Bhushan Trivedi, Oxford University Press, 2016**
2. **Data and Computer Communication, William Stallings, PHI, 2017**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP402	**Computer Networks – Practical**	**1**	**3**
1	Using, linux-terminal or Windows-cmd, execute following networking commands and note the output: ping, traceroute, netstat, arp, ipconfig, Getmac, hostname, NSLookUp, pathping, SystemInfo		
2	Using Packet Tracer, create a basic network of two computers using appropriate network wire. Use Static IP address allocation and show connectivity		
3	Using Packet Tracer, create a basic network of One server and two computers using appropriate network wire. Use Dynamic IP address allocation and show connectivity		
4	Using Packet Tracer, create a basic network of One server and two computers and two mobile / movable devices using appropriate network wire. Show connectivity		
5	Using Packet Tracer, create a network with three routers with RIPv1 and each router associated network will have minimum three PC. Show Connectivity		
6	Using Packet Tracer, create a network with three routers with RIPv2 and each router associated network will have minimum three PC. Show Connectivity		
7	Using Packet Tracer, create a network with three routers with OSPF and each router associated network will have minimum three PC. Show Connectivity		
8	Using Packet Tracer, create a network with three routers with BGP and each router associated network will have minimum three PC. Show Connectivity		
9	Using Packet Tracer, create a wireless network of multiple PCs using appropriate access point.		
10	Using Wireshark, network analyzer, set the filter for ICMP, TCP, HTTP, UDP, FTP and perform respective protocol transactions to show/prove that the network analyzer is working		

Course Code	Course Title	Credits	Lectures /Week
USCS403	**Software Engineering**	**2**	**3**

About the Course:
This course covers a collection of methods which embody an "engineering" approach to the development of software. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and estimations. It also underlines the topics on software testing and quality assurance.

Course Objectives:

- To learn and understand the Concepts of Software Engineering
- To learn and understand Software Development Life Cycle
- To apply the project management and analysis principles to software project development.
- To apply the design & testing principles to software project development.

Learning Outcomes:
After successful completion of this course, students would be able to

- Plan a software engineering process life cycle, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements
- Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology.
- Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice
- Able to use modern engineering tools necessary for software project management, time management and software reuse.

Unit	Topics	No of Lectures
I	**Introduction:** The Nature of Software, Software Engineering, Professional Software Development, Layered Technology, Process framework, CMM, Process Patterns and Assessment Prescriptive Models: Waterfall Model, Incremental, RAD Models Evolutionary Process Models: Prototyping, Spiral and Concurrent Development Model Specialized Models: Component based, Aspect Oriented development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming **Requirement Analysis and System Modeling:** Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS, Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram	**15**

II	**System Design:** System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design **Software Measurement and Metrics:** Process Metrics and Project Metrics, Software Measurement, Object Oriented Metrics, Software Project Estimation, Decomposition Techniques, LOC based, FP based and Use case based estimations, Empirical estimation Models **SoftwareProjectManagement:Estimation**inProjectPlanningProcess –Software Scope and Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision **Project Scheduling -** Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts	15
III	**RiskManagement**-Risk strategies, Software risks, Risk Identification, projection, RMMM Quality Concepts **Software Quality Assurance** SQA activities, Software reviews, FTR, Software reliability and measures, SQA plan Software Configuration Management, elements of SCM, SCM Process, Change Control Capability Maturity Model **Software Testing :**Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-CaseDesign	15

Textbooks:

1. Software Engineering, A Practitioner’s Approach, Roger S, Pressman, 2019
2. Software Engineering: principles and Practices, Deepak Jain, OXFORD University Press, 2008

Additional References:

1. **Software Engineering, Ian Sommerville, Pearson Education, 2017**
2. **Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI, 2018**
3. **Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons, 2010**
4. **A Concise Introduction to Software Engineering, Pankaj Jalote, Springer**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP403	**Software Engineering – Practical**	**1**	**3**
Perform the following exercises for any two projects given in the list of sample projects or any other projects:			
1	**Write down the problem statement for a suggested system of relevance**		
2	**Perform requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.**		
3	**Draw the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.**		
4	**Draw the user‘s view analysis for the suggested system: Use case diagram.**		
5	**Draw the structural view diagram for the system: Class diagram, object diagram.**		
6	**Draw the behavioral view diagram : State-chart diagram, Activity diagram**		
7	**Draw the behavioral view diagram for the suggested system: Sequence diagram, Collaboration diagram**		
8	**Draw the implementation and environmental view diagram: Component diagram, Deployment diagram**		
9	**Perform Estimation of effort using FP Estimation**		
10	**Prepare time line chart/Gantt Chart/PERT Chart**		
11	**Develop test cases for unit testing and integration testing**		
12	**Develop test cases for various white box and black box testing**		

List of sample projects

a. **Student Result Management System**
b. **Library management system**
c. **Inventory control system**
d. **Accounting system**
e. **Fast food billing system**
f. **Bank loan system**
g. **Blood bank system**
h. **Railway reservation system**
i. **Automatic teller machine**
j. **Video library management system**
k. **Hotel management system**
l. **Hostel management system**
m. **Share online trading**
n. **Hostel management system**
o. **Resource management system**
p. **Court case management system**

Course Code	Course Title	Credits	Lectures /Week
USCS404	IoT Technologies	2	3

About the Course:

The course aims to provide basic understanding of SoC architectures; IoT, different types of IoT platforms and different types of applications that can be built.

Course Objectives:

- Introduce concepts of SoC and IoT
- Introduce various types of IoT platforms
- Interfacing various types of devices using different protocols with IoT
- Understand practical applications of IoT in real life world

Learning Outcomes:

After successful completion of this course, students would be able to

- understand SoC and IoT
- use different types of IoT Platforms and interfaces
- understand and implement an idea of various types of applications built using IoT

Unit	Topics	No of Lectures
I	**Fundamentals of IoT:** Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M. **System on Chip:** What is System on chip? Structure of System on Chip. SoCElements: FPGA, GPU, APU, Compute Units. **Different types of IoT/SoC Platforms:** Introduction to Raspberry Pi, Arduino &NodeMCU, Introduction to SoC-ARM Architecture, atmega328 architecture	15
II	**Interfacing with IoT Platforms:** Basic hardware components like LED, Button, Camera, 8X8 LED Grid, Motor etc and interfacing them for input/output with IoT devices using PWM, UART, GPIO, I2C, SPI **Using Sensor &Actuators**:Overview of Sensors working, Analog and Digital Sensors, Interfacing of Temperature, Humidity, Motion, Light and Gas Sensor, Level Sensors, Ultrasonic sensors, Interfacing of Actuators, Interfacing of Relay Switch and Servo Motor **IoT and Protocols IoT Security:** HTTP, UPnP, CoAP, MQTT, XMPP, Privacy and Security Issues in IoT.	15
III	**IoT& Web:** Web server for IoT, Sending/Receiving data between web server &IoT device,Cloud for IoT, Node RED, M2M vs IoT	15

	Communication Protocols, Basics of WSNs, WSN architecture and types, **IoT Applications:** Modern IoT case studies / applications used in the areas of transportation, agriculture, health care etc **Edge Computing:**Edge computing purpose anddefinition, Edge computing use cases, Edge computing hardware architectures, Edge platforms,Edge vs Fog Computing, Communication Models - Edge, Fog and M2M.	

Textbooks:

1. Introduction to IoT Paperback by SudipMisra , Anandarup Mukherjee , Arijit Roy , Cambridge Press, 2022
2. Jain, Prof. Satish, Singh, Shashi, “Internet of Things and its Applications”, 1st Edition, BPB, 2020.
3. Shriram K Vasudevan, Abhishek S Nagarajan, RMD Sundaram, Internet of Things, Wiley, India, 2019
4. IoT and Edge Computing for Architects - Second Edition, by Perry Lea, Publisher: PacktPublishing, 2020

Additional References:

1. Internet of Things by VinayakShinde, SYBGEN Learning India Pvt. Ltd, 2020
2. Internet of things, Dr. KamleshLakhwani, Dr. Hemant kumarGianey, Josef Kofi Wireko, KamalkantHiran, BPB Publication, 2020
3. Arduino, Raspberry Pi, NodeMCU Simple projects in easy way by Anbazhagan k and AmbikaParameswari k, 2019.
4. IoT based Projects: Realization with Raspberry Pi, NodeMCU Paperback – February 2020, by Rajesh Singh Anita Gehlot, 2020
5. Mastering the Raspberry Pi, Warren Gay, Apress, 2014

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP404	**IoT Technologies – Practical**	**1**	**3**
1	Preparing Raspberry Pi: Hardware preparation and Installation		
2	Demonstrate Arduino Uno and its pins interfacing with IDE.		
3	GPIO: Light the LED with Python with/without a button using either Uno/Raspberry Pi.		
4	SPI: Camera Connection and capturing Images/Videos using SPI		
5	GPIO: LED Grid Module: Program the 8X8 Grid with Different Formulas		
6	Stepper Motor Control: PWM to manage stepper motor speed using Uno/Raspberry Pi.		
7	Node RED: Connect LED to Internet of Things		
8	Use different types of sensors (LDR, Temperature) with Raspberry Pi/Uno.		
9	Trigger a set of led GPIO on any IoT platform via any related web server		
10	Interface with any sensor and send its value over the internet to the server using any suitable protocol		

Course Code	Course Title	Credits	Lectures /Week
USCS405	**Android Application Development**	**2**	**3**

About the Course:

This course is aimed at creating a skilled IT workforce that is focused on developing Apps for mobile and smart Android-based computing platforms. It familiarises the development of android applications using Kotlin for problems that address real-life needs ranging from intuitive UI to rich multimedia experience.

Course Objectives:

- Kotlin Programming Language for application development
- Creating robust mobile applications on simulators and physical devices
- Creating intuitive, reliable mobile apps using the android services and components
- Handling data local and remote data storage
- Create a seamless user interface that works with different mobile screens

Learning Outcomes:

After successful completion of this course, students would be able to

- Build useful mobile applications using Kotlin language on Android
- Install and configure Android Studio for application development
- Master basic to intermediate concepts of Kotlin required for mobile application development
- Use built-in widgets and components, work with the database to store data
- Master key Android programming concepts and deploy the application on Google Play

Unit	Topics	No of Lectures
I	**Introduction to Kotlin**: Basics of Kotlin, type conversions, comments, Kotlin operators, variables in Kotlin, packages, visibility modifiers, control flow statements, Concept of OOPS in Kotlin, classes in Kotlin, delegation and extension functions, the companion object, **Advanced Concepts in Kotlin:** declaring and calling functions, parameters, and arguments in Kotlin, default argument, variable number of arguments, unit-returning function, explicit return type, lambda expression, coroutines, Collections in Kotlin, Mutable and Immutable Collections, Ranges, type Checks, casting concept, this expression, Null safety, exception handling, annotations **App Development with Android Studio**: Android Architecture, Android Application Framework, Android Virtual Device, Creating and running First Android Application, working with Physical Android Device, Adding Kotlin Files in Android Studio **Basics Of Android-** Application Components: Activities, Intent, and Broadcast Receiver, Services, Fragment, Activity Life Cycle, Content Provider, Widgets, and Notifications	15

II	**Designing Android UI**: User Interface (UI), Layout and Its Types, Layout Attribute, working with Views, Android UI Controls, Styles and Themes, Event Handler, setting up themes in Manifest and from the application, dialog in activity, using intents, fragments **Handle Images, ListviewAnd Menu**: ImageView, ImageSwitcher, ListView, Menu, and its types,Designing menu in XML, Option menu, Context menu, popup menu, **Screen** Navigation, RecyclerView, Interaction of Views **Data binding in Android-**AdapterView, Spinner, Gallery view, AutotextCompleteView, screen orientation, Designtheviewdynamically **Implementing Data Persistence**: Data Storage-Shared Preference, Internal And External Storage Storing Data Using SQLite Databases, Content Provider, Firebase Real-Time Data	15
III	**Graphics, Animations, and Integrating Media in Android:**Drawable Class, Animation in Android, MediaPlayer API and in Android, Mediaplayer and AudioManger Class, **Interacting With Camera** and input gestures: Android Camera, Input gestures-multiple touch, swipe, drag, scroll, zoom, Recording **Gathering Location Data:** **Managing Background Tasks:**Broadcaset Receivers, Services, Threads and Process, AsyncTask, JobScheduler, Manage device Awake State **Deploying Android applications on Google Play**-Publishing/Deploy the application, Versioning, signing Application	15

Textbooks:

1. How to Build Android Apps with Kotlin: A hands-on guide to developing, testing, and publishing your first apps with Android, Alex Forrester, Packt Publishing, 2021
2. Android Programming: Crafting UI/UX using Kotlin, SYBGEN Learning, 2020

Additional References:

1. **Head First Android Development: A Learner's Guide to Building Android Apps with Kotlin Dawn Griffiths, 3rd Edition, O'Reilly Media, 2021**
2. **Android Studio 4.2 Development Essentials - Kotlin Edition: Developing Android Apps Using Android Studio 4.2, Kotlin and Android Jetpack, Neil Smyth, Payload Media, 2021**
3. **Android Programming with Kotlin for Beginners, John Horton, Packt Publishing, 2019**
4. **Android Development with Kotlin: Enhance your skills for Android development using Kotlin, Marcin Moskala, Packt Publishing**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP405	**Android Application Development – Practical**	**1**	**3**
1	i. Write a program using Kotlin to implement control structures and loops. ii. Write a program to implement object-oriented concepts in Kotlin.		
2	i. Create an Android application to design screens using different layouts and UI including Button, Edittext, Textview, Radio Button etc. ii. Write an android application demonstrating response to event/user interaction for a. Checkbox b. Radio button c. Button d. Spinner		
3	i. Create an application to create Image Flipper and Image Gallery. On click on the image display the information about the image. ii. Create an application to use Gridview for shopping cart application.		
4	i. Create an Android application to demonstrate implicit and explicit intents ii. Create an application to demonstrate shared preferences		
5	i. Create an Android application to demonstrate the use of Broadcast listeners. ii. Create an Android application to create and use services.		
6	i. Create an Android application to demonstrate XML based animation ii. Create an Android application to display canvas and allow the user to draw on it.		
7	i. Create a media player application in android that plays audio. Implement play, pause, and loop features. ii. Create an Android application to use a camera and capture image/video and display them on the screen.		
8	i. Create an android application to implement Asynctask and threading concepts. ii. Create an Android application to demonstrate the different types of menus. a. Pop-up Menu b. Context Menu c. Option Menu		
9	Create an Android application to record the current location. Based on the current location allow the user to use some useful services/applications		
10	Create a suitable Android application to store and retrieve data in the SQLite database.		
11	Create a suitable Android application to work with Firebase for storing and manipulating data.		

Course Code	Course Title	Credits	Lectures /Week
USCS406	**Advanced Application Development**	**2**	**3**

About the Course:
The course aims at developing scalable, robust, and maintainable web applications using MEAN stack and developing advanced mobile applications using Flutter

Course Objectives:

- To understand all the necessary and important technologies such as MongoDB, Express.js, AngularJS, and Node.js.
- To understand modern app development using Flutter

Learning Outcomes:
After successful completion of this course, students would be able to

- Store the data in NoSQL, document-oriented MongoDB database that brings performance and scalability.
- Use Node.js and Express Framework for building fast, scalable network applications
- Use AngularJS framework that offers declarative, two-way data binding for web applications.
- Integrate the front-end and back-end components of the MEAN stack.
- Develop robust mobile applications using Flutter.

Unit	Topics	No of Lectures
I	**Node.js (N):** Introduction to Node.js. Installing Node.js. The package.json File. The Node.js Event Loop. The I/O Cycle. The Anatomy of a Node.js Module. Creating Node Modules. Exploring the Node.js HTTP Module. Creating an HTTP Webserver with Node.js. Responding to HTTP Requests. Routing in Node.js. Creating a Sample Node.js Application. **MongoDB(M):** Introduction to MongoDB. Installing MongoDB. Using MongoDB Compass. Using Mongo Shell Interface. Connecting to MongoDB. Creating Schemas and Models. Querying Documents Using find(). Inserting Documents Using create(). Updating Documents Using findOneAndUpdate(). Deleting Documents Using findOneAndDelete() &deleteMany()	**15**
II	**Server-Side Development with Express (E):** Introduction to the Express Framework. Installing and Testing Express. Creating a Node.js Express App. Restructuring an Express App. Creating Templates. Using Express Middleware Functions. Creating the List Page. Creating the Details Page. Creating the Edit Page. Creating the Add Page. Deleting Data. REST API Basics. Testing REST APIs. Refactoring APIs. **Understanding Angular.JS(A):** Getting Started with Angular. Creating an Angular Application. Angular Project File Structure. Anatomy of an Angular Component. One-way Data Binding. Two-way Data Binding.	**15**

	Using NglfDirective. Using NgForOf Directive. Angular Modules. Creating NgModulesUsing Angular Router. Configuring Templates. Creating Navigations. Working with Template-driven Forms. Working with Reactive Forms. Validating Form Data. Services Dependency Injection (DI). Reading Data from Database. Inserting Data into Database. Updating Data in the Database. Delete Data from Database.	
III	**Understanding Flutter:** Importance of Flutter, Flutter Framework, Android Studio, Flutter SDK, Installing and Configuring Flutter SDK. **Dart Programming**: main() function, Dart Variables, Dart Data Types, Dart Conditional Operators, Control Flow & Loops. Dart Functions - Functions, Function Structure, creating a Function, Function Returning Expression. Object-Oriented Programming (OOP) - Creating a Class, Adding Methods to Classes, Class — Getters and Setters, Class Inheritance, Abstract Class. **Flutter Widgets Fundamentals**: Scaffold Widget, Image Widget, Container Widget, Column and Row Widgets, Icon Widget, Layouts in Flutter, Card Widget, Hot Reload and Hot Restart, Stateful and Stateless Widgets **Navigation and Routing:** Button Widget, App Structure and Navigation, Navigate to a New Screen and Back, Navigate with Named Routes, Send and Return Data among Screens, Animate a Widget across Screens, WebView Widget in Flutter	**15**

Textbooks:

1. Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications by Brad Dayley, Brendan Dayley, Caleb Dayley, Pearson, 2018.
2. Beginning Flutter: A Hands On Guide to App Development by Marco L. Napoli, Wrox, 2019

Additional References:

1. **Full Stack Javascript Development with Mean - MongoDB, Express, AngularJS, and Node.JS by Adam Bretz, Colin J Ihrig, Shroff/SitePoint, 2015**
2. **Practical Flutter by Zammetti Frank, Apress, 2019**

Course Code	**Course Title**	**Credits**	**Lectures /Week**
USCSP406	**Advanced Application Development – Practical**	**1**	**3**
1	Write a program to implement MongoDB data models		
2	Write a program to implement CRUD operations on MongoDB		
3	Write a program to perform validation of a form using AngularJS		
4	Write a program to create and implement modules and controllers in Angular JS		
5	Write a program to implement Error Handling in Angular JS		
6	Create an application for Customer / Students records using AngularJS		
7	Write a program to create a simple web application using Express, Node JS and Angular JS		
8	Create a simple HTML “Hello World” Project using AngularJS Framework and apply ng-controller, ng-model and expressions		
9	Create an app using Flutter for User Authentication		
10	Create an app using Flutter to implement an Image Gallery		
11	Create an app using Flutter to demonstrate the use of different layouts		
12	Create an app using Flutter to demonstrate navigation in an App		

<table>
<tr><th>Course Code</th><th>Course Title</th><th>Credits</th><th>Lectures /Week</th></tr>
<tr><td><b>USCS4071</b></td><td><b>Research Methodology</b></td><td><b>2</b></td><td><b>3</b></td></tr>
<tr><td colspan="4"></td></tr>
<tr><td colspan="4"><b>About the Course:</b>
The course aims to understand the basics research, how research problems are defined, research methods are adopted and/or developed, research is undertaken, and how research results are communicated to the peers.</td></tr>
<tr><td colspan="4"><b>Course Objectives:</b>
● The research methodology course is proposed to assist students in planning and carrying out research projects.
● The students are exposed to the principles, procedures and techniques of implementing research project.
● The course starts with an introduction to research and carries through the various methodologies involved.
● It continues with finding out the literature using technology, basic statistics required for research and finally report writing.</td></tr>
<tr><td colspan="4"><b>Learning Outcomes:</b>
After successful completion of this course, students would be able to
● Define research, formulate problem and describe the research process and research methods.
● Understand and apply basic research methods including research design, data analysis and interpretation.
● Understand ethical issues in research, write research report, research paper and publish the paper.</td></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Unit</b></td><td colspan="2"><b>Topics</b></td><td><b>No of Lectures</b></td></tr>
<tr><td><b>I</b></td><td colspan="2"><b>Introduction to Research Methodology:</b> Meaning of Research, Objectives of Research, Motivations in Research, types of Research, Research Approaches, Significance of Research, Research Methods v/s Methodology, Research and Scientific Methods, Research Process, Criteria of Good Research.
<b>Defining the Research Problem:</b> Concept and need, Identification of Research problem, defining and delimiting Research problem.
<b>Formulating a Research Problem:</b> Reviewing Literature, formulating a Research Problem, Research Question, Identifying Variables, Constructing Hypothesis
<b>The Research Design</b>: Meaning, Need for Research Design, Important Concepts, Different Research Designs, Basic Principles of Experimental Designs.</td><td><b>15</b></td></tr>
<tr><td><b>II</b></td><td colspan="2"><b>Tools for Data Collection</b>: Collections of Primary Data, Collection of Data</td><td><b>15</b></td></tr>
</table>

	through questionnaire and Schedules, other Observation Interview Methods, Collection of Secondary Data, Selection of appropriate method for data collection, Case Study, Focus Group Discussion, Techniques of developing research tools, viz. Questionnaire and rating scales etc. Reliability and validity of Research tools. **Sampling Design:** Steps in Sampling Design, Criteria of Selecting a Sampling Procedure, Characteristics of a Good Sample Design, Different Types of Sample Designs, how to Select a Random Sample. Probability and Non-Probability sampling types and criteria for selection, Developing sampling Frames. **Overview of Hypothesis Testing:** What is a Hypothesis?Characteristics of good Hypothesis. Basic Concepts, Procedure for Hypothesis Testing, Flow Diagram for Hypothesis Testing,Tests of Hypotheses, and One sided and two-sided hypothesis,Type – I and Type – II errors, Null Hypothesis-Alternative Hypothesis.	
III	**Technical Writing:** Writing a Research Proposal, what is a Scientific Paper? Ethics in Scientific Publishing. **Preparing the Text:** How to Prepare the Title, how to List the Authors and Addresses, how to Prepare the Abstract, how to Write the Introduction, how to Write the Materials and Methods Section, how to Write the Results, how to Write the Discussion, how to State the Acknowledgments, how to Cite the References. **Preparing the Tables and Figures:** How to Design Effective Tables, how to Prepare Effective Graphs, how to Prepare Effective Photographs. **Publishing the Paper:** Rights and Permissions, How to Submit the Manuscript, How and When to Use Abbreviations, How to Write a thesis, Outcome of Research, Ethical issues in research	**15**

Textbooks:

1. Kothari C.R., Research Methodology, New Age International Publication, 2019
2. Research Methodology-A Step-by-Step Guide for Beginners, (4th ed.), Ranjit Kumar, Singapore, Pearson Education, 2018
3. Research Methodology, VaishaliKhairnar, Staredu Solutions India Pvt Ltd, 2020

Additional References:

1. **Research Methodology: Methods and Techniques, Dr. R. K. Jain, , Fifth Edition, VEI, 2021**
2. **Research Methodology, R. Panneerselvam, Second Edition, PHI, 2014**
3. **Dr. Rachna Jain, Research Methodology, Maximax Publishing House**
4. **How to Write and Publish a Scientific Paper, Cambridge University Press, Barbara Gastel and Robert A. Day, 2017**

Course Code	Course Title	Credits	Lectures /Week
USCS4072	**Management & Entrepreneurship**	**2**	**3**

About the Course:
The aim of the course is to develop conceptual understanding of management and administration, and comprehend the environment of making of an entrepreneur. The course focuses on giving students the business management and innovation skills required to succeed in a startup

Course Objectives:

- To understand the idea of management, process and its levels.
- To understand the perception of entrepreneurship, process and its types.
- To understand the concept SSI and steps to start SSI.
- To understand the selection of project, project report, project appraisal, and its feasibility.

Learning Outcomes:
After successful completion of this course, students would be able to

- Understand the meaning of management, functions, administration and its process.
- Understand the foundation of entrepreneurship and its theory, types and its process.
- Identify the steps involved in an entrepreneurial venture (SSI).
- Understand an entrepreneur is converting his business ideas into running concern by selecting the project.

Unit	Topics	No of Lectures
I	**Introduction: Meaning,** Meaning, Characteristics of Management, Nature of Management, Management Functions, Functional Areas of Management, Management and Administration, Role of Management, Levels of Management, Evolution of Management **Planning:** Meaning, Nature, importance, types of planning, types of plans, planning process, decision-making. **Organizing and staffing:** Meaning and Definitions of Organizing, Steps in Organizing, Nature of Organization, Organization Structure, Purpose of Organization, Principles of Organization, Departmentation, Types of Organization, Span of Control, Authority, Power and Responsibility, Delegation of Authority, Centralization and Decentralization, Delegation vs Decentralization, Management by Objectives [MBO], Meaning of Staffing, Nature and Importance of Staffing, Recruitment, Selection.	**15**
II	**Directing and Controlling:** Meaning and Nature of Direction, Principles of Directing, Leadership and Leadership Styles, Motivation, Communication, Noise and Feedback in Communication, Importance of Communication, Channels of Communication, Types of Communication, Forms of Communication, Coordination, Coordination and Cooperation, Importance of Coordination, Techniques of Coordination, Managerial	**15**

	Control, Steps in a Control Process, Essentials of a Sound Control System, Control Methods. **Entrepreneurship:** Evolution of Concept of Entrepreneur, Concept of Entrepreneur, Characteristics of Entrepreneur, Distinction between Entrepreneur and Manager, Technical Entrepreneur, Charms of Being an Entrepreneur, Functions of an Entrepreneur, Types of Entrepreneurs, Intrapreneurs, Ultrapreneurs, Concept of Entrepreneurship, Evolution of Entrepreneurship, Role of Entrepreneurship in Economic Development, Stages in the Entrepreneurial Process, Barriers to Entrepreneurship	
III	**Small Scale Industry:** Meaning and Definition of Small-Scale Industry, Characteristics of SSI, Objectives, Scope, Role of SSI in Economic Development, Advantages of Small-Scale Industries, steps to Start an SSI,Government Policy towards SSI **Preparation of Project:** Meaning, Project Classification, Project Identification, Project Report and its significance, Contents of a Project Report, Formulation of Project Report, Planning Commission Guidelines, Network Analysis, Common Mistakes by Entrepreneurs in Project Formulation, Project Appraisal, Identification of Opportunity, Project Feasibility study.	**15**

Textbooks:

1. HavinalVeerabhadrappa, Management and Entrepreneurship, New Age International Publishers.
2. KanishkaBedi, Management and Entrepreneurship, Oxford University Press
3. Dr. R. K. Singal, Entrepreneurship Development and Management

Additional References:

1. **P. N. Singh, J. C. Saboo, Entrepreneurship Management, 6th Edition, Dr. P. N. Singh Centre for Hrd Publications.**
2. **Donald L. Sexton & Raymond W. Smilor, The Art and Science of Entrepreneurship, Ballinger, 2022**
3. **Clifford M.Baumback& Joseph R.Mancuso, Entrepreneurship And Venture Management, Prentice Hall**

EvaluationScheme

I. Internal Evaluation for Theory Courses – 25 Marks

(i) Mid-Term Class Test– 15Marks

- It should be conducted using any **learning management system** such as **Moodle**(Modularobject-orienteddynamiclearning environment)
- The test should have **15 MCQ's** which should be solved in a time duration of **30 minutes.**

(ii) Assignment/ Case study/ Presentations– 10 Marks

- Assignment / Case Study Report / Presentation can be uploaded onany**learning management system**.

II. External Examination for Theory Courses – 75 Marks

- Duration: **2.5 Hours**
- Theory question paper pattern:

	All questions are compulsory.		
Question	**Based on**	**Options**	**Marks**
Q.1	Unit I	*Any 4 out of 6*	20
Q.2	Unit II	*Any 4 out of 6*	20
Q.3	Unit III	*Any 4 out of 6*	20
Q.4	Unit I,II and III	*Any 5 out of 6*	15

- All questions shall be compulsorywith internal choicewithinthequestions.
- Each Question maybe sub-divided into subquestions as a, b, c, d, etc.&the allocation ofMarksdependsontheweightageofthetopic.

III. Practical Examination

- Each core subjectcarries50 Marks
  40 marks + 05 marks (journal) + 05 marks (viva)
- Duration: **2 Hours**for each practical course.
- Minimum **80% practical** from each core subjects are required to be completed.
- **Certified Journal is compulsory for appearing at the time of Practical Exam**
- The final submission and evaluation of **journal in electronic form** using a Learning Management System / Platform can be promoted by college.

UNIVERSITY OF MUMBAI

No. UG/63 of 2018-19

CIRCULAR:-

Attention of the Principals of the affiliated Colleges and Directors of the recognized Institutions in Science & Technology Faculty is invited to this office Circular Nos. UG/108 of 2017-18, dated 27th July, 2017 relating to syllabus of the Bachelor of Science (B.Sc.) degree course.

They are hereby informed that the recommendations made by the Ad-hoc Board of Studies in Computer Science at its meeting held on 10th May, 2018 have been accepted by the Academic Council at its meeting held on 14th June, 2018 **vide** item No. 4.40 and that in accordance therewith, the revised syllabus as per the (CBCS) for the T.Y.B.Sc. in Computer Science (Sem - V & VI), has been brought into force with effect from the academic year 2018-19, accordingly. (The same is available on the University's website www.mu.ac.in).

MUMBAI – 400 032
6th July, 2018

(Dr. Dinesh Kamble)
I/c REGISTRAR

To

The Principals of the affiliated Colleges & Directors of the recognized Institutions in Science & Technology Faculty. (Circular No. UG/334 of 2017-18 dated 9th January, 2018.)

A.C./4.40/14/06/2018

****

No. UG/ 63 -A of 2018 MUMBAI-400 032 6th July, 2018

Copy forwarded with Compliments for information to:-

1) The I/c Dean, Faculty of Science & Technology,
2) The Chairman, Ad-hoc Board of Studies in Computer Science,
3) The Director, Board of Examinations and Evaluation,
4) The Director, Board of Students Development,
5) The Co-Ordinator, University Computerization Centre,

(Dr. Dinesh Kamble)
I/c REGISTRAR

Academic Council
Item No: ________________

UNIVERSITY OF MUMBAI

Syllabus for SemV&VI
Program: Bachelor of Science
Course: Computer Science

Credit Based Semester and Grading System with effect from
Academic Year 2018-2019

Preamble

This is the third year curriculum in the subject of Computer Science. The revised structure is designed to transform students into technically competent, socially responsible and ethical Computer Science professionals. In these Semesters we have made the advancements in the subject based on the previous Semesters Knowledge.

In the first year basic foundation of important skills required for software development is laid. Second year of this course is about studying core computer science subjects. The third year is the further advancement which covers developing capabilities to design formulations of computing models and its applications in diverse areas.

The proposed curriculum contains two semesters, each Semester contains two Electives: Elective-I and II. Every Elective contains three papers based on specific areas of Computer Science. It also includes one Skill Enhancement paper per semester, helps the student to evaluate his/her computer science domain specific skills and also to meet industry expectations. This revised curriculum has not only taken the specific areas of computer science into consideration but will also give the opportunity to the student to prove his/her ability in the subject practically through the Project Implementation. In Semester V and Semester VI student has to undertake a Project. It can boost his/her confidence and also can encourage the student to perform innovations in the subject as the choice of the Project topic is kept open covering most of the areas of Computer Science subject as per the students interest and the subject they have learned during the Course.

Proposed Curriculum contains challenging and varied subjects aligned with the current trend with the introduction of Machine Intelligence specific subject such as Artificial Intelligence, Information Retrieval. Data Management related subjects such as Cloud Computing and Data Science. Image processing topics such as Game Programming, Digital Image Processing. Introduction of physical world through Architecting of IoT and Wireless Sensor Networks and Mobile Communication. Security domain is also evolved by the introduction of Ethical Hacking, Cyber Forensic and Information and Network Security. To get the hands on experience Linux Server Administration and Web Services topics are included.

In essence, the objective of this syllabus is to create a pool of technologically savvy, theoretically strong, innovatively skilled and ethically responsible generation of computer science professionals. Hope that the teacher and student community of University of Mumbai will accept and appreciate the efforts.

T.Y.B.Sc. (Semester V and VI)
Computer Science Syllabus
Credit Based Semester and Grading System
To be implemented from the Academic year 2018-2019

SEMESTER V			
Course	**TOPICS**	**Credits**	**L / Week**
	Elective-I (Select Any Two)		
USCS501	Artificial Intelligence	3	3
USCS502	Linux Server Administration	3	3
USCS503	Software Testing and Quality Assurance	3	3
	Elective-II (Select Any Two)		
USCS504	Information and Network Security	3	3
USCS505	Architecting of IoT	3	3
USCS506	Web Services	3	3
	Skill Enhancement		
USCS507	Game Programming	2	3
	Practical		
USCSP501	Practical of Elective-I	2	6
USCSP502	Practical of Elective-II	2	6
USCSP503	Project Implementation	1	3
USCSP504	Practical of Skill Enhancement : USCS507	1	3

SEMESTER VI			
Course	**TOPICS**	**Credits**	**L / Week**
	Elective-I (Select Any Two)		
USCS601	Wireless Sensor Networks and Mobile Communication	3	3
USCS602	Cloud Computing	3	3
USCS603	Cyber Forensics	3	3
	Elective-II (Select Any Two)		

USCS604	Information Retrieval	3	3
USCS605	Digital Image Processing	3	3
USCS606	Data Science	3	3
	Skill Enhancement		
USCS607	Ethical Hacking	2	3
	Practical		
USCSP601	Practical of Elective-I	2	6
USCSP602	Practical of Elective-II	2	6
USCSP603	Project Implementation	1	3
USCSP604	Practical of Skill Enhancement : USCS607	1	3

SEMESTER V

THEORY

<table>
<tr><td><b>Course:</b>
<b>USCS501</b></td><td colspan="2"><b>TOPICS (Credits : 03 Lectures/Week:03)</b>
<b>Artificial Intelligence</b></td></tr>
<tr><td colspan="3"><b>Objectives:</b>
Artificial Intelligence (AI) and accompanying tools and techniques bring transformational changes in the world. Machines capability to match, and sometimes even surpass human capability, make AI a hot topic in Computer Science. This course aims to introduce the learner to this interesting area.
<b>Expected Learning Outcomes:</b>
After completion of this course, learner should get a clear understanding of AI and different search algorithms used for solving problems. The learner should also get acquainted with different learning algorithms and models used in machine learning.</td></tr>
<tr><td><b>Unit I</b></td><td><b>What Is AI:</b> Foundations, History and State of the Art of AI.
<b>Intelligent Agents:</b> Agents and Environments, Nature of Environments, Structure of Agents.
<b>Problem Solving by searching:</b> Problem-Solving Agents, Example Problems, Searching for Solutions, Uninformed Search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.</td><td><b>15L</b></td></tr>
<tr><td><b>Unit II</b></td><td><b>Learning from Examples:</b> Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, Theory of Learning, Regression and Classification with Linear Models, Artificial Neural Networks, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning</td><td><b>15L</b></td></tr>
</table>

Unit III	**Learning probabilistic models:** Statistical Learning, Learning with Complete Data, Learning with Hidden Variables: The EM Algorithm. Reinforcement learning: Passive Reinforcement Learning, Active Reinforcement Learning, Generalization in Reinforcement Learning, Policy Search, Applications of Reinforcement Learning.	**15L**

Textbook(s):

1) Artificial Intelligence: A Modern Approach, Stuart Russell and Peter Norvig,3rd Edition, Pearson, 2010.

Additional Reference(s):

1) Artificial Intelligence: Foundations of Computational Agents, David L Poole,Alan K. Mackworth, 2nd Edition, Cambridge University Press ,2017.
2) Artificial Intelligence, Kevin Knight and Elaine Rich, 3rd Edition, 2017
3) The Elements of Statistical Learning, Trevor Hastie, Robert Tibshirani and Jerome Friedman, Springer, 2013

Course: **USCS502**	**TOPICS (Credits : 03 Lectures/Week:03)** **Linux Server Administration**

Objectives:

Demonstrate proficiency with the Linux command line interface, directory & file management techniques, file system organization, and tools commonly found on most Linux distributions. Effectively operate a Linux system inside of a network environment to integrate with existing service solutions. Demonstrate the ability to troubleshoot challenging technical problems typically encountered when operating and administering Linux systems.

Expected Learning Outcomes:

Learner will be able to develop Linux based systems and maintain. Learner will be able to install appropriate service on Linux server as per requirement. Learner will have proficiency in Linux server administration.

Unit I	**Introduction:** Technical Summary of Linux Distributions, Managing Software **Single-Host Administration:** Managing Users and Groups, Booting and shutting down processes, File Systems, Core System Services, Process of configuring, compiling, Linux Kernel **Networking and Security:** TCP/IP for System Administrators, basic network Configuration, Linux Firewall (Netfilter), System and network security	**15L**
Unit II	**Internet Services:** Domain Name System (DNS), File Transfer Protocol (FTP), Apache web server, Simple Mail Transfer Protocol (SMTP), Post Office Protocol and Internet Mail Access Protocol (POP and IMAP), Secure Shell (SSH), Network Authentication, OpenLDAP Server, Samba and LDAP, Network authentication system (Kerberos), Domain Name Service (DNS), Security	**15L**
Unit III	**Intranet Services:** Network File System (NFS), Samba, Distributed File Systems (DFS), Network Information Service (NIS), Lightweight Directory Access Protocol (LDAP), Dynamic Host Configuration Protocol (DHCP), MySQL, LAMP Applications File Servers, Email Services, Chat Applications, Virtual Private Networking.	**15L**

Textbook(s):

1) Linux Administration: A Beginner’s Guide, Wale Soyinka, Seventh Edition, McGraw-Hill Education, 2016
2) Ubuntu Server Guide, Ubuntu Documentation Team, 2016

Additional Reference(s):

1) Mastering Ubuntu Server, Jay LaCroix, PACKT Publisher, 2016

Course: **USCS503**	**TOPICS (Credits : 03 Lectures/Week:03)** **Software Testing and Quality Assurance**	
Objectives: To provide learner with knowledge in Software Testing techniques. To understand how testing methods can be used as an effective tools in providing quality assurance concerning for software. To provide skills to design test case plan for testing software **Expected Learning Outcomes:** Understand various software testing methods and strategies. Understand a variety of software metrics, and identify defects and managing those defects for improvement in quality for given software. Design SQA activities, SQA strategy, formal technical review report for software quality control and assurance.		
Unit I	**Software Testing and Introduction to quality :** Introduction, Nature of errors, an example for Testing, Definition of Quality , QA, QC, QM and SQA , Software Development Life Cycle, Software Quality Factors **Verification and Validation :** Definition of V &V , Different types of V & V Mechanisms, Concepts of Software Reviews, Inspection and Walkthrough **Software Testing Techniques :** Testing Fundamentals, Test Case Design, White Box Testing and its types, Black Box Testing and its types	**15L**
Unit II	**Software Testing Strategies :** Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing **Software Metrics :** Concept and Developing Metrics, Different types of Metrics, Complexity metrics **Defect Management:** Definition of Defects, Defect Management Process, Defect Reporting, Metrics Related to Defects, Using Defects for Process Improvement.	**15L**
Unit III	**Software Quality Assurance :** Quality Concepts, Quality Movement, Background Issues, SQA activities, Software Reviews, Formal Technical Reviews, Formal approaches to SQA, Statistical Quality Assurance, Software Reliability, The ISO 9000 Quality Standards, , SQA Plan , Six sigma, Informal Reviews	**15L**

	Quality Improvement : Introduction, Pareto Diagrams, Cause-effect Diagrams, Scatter Diagrams, Run charts **Quality Costs :** Defining Quality Costs, Types of Quality Costs, Quality Cost Measurement, Utilizing Quality Costs for Decision-Making	

Textbook(s):

1. Software Engineering for Students, A Programming Approach, Douglas Bell, 4th Edition,, Pearson Education, 2005
2. Software Engineering – A Practitioners Approach, Roger S. Pressman, 5th Edition, Tata McGraw Hill, 2001
3. Quality Management, Donna C. S. Summers, 5th Edition, Prentice-Hall, 2010.
4. Total Quality Management, Dale H. Besterfield, 3rd Edition, Prentice Hall, 2003.

Additional Reference(s):

1. Software engineering: An Engineering approach, J.F. Peters, W. Pedrycz , John Wiley,2004
2. Software Testing and Quality Assurance Theory and Practice, Kshirsagar Naik, Priyadarshi Tripathy , John Wiley & Sons, Inc. , Publication, 2008
3. Software Engineering and Testing, B. B. Agarwal, S. P. Tayal, M. Gupta, Jones and Bartlett Publishers, 2010

Course: **USCS504**	**TOPICS (Credits : 03 Lectures/Week:03)** **Information and Network Security**

Objectives:

To provide students with knowledge of basic concepts of computer security including network security and cryptography.

Expected Learning Outcomes:

Understand the principles and practices of cryptographic techniques. Understand a variety of generic security threats and vulnerabilities, and identify & analyze particular security problems for a given application. Understand various protocols for network security to protect against the threats in a network

Unit I	**Introduction:** Security Trends, The OSI Security Architecture, Security Attacks, Security Services, Security Mechanisms **Classical Encryption Techniques:** Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Steganography, Block Cipher Principles, The Data Encryption Standard, The Strength of DES, AES (round details not expected), Multiple Encryption and Triple DES, Block Cipher Modes of Operation, Stream Ciphers **Public-Key Cryptography and RSA:** Principles of Public-Key Cryptosystems, The RSA Algorithm	**15L**
Unit II	**Key Management:** Public-Key Cryptosystems, Key Management, Diffie-Hellman Key Exchange **Message Authentication and Hash Functions**: Authentication Requirements, Authentication Functions, Message Authentication Codes, Hash Functions, Security of Hash Functions and Macs, Secure Hash Algorithm, HMAC **Digital Signatures and Authentication:** Digital Signatures, Authentication Protocols, Digital Signature Standard **Authentication Applications:** Kerberos, X.509 Authentication, Public-Key Infrastructure	**15L**
Unit III	**Electronic Mail Security:** Pretty Good Privacy, S/MIME **IP Security:** Overview, Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations, Key Management **Web Security:** Web Security Considerations, Secure Socket Layer and Transport Layer Security, Secure Electronic Transaction **Intrusion:** Intruders, Intrusion Techniques, Intrusion Detection **Malicious Software:** Viruses and Related Threats, Virus Countermeasures, DDOS **Firewalls:** Firewall Design Principles, Types of Firewalls	**15L**
Textbook(s): 1) Cryptography and Network Security: Principles and Practice 5th Edition, William		

Stallings, Pearson,2010

Additional Reference(s):

1) Cryptography and Network Security, Atul Kahate, Tata McGraw-Hill, 2013.
2) Cryptography and Network, Behrouz A Fourouzan, Debdeep Mukhopadhyay, 2nd Edition,TMH,2011

<table>
<tr><th><b>Course</b>:
<b>USCS505</b></th><th>TOPICS (Credits : 03 Lectures/Week:03)
Architecting of IoT</th><th></th></tr>
<tr><td colspan="3"><b>Objectives:</b>
Discovering the interconnection and integration of the physical world. Learner should get knowledge of the architecture of IoT.
<b>Expected Learning Outcomes:</b>
Learners are able to design & develop IoT Devices. They should also be aware of the evolving world of M2M Communications and IoT analytics.</td></tr>
<tr><td><b>Unit I</b></td><td><b>IoT-An Architectural Overview</b>: Building architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.
<b>IoT Architecture-State of the Art</b> : Introduction, State of the art, Reference Model and architecture, IoT reference Model - IoT Reference Architecture Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views</td><td><b>15L</b></td></tr>
<tr><td><b>Unit II</b></td><td><b>IoT Data Link Layer and Network Layer Protocols:</b>
PHY/MAC Layer(3GPP MTC, IEEE 802.11, IEEE 802.15), Wireless HART,Z-Wave, Bluetooth Low Energy, Zigbee Smart Energy DASH7
<b>Network Layer</b>:IPv4, IPv6, 6LoWPAN, 6TiSCH,ND, DHCP, ICMP, RPL, CORPL, CARP</td><td><b>15L</b></td></tr>
</table>

Unit III	**Transport layer protocols :** Transport Layer (TCP, MPTCP, UDP, DCCP, SCTP)-(TLS, DTLS) **Session layer:** Session Layer-HTTP, CoAP, XMPP, AMQP, MQTT **Service layer protocols:** Service Layer -oneM2M, ETSI M2M, OMA, BBF	**15L**

Textbook(s):

1. From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence, Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle,1st Edition, Academic Press, 2014.
2. Learning Internet of Things, Peter Waher, PACKT publishing, BIRMINGHAM – MUMBAI,2015

Additional References(s):

1. Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications, Daniel Minoli, Wiley Publications,2013
2. Internet of Things (A Hands-onApproach), Vijay Madisetti and ArshdeepBahga,1st Edition, VPT, 2014.
3. http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html

Course: **USCS506**	**TOPICS (Credits : 03 Lectures/Week:03)** **Web Services**

Objectives:

To understand the details of web services technologies like SOAP, WSDL, and UDDI. To learn how to implement and deploy web service client and server. To understand the design principles and application of SOAP and REST based web services (JAX-Ws and JAX-RS).To understand WCF service. To design secure web services and QoS of Web Services

Expected Learning Outcomes:

Emphasis on SOAP based web services and associated standards such as WSDL. Design SOAP based / RESTful / WCF services Deal with Security and QoS issues of Web Services

Unit I	**Web services basics :** What Are Web Services? Types of Web Services Distributed computing infrastructure, overview of XML, SOAP, Building Web Services with JAX-WS, Registering and Discovering Web Services, Service Oriented Architecture, Web Services Development Life Cycle, Developing and consuming simple Web Services across platform	**15L**
Unit II	**The REST Architectural style :** Introducing HTTP, The core architectural elements of a RESTful system, Description and discovery of RESTful web services, Java tools and frameworks for building RESTful web services, JSON message format and tools and frameworks around JSON, Build RESTful web services with JAX-RS APIs, The Description and Discovery of RESTful Web Services, Design guidelines for building RESTful web services, Secure RESTful web services	**15L**
Unit III	**Developing Service-Oriented Applications with WCF :** What Is Windows Communication Foundation, Fundamental Windows Communication Foundation Concepts, Windows Communication Foundation Architecture, WCF and .NET Framework Client Profile, Basic WCF Programming, WCF Feature Details. Web Service QoS	**15L**

Textbook(s):

1) Web Services: Principles and Technology, Michael P. Papazoglou, Pearson Education Limited, 2008
2) RESTful Java Web Services, Jobinesh Purushothaman, PACKT Publishing,2nd Edition, 2015
3) Developing Service-Oriented Applications with WCF, Microsoft, 2017 https://docs.microsoft.com/en-us/dotnet/framework/wcf/index

Additional Reference(s):

1) Leonard Richardson and Sam Ruby, RESTful Web Services, O'Reilly, 2007
2) The Java EE 6Tutorial, Oracle, 2013

Course: **USCS507**	**TOPICS (Credits : 03 Lectures/Week: 03)** **Game Programming**	
Objectives: Learner should get the understanding computer Graphics programming using Directx or Opengl. Along with the VR and AR they should also aware of GPU, newer technologies and programming using most important API for windows. **Expected Learning Outcomes:** Learner should study Graphics and gamming concepts with present working style of developers where everything remains on internet and they need to review it, understand it, be a part of community and learn.		
Unit I	**Mathematics for Computer Graphics, DirectX Kickstart:** **Cartesian Coordinate system:** The Cartesian XY-plane, Function Graphs, Geometric Shapes, Polygonal Shapes, Areas of Shapes, Theorem of Pythagoras in 2D, Coordinates, Theorem of Pythagoras in 3D, 3D Polygons, Euler's Rule **Vectors:** Vector Manipulation, multiplying a Vector by a Scalar, Vector Addition and Subtraction, Position Vectors, Unit Vectors, Cartesian Vectors, Vector Multiplication, Scalar Product, Example of the Dot Product, The Dot Product in Lighting Calculations, The Dot Product in Back-Face Detection, The Vector Product, The Right-Hand Rule, deriving a Unit Normal Vector for a Triangle Areas, Calculating 2D Areas **Transformations:** 2D Transformations, Matrices, Homogeneous Coordinates, 3D Transformations, Change of Axes, Direction Cosines, rotating a Point about an Arbitrary Axis, Transforming Vectors, Determinants, Perspective Projection, Interpolation **DirectX:** Understanding GPU and GPU architectures. How they are different from CPU Architectures? Understanding how to solve by GPU?	**15L**

Unit II	**DirectX Pipeline and Programming:** **Introduction To DirectX 11:** COM, Textures and Resources Formats, The swap chain and Page flipping, Depth Buffering, Texture Resource Views, Multisampling Theory and MS in Direct3D, Feature Levels **Direct3D 11 Rendering Pipeline:** Overview, Input Assembler Stage (IA), Vertex Shader Stage (VS), The Tessellation Stage (TS), Geometry Shader Stage (GS), Pixel Shader Stage (PS), Output merger Stage (OM) Understanding Meshes or Objects, Texturing, Lighting, Blending. **Interpolation and Character Animation:** **Trigonometry:** The Trigonometric Ratios, Inverse Trigonometric Ratios, Trigonometric Relationships, The Sine Rule, The Cosine Rule, Compound Angles, Perimeter Relationships **Interpolation**: Linear Interpolant, Non-Linear Interpolation, Trigonometric Interpolation, Cubic Interpolation, Interpolating Vectors, Interpolating Quaternions **Curves:** Circle, Bezier, B-Splines **Analytic Geometry:** Review of Geometry, 2D Analytic Geometry, Intersection Points, Point in Triangle, and Intersection of circle with straight line.	**15L**
Unit III	**Introduction to Rendering Engines:** Understanding the current market Rendering Engines. Understanding AR, VR and MR.Depth Mappers, Mobile Phones, Smart Glasses, HMD’s **Unity Engine: Multi-platform publishing, VR + AR:** Introduction and working in Unity, 2D, Graphics, Physics, Scripting, Animation, Timeline, Multiplayer and Networking, UI, Navigation and Pathfinding, XR, Publishing. **Scripting:** Scripting Overview, Scripting Tools and Event Overview **XR:** VR, AR, MR, Conceptual Differences. SDK, Devices	**15L**
Text Book(s): 1) Mathematics for Computer Graphics, John Vince, Springer-Verlag London, 5th Edition,2017 2) Mathematics for 3D Game Programming and Computer Graphic, Eric Lengyel, Delmar		

Cengage Learning, Delmar Cengage Learning,2011

3) Introduction To 3D Game Programming With Directx® 11,Frank D Luna, Mercury Learning And Information,2012.
4) https://docs.unity3d.com/Manual/index.html - Free

Additional Reference(s):

1) Computer Graphics, C Version, Donald Hern and Pauline Baker, Pearson Education, 2nd Edition, 1997
2) HLSL Development Cookbook, Doron Feinstein, PACKT Publishing,2013

Suggested List of Practical- SEMESTER V

Course: **USCSP501**	**(Credits : 02 Lectures/Week: 06)** **Practical of Elective-I**	

USCS501: Artificial Intelligence

Practical shall be implemented in LISP

1. Implement Breadth first search algorithm for Romanian map problem.
2. Implement Iterative deep depth first search for Romanian map problem.
3. Implement A* search algorithm for Romanian map problem.
4. Implement recursive best-first search algorithm for Romanian map problem.
5. Implement decision tree learning algorithm for the restaurant waiting problem.
6. Implement feed forward back propagation neural network learning algorithm for the restaurant waiting problem.
7. Implement Adaboost ensemble learning algorithm for the restaurant waiting problem.
8. Implement Naive Bayes' learning algorithm for the restaurant waiting problem.
9. Implement passive reinforcement learning algorithm based on adaptive dynamic programming (ADP) for the 3 by 4 world problem
10. Implement passive reinforcement learning algorithm based on temporal differences (TD) for 3 by 4 world problem.

USCS502: Linux Server Administration

- Practical shall be performed using any Linux Server (with 8GB RAM).

- Internet connection will be required so that Linux server (command line mode) can be connected to Internet.

1. Install DHCP Server in Ubuntu 16.04
2. Initial settings: Add a User, Network Settings, Change to static IP address, Disable IPv6 if not needed, Configure Services, display the list of services which are running, Stop and turn OFF auto-start setting for a service if you don't need it, Sudo Settings
3. Configure NTP Server (NTPd), Install and Configure NTPd, Configure NTP Client (Ubuntu and Windows)
4. SSH Server : Password Authentication

Configure SSH Server to manage a server from the remote computer, SSH Client : (Ubuntu and Windows)

5. Install DNS Server BIND, Configure DNS server which resolves domain name or IP address, Install BIND 9, Configure BIND, Limit ranges you allow to access if needed.
6. Configure DHCP Server, Configure DHCP (Dynamic Host Configuration Protocol) Server, Configure NFS Server to share directories on your Network, Configure NFS Client. (Ubuntu and Windows Client OS)
7. Configure LDAP Server, Configure LDAP Server in order to share users' accounts in your local networks, Add LDAP User Accounts in the OpenLDAP Server, Configure LDAP Client in order to share users' accounts in your local networks. Install phpLDAPadmin to operate LDAP server via Web browser.
8. Configure NIS Server in order to share users' accounts in your local networks, Configure NIS Client to bind NIS Server.
9. Install MySQL to configure database server, Install phpMyAdmin to operate MySQL on web browser from Clients.
10. Install Samba to share folders or files between Windows and Linux.

USCS503: Software Testing and Quality Assurance

1. Install Selenium IDE; Write a test suite containing minimum 4 test cases for different formats.
2. Conduct a test suite for any two web sites.
3. Install Selenium server (Selenium RC) and demonstrate it using a script in Java/PHP.
4. Write and test a program to login a specific web page.
5. Write and test a program to update 10 student records into table into Excel file
6. Write and test a program to select the number of students who have scored more than 60 in any one subject (or all subjects).
7. Write and test a program to provide total number of objects present / available on the page.
8. Write and test a program to get the number of items in a list / combo box.
9. Write and test a program to count the number of check boxes on the page checked and unchecked count.
10. Load Testing using JMeter, Android Application testing using Appium Tools, Bugzilla Bug tracking tools.

Course: **USCSP502**	**(Credits : 02 Lectures/Week: 06)** **Practical of Elective-II**	

USCS504: Information and Network security

1.Write programs to implement the following Substitution Cipher Techniques:

- Caesar Cipher
- Monoalphabetic Cipher

2 Write programs to implement the following Substitution Cipher Techniques:

- Vernam Cipher
- Playfair Cipher

3 Write programs to implement the following Transposition Cipher Techniques:

- Rail Fence Cipher
- Simple Columnar Technique

4 Write program to encrypt and decrypt strings using

- DES Algorithm
- AES Algorithm

5 Write a program to implement RSA algorithm to perform encryption / decryption of a given string.

6 Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.

7 Write a program to implement the MD5 algorithm compute the message digest.

8 Write a program to calculate HMAC-SHA1 Signature

9 Write a program to implement SSL.

10 Configure Windows Firewall to block:

- A port
- An Program
- A website

USCS505: Architecting of IoT

1. a) Edit text files with nano and cat editor, Learn sudo privileges and Unix shell commands such as cd , ls , cat, etc

b) Learn to set dynamic and static IP. Connect to and Ethernet and WiFi network. Learn to vnc and ssh into a raspberry pi using vnc and putty from a different computer on the network.

c) Write a basic bash script to open programs in kiosk mode. Learn how to autostart programs on boot.

2. Run the node red editor and run simple programs and trigger gpios. Use basic nodes such as inject, debug, gpio
3. Open the python idle editor and run simple Python scripts such as to print Fibonacci numbers, string functions. Learn how to install modules using Pip and write functions
4. Setup a physical button switch and trigger an led in node red and python w debounce
5. Write simple JavaScript functions in Node-Red simple HTTP server page using node red
6. Setup a TCP server and client on a raspberry pi using Python modules to send messages and execute shell commands from within python such as starting another application
7. Trigger a set of led Gpios on the pi via a Python Flask web server
8. Interface the raspberry pi with a 16x2 LCD display and print values.
9. Setup a Mosquitto MQTT server and client and write a Python script to communicate data between Pi's.
10. Interface with an Accelerometer Gyro Mpu6050 on the i2c bus and send sensor values over the internet via mqtt.

USCS506: Web Services

1. Write a program to implement to create a simple web service that converts the temperature from Fahrenheit to Celsius and vice a versa.
2. Write a program to implement the operation can receive request and will return a response in two ways. a) One - Way operation b) Request –Response
3. Write a program to implement business UDDI Registry entry.
4. Develop client which consumes web services developed in different platform.
5. Write a JAX-WS web service to perform the following operations. Define a Servlet / JSP that consumes the web service.
6. Define a web service method that returns the contents of a database in a JSON string. The contents should be displayed in a tabular format.
7. Define a RESTful web service that accepts the details to be stored in a database and performs

CRUD operation.

8. Implement a typical service and a typical client using WCF.
9. Use WCF to create a basic ASP.NET Asynchronous JavaScript and XML (AJAX) service.
10. Demonstrates using the binding attribute of an endpoint element in WCF.

Course: **USCSP503**	**(Credits : 01 Lectures/Week: 03)** **Project Implementation**	

Please Refer to Project Implementation Guidelines

Course: **USCSP504**	**(Credits : 01 Lectures/Week: 03)** **Practical of Skill Enhancement**	

USCS507 : Game Programming

1. Setup DirectX 11, Window Framework and Initialize Direct3D Device
2. Buffers, Shaders and HLSL (Draw a triangle using Direct3D 11)
3. Texturing (Texture the Triangle using Direct 3D 11)
4. Lightning (Programmable Diffuse Lightning using Direct3D 11)
5. Specular Lightning (Programmable Spot Lightning using Direct3D 11)
6. Loading models into DirectX 11 and rendering.

Perform following Practical using online content from the Unity Tutorials Web--sites: https://unity3d.com/learn/tutorials/s/interactive-tutorials

7. https://unity3d.com/learn/tutorials/s/2d-ufo-tutorial
8. https://unity3d.com/learn/tutorials/s/space-shooter-tutorial
9. https://unity3d.com/learn/tutorials/s/roll-ball-tutorial
10. https://unity3d.com/learn/tutorials/topics/vr/introduction?playlist=22946

SEMESTER VI

THEORY

<table>
<tr><td><b>Course:</b>
<b>USCS601</b></td><td><b>TOPICS (Credits : 03 Lectures/Week: 03)</b>
<b>Wireless Sensor Networks and Mobile Communication</b></td><td></td></tr>
<tr><td colspan="3"><b>Objectives:</b>
In this era of wireless and adhoc network, connecting different wireless devices and understanding their compatibility is very important. Information is gathered in many different ways from these devices. Learner should be able to conceptualize and understand the framework. On completion, will be able to have a firm grip over this very important segment of wireless network.
<b>Expected Learning Outcomes:</b>
After completion of this course, learner should be able to list various applications of wireless sensor networks, describe the concepts, protocols, design, implementation and use of wireless sensor networks. Also implement and evaluate new ideas for solving wireless sensor network design issues.</td></tr>
<tr><td><b>Unit I</b></td><td><b>Introduction:</b> Introduction to Sensor Networks, unique constraints and challenges.
Advantage of Sensor Networks, Applications of Sensor Networks, Mobile Adhoc NETworks (MANETs) and Wireless Sensor Networks, Enabling technologies for Wireless Sensor Networks.
<b>Sensor Node Hardware and Network Architecture:</b> Single-node architecture, Hardware components & design constraints, Operating systems and execution environments, introduction to TinyOS and nesC.
Network architecture, Optimization goals and figures of merit, Design principles for WSNs, Service interfaces of WSNs, Gateway concepts.</td><td><b>15L</b></td></tr>
<tr><td><b>Unit II</b></td><td><b>Medium Access Control Protocols:</b> Fundamentals of MAC Protocols, MAC Protocols for WSNs, Sensor-MAC Case Study.
<b>Routing Protocols</b> : Data Dissemination and Gathering, Routing Challenges and Design Issues in Wireless
Sensor Networks, Routing Strategies in Wireless Sensor Networks.
<b>Transport Control Protocols :</b> Traditional Transport Control Protocols,</td><td><b>15L</b></td></tr>
</table>

	Transport Protocol Design Issues, Examples of Existing Transport Control Protocols, Performance of Transport Control Protocols.	
Unit III	**Introduction, Wireless Transmission and Medium Access Control:** Applications, A short history of wireless communication. **Wireless Transmission:** Frequency for radio transmission, Signals, Antennas, Signal propagation, Multiplexing, Modulation, Spread spectrum, Cellular systems. **Telecommunication, Satellite and Broadcast Systems: GSM:** Mobile services, System architecture, Radio interface, Protocols, Localization And Calling, Handover, security, New data services; DECT: System architecture, Protocol architecture; ETRA, UMTS and IMT- 2000. Satellite Systems: History, Applications, Basics: GEO, LEO, MEO; Routing, Localization, Handover.	**15L**

Textbook(s):

1) Protocols and Architectures for Wireless Sensor Network, Holger Kerl, Andreas Willig, John Wiley and Sons, 2005
2) Wireless Sensor Networks Technology, Protocols, and Applications ,Kazem Sohraby, Daniel Minoli and TaiebZnati, John Wiley & Sons, 2007
3) Mobile communications, Jochen Schiller,2nd Edition, Addison wisely , Pearson Education,2012

Additional Reference(s):

1) Fundamentals of Wireless Sensor Networks, Theory and Practice, Waltenegus Dargie, Christian Poellabauer , Wiley Series on wireless Communication and Mobile Computing, 2011
2) Networking Wireless Sensors, Bhaskar Krishnamachari , Cambridge University Press, 2005

<table>
<tr><th><strong>Course</strong>:
<strong>USCS602</strong></th><th><strong>TOPICS (Credits : 03 Lectures/Week: 03)</strong>
<strong>Cloud Computing</strong></th><th></th></tr>
<tr><td colspan="3"><strong>Objectives</strong>:
To provide learners with the comprehensive and in-depth knowledge of Cloud Computing concepts, technologies, architecture, implantations and applications. To expose the learners to frontier areas of Cloud Computing, while providing sufficient foundations to enable further study and research.
<strong>Expected Learning Outcomes:</strong>
After successfully completion of this course, learner should be able to articulate the main concepts, key technologies, strengths, and limitations of cloud computing and the possible applications for state-of-the-art cloud computing using open source technology. Learner should be able to identify the architecture and infrastructure of cloud computing, including SaaS, PaaS, IaaS, public cloud, private cloud, hybrid cloud, etc. They should explain the core issues of cloud computing such as security, privacy, and interoperability.</td></tr>
<tr><td><strong>Unit I</strong></td><td>Introduction to Cloud Computing, Characteristics and benefits of Cloud Computing, Basic concepts of Distributed Systems, Web 2.0, Service-Oriented Computing, Utility-Oriented Computing. Elements of Parallel Computing. Elements of Distributed Computing. Technologies for Distributed Computing. Cloud Computing Architecture. The cloud reference model. Infrastructure as a service. Platform as a service. Software as a service. Types of clouds.</td><td><strong>15L</strong></td></tr>
<tr><td><strong>Unit II</strong></td><td>Characteristics of Virtualized Environments. Taxonomy of Virtualization Techniques. Virtualization and Cloud Computing. Pros and Cons of Virtualization. Virtualization using KVM, Creating virtual machines, oVirt - management tool for virtualization environment. Open challenges of Cloud Computing</td><td><strong>15L</strong></td></tr>
<tr><td><strong>Unit III</strong></td><td>Introduction to OpenStack, OpenStack test-drive, Basic OpenStack operations, OpenStack CLI and APIs, Tenant model operations, Quotas, Private cloud building blocks, Controller deployment, Networking deployment, Block Storage deployment, Compute deployment, deploying and utilizing OpenStack in production environments, Building a production environment, Application orchestration using OpenStack Heat</td><td><strong>15L</strong></td></tr>
</table>

Textbook(s):

1) Mastering Cloud Computing, Rajkumar Buyya, Christian Vecchiola, S Thamarai Selvi, Tata McGraw Hill Education Private Limited, 2013
2) OpenStack in Action, V. K. CODY BUMGARDNER, Manning Publications Co, 2016

Additional Reference(s)*:*

1) OpenStack Essentials, Dan Radez, PACKT Publishing, 2015
2) OpenStack Operations Guide, Tom Fifield, Diane Fleming, Anne Gentle, Lorin Hochstein, Jonathan Proulx, Everett Toews, and Joe Topjian, O'Reilly Media, Inc., 2014
3) https://www.openstack.org

Course: **USCS603**	**TOPICS (Credits :03 Lectures/Week:03)** **Cyber Forensics**	

Objectives:

To understand the procedures for identification, preservation, and extraction of electronic evidence, auditing and investigation of network and host system intrusions, analysis and documentation of information gathered

Expected Learning Outcomes :

The student will be able to plan and prepare for all stages of an investigation - detection, initial response and management interaction, investigate various media to collect evidence, report them in a way that would be acceptable in the court of law.

Unit I	**Computer Forensics :** Introduction to Computer Forensics and standard procedure, Incident Verification and System Identification ,Recovery of Erased and damaged data, Disk Imaging and Preservation, Data Encryption and Compression, Automated Search Techniques, Forensics Software **Network Forensic :** Introduction to Network Forensics and tracking network traffic, Reviewing Network Logs, Network Forensics Tools, Performing Live Acquisitions, Order of Volatility, Standard Procedure **Cell Phone and Mobile Device Forensics:** Overview, Acquisition Procedures for Cell Phones and Mobile Devices	**15L**

Unit II	**Internet Forensic :** **Introduction** to Internet Forensics, World Wide Web Threats, Hacking and Illegal access, Obscene and Incident transmission, Domain Name Ownership Investigation, Reconstructing past internet activities and events **E-mail Forensics :** e-mail analysis, e-mail headers and spoofing, Laws against e-mail Crime, **Messenger Forensics:** Yahoo Messenger **Social Media Forensics:** Social Media Investigations **Browser Forensics:** Cookie Storage and Analysis, Analyzing Cache and temporary internet files, Web browsing activity reconstruction	**15L**
Unit III	**Investigation, Evidence presentation and Legal aspects of Digital Forensics:** Authorization to collect the evidence , Acquisition of Evidence, Authentication of the evidence, Analysis of the evidence, Reporting on the findings, Testimony **Introduction to Legal aspects of Digital Forensics:** Laws & regulations, Information Technology Act, Giving Evidence in court, Case Study – Cyber Crime cases, Case Study – Cyber Crime cases	**15L**

Textbook(s):

1. Guide to computer forensics and investigations, Bill Nelson, Amelia Philips and Christopher Steuart, course technology,5th Edition,2015

Additional Reference(s)*:*

2. Incident Response and computer forensics, Kevin Mandia, Chris Prosise, Tata McGrawHill,2nd Edition,2003

Course: **USCS604**	**TOPICS (Credits : 03 Lectures/Week: 03)** **Information Retrieval**	

Objectives:

To provide an overview of the important issues in classical and web information retrieval. The focus is to give an up-to- date treatment of all aspects of the design and implementation of systems for gathering, indexing, and searching documents and of methods for evaluating systems.

Expected Learning Outcomes:

After completion of this course, learner should get an understanding of the field of information retrieval and its relationship to search engines. It will give the learner an understanding to apply information retrieval models.

Unit I	**Introduction to Information Retrieval:** Introduction, History of IR, Components of IR, and Issues related to IR, Boolean retrieval, Dictionaries and tolerant retrieval.	**15L**
Unit II	**Link Analysis and Specialized Search:** Link Analysis, hubs and authorities, Page Rank and HITS algorithms, Similarity, Hadoop & Map Reduce, Evaluation, Personalized search, Collaborative filtering and content-based recommendation of documents and products, handling “invisible” Web, Snippet generation, Summarization, Question Answering, Cross- Lingual Retrieval.	**15L**
Unit III	**Web Search Engine:** Web search overview, web structure, the user, paid placement, search engine optimization/spam, Web size measurement, search engine optimization/spam, Web Search Architectures. **XML retrieval:** Basic XML concepts, Challenges in XML retrieval, A vector space model for XML retrieval, Evaluation of XML retrieval, Text-centric versus data-centric XML retrieval.	**15L**

Text book(s):

1) Introduction to Information Retrieval, C. Manning, P. Raghavan, and H. Schütze, Cambridge University Press, 2008
2) Modern Information Retrieval: The Concepts and Technology behind Search, Ricardo Baeza -Yates and Berthier Ribeiro – Neto, 2nd Edition, ACM Press Books 2011.
3) Search Engines: Information Retrieval in Practice, Bruce Croft, Donald Metzler and Trevor Strohman, 1st Edition, Pearson, 2009.

Additional Reference(s):

1) Information Retrieval Implementing and Evaluating Search Engines, Stefan Büttcher, Charles L. A. Clarke and Gordon V. Cormack, The MIT Press; Reprint edition (February 12, 2016)

<table>
<tr><td>**Course**:
USCS605</td><td>**TOPICS (Credits : 03 Lectures/Week: 03)**
Digital Image Processing</td><td></td></tr>
<tr><td colspan="3">**Objectives:**
To study two-dimensional Signals and Systems. To understand image fundamentals and transforms necessary for image processing. To study the image enhancement techniques in spatial and frequency domain. To study image segmentation and image compression techniques.
Expected Learning Outcomes:
Learner should review the fundamental concepts of a digital image processing system. Analyze the images in the frequency domain using various transforms. Evaluate the techniques for image enhancement and image segmentation. Apply various compression techniques. They will be familiar with basic image processing techniques for solving real problems.</td></tr>
<tr><td>**Unit I**</td><td>**Introduction to Image-processing System :** Introduction, Image Sampling, Quantization, Resolution, Human Visual Systems, Elements of an Image-processing System, Applications of Digital Image Processing

2D Signals and Systems : 2D signals, separable sequence, periodic sequence, 2D systems, classification of 2D systems, 2D Digital filter

Convolution and Correlation : 2D Convolution through graphical method, Convolution through 2D Z—transform, 2D Convolution through matrix analysis, Circular Convolution, Applications of Circular Convolution, 2D Correlation

Image Transforms: Need for transform, image transforms, Fourier transform, 2D Discrete Fourier Transform, Properties of 2D DFT, Importance of Phase, Walsh transform, Hadamard transform, Haar transform, Slant transform, Discrete Cosine transform, KL transform</td><td>**15L**</td></tr>
<tr><td>**Unit II**</td><td>**Image Enhancement :**Image Enhancement in spatial domain, Enhancement trough Point operations, Histogram manipulation, Linear and nonlinear Gray Level Transformation, local or neighborhood operation, Median Filter, Spatial domain High pass filtering, Bit-plane slicing, Image Enhancement in frequency domain, Homomorphic filter, Zooming operation, Image Arithmetic</td><td>**15L**</td></tr>
</table>

	Binary Image processing : Mathematical morphology, Structuring elements, Morphological image processing, Logical operations, Morphological operations, Dilation and Erosion, Distance Transform **Colour Image processing :**Colour images, Colour Model, Colour image quantization, Histogram of a colour image	
Unit III	**Image Segmentation:** Image segmentation techniques, Region approach, Clustering techniques, Thresholding, Edge-based segmentation, Edge detection, Edge Linking, Hough Transform **Image Compression:** Need for image compression, Redundancy in images, Image-compression scheme, Fundamentals of Information Theory, Run-length coding, Shannon-Fano coding, Huffman Coding, Arithmetic Coding, Transform-based compression, Image-compression standard	**15L**

Textbook(s):

1) Digital Image Processing, S Jayaraman, S Esakkirajan, T Veerakumar,Tata McGraw-Hill Education Pvt. Ltd., 2009

Additional Reference(s)*:*

1) Digital Image Processing 3rd Edition, Rafael C Gonzalez, Richard E Woods, Pearson, 2008
2) Scilab Textbook Companion for Digital Image Processing, S. Jayaraman, S. Esakkirajan And T. Veerakumar, 2016 (https://scilab.in/textbook_companion/generate_book/125)

Course: **USCS606**	**TOPICS (Credits : 03 Lectures/Week: 03)** **Data Science**	

Objectives:

Understanding basic data science concepts. Learning to detect and diagnose common data issues, such as missing values, special values, outliers, inconsistencies, and localization. Making aware of how to address advanced statistical situations, Modeling and Machine Learning.

Expected Learning Outcomes:

After completion of this course, the students should be able to understand & comprehend the problem; and should be able to define suitable statistical method to be adopted.

Unit I	**Introduction to Data Science:** What is Data? Different kinds of data,	**15L**

	Introduction to high level programming language + Integrated Development Environment (IDE), Exploratory Data Analysis (EDA) + Data Visualization, Different types of data sources, **Data Management**: Data Collection, Data cleaning/extraction, Data analysis & Modeling	
Unit II	**Data Curation:** Query languages and Operations to specify and transform data, Structured/schema based systems as users and acquirers of data Semi-structured systems as users and acquirers of data, Unstructured systems in the acquisition and structuring of data, Security and ethical considerations in relation to authenticating and authorizing access to data on remote systems, Software development tools, Large scale data systems, Amazon Web Services (AWS)	**15L**
Unit III	**Statistical Modelling and Machine Learning:** Introduction to model selection: Regularization, bias/variance tradeoff e.g. parsimony, AIC, BIC, Cross validation, Ridge regressions and penalized regression e.g. LASSO **Data transformations**: Dimension reduction, Feature extraction, Smoothing and aggregating **Supervised Learning**: Regression, linear models, Regression trees, Time-series Analysis, Forecasting, Classification: classification trees, Logistic regression, separating hyperplanes, k-NN **Unsupervised Learning**: Principal Components Analysis (PCA), k-means clustering, Hierarchical clustering, Ensemble methods	**15L**

Textbook(s):

1) Doing Data Science, Rachel Schutt and Cathy O'Neil, O'Reilly,2013
2) Mastering Machine Learning with R, Cory Lesmeister, PACKT Publication,2015

Additional Reference(s):

1) Hands-On Programming with R, Garrett Grolemund,1st Edition, 2014
2) An Introduction to Statistical Learning, James, G., Witten, D., Hastie, T., Tibshirani, R.,Springer,2015

Course: **USCS607**	**TOPICS (Credits : 02 Lectures/Week: 03)** **Ethical Hacking**	
Objectives: To understand the ethics, legality, methodologies and techniques of hacking. **Expected Learning Outcomes:** Learner will know to identify security vulnerabilities and weaknesses in the target applications. They will also know to test and exploit systems using various tools and understand the impact of hacking in real time machines.		
Unit I	**Information Security : Attacks and Vulnerabilities** **Introduction to information security** : Asset, Access Control, CIA, Authentication, Authorization, Risk, Threat, Vulnerability, Attack, Attack Surface, Malware, Security-Functionality-Ease of Use Triangle **Types of malware** :Worms, viruses, Trojans, Spyware, Rootkits **Types of vulnerabilities** : OWASP Top 10 : cross-site scripting (XSS), cross site request forgery (CSRF/XSRF), SQL injection, input parameter manipulation, broken authentication, sensitive information disclosure, XML External Entities, Broken access control, Security Misconfiguration, Using components with known vulnerabilities, Insufficient Logging and monitoring, OWASP Mobile Top 10, CVE Database **Types of attacks and their common prevention mechanisms** : Keystroke Logging, Denial of Service (DoS /DDoS), Waterhole attack, brute force, phishing and fake WAP, Eavesdropping, Man-in-the-middle, Session Hijacking, Clickjacking, Cookie Theft, URL Obfuscation, buffer overflow, DNS poisoning, ARP poisoning, Identity Theft, IoT Attacks, BOTs and BOTNETs **Case-studies** : Recent attacks – Yahoo, Adult Friend Finder, eBay, Equifax, WannaCry, Target Stores, Uber, JP Morgan Chase, Bad Rabbit	**15L**
Unit II	**Ethical Hacking – I (Introduction and pre-attack)** **Introduction**: Black Hat vs. Gray Hat vs. White Hat (Ethical) hacking, Why is Ethical hacking needed?, How is Ethical hacking different from security auditing and digital forensics?, Signing NDA, Compliance and Regulatory	**15L**

	concerns, Black box vs. White box vs. Black box, Vulnerability assessment and Penetration Testing. **Approach : Planning -** Threat Modeling, set up security verification standards, Set up security testing plan – When, which systems/apps, understanding functionality, black/gray/white, authenticated vs. unauthenticated, internal vs. external PT, Information gathering, Perform Manual and automated (Tools: WebInspect/Qualys, Nessus, Proxies, Metasploit) VA and PT, How WebInspect/Qualys tools work: Crawling/Spidering, requests forging, pattern matching to known vulnerability database and Analyzing results, Preparing report, Fixing security gaps following the report **Enterprise strategy** : Repeated PT, approval by security testing team, Continuous Application Security Testing, **Phases:** Reconnaissance/foot-printing/Enumeration, **Phases**: Scanning, Sniffing	
Unit III	**Ethical Hacking :Enterprise Security** **Phases : Gaining and Maintaining Access** : **Systems hacking** – Windows and Linux – Metasploit and Kali Linux, Keylogging, Buffer Overflows, Privilege Escalation, Network hacking - ARP Poisoning, Password Cracking, WEP Vulnerabilities, MAC Spoofing, MAC Flooding, IPSpoofing, SYN Flooding, Smurf attack, **Applications hacking** : SMTP/Email-based attacks, VOIP vulnerabilities, Directory traversal, Input Manipulation, Brute force attack, Unsecured login mechanisms, SQL injection, XSS, Mobile apps security, **Malware analysis** : Netcat Trojan, wrapping definition, reverse engineering **Phases :** Covering your tracks : Steganography, Event Logs alteration **Additional Security Mechanisms** : IDS/IPS, Honeypots and evasion techniques, Secure Code Reviews (Fortify tool, OWASP Secure Coding Guidelines)	**15L**

Textbook(s):

1) Certified Ethical Hacker Study Guide v9, Sean-Philip Oriyano, Sybex; Study Guide Edition,2016
2) CEH official Certified Ethical Hacking Review Guide, Wiley India Edition, 2007

Additional Reference(s):

1) Certified Ethical Hacker: Michael Gregg, Pearson Education,1st Edition, 2013

2) Certified Ethical Hacker: Matt Walker, TMH,2011

3) http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines

4) https://www.owasp.org/index.php/Category:OWASP_Top_Ten_2017_Project

5) https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10

6) https://www.owasp.org/index.php/OWASP_Testing_Guide_v4_Table_of_Contents

7) https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

8) https://cve.mitre.org/

9) https://access.redhat.com/blogs/766093/posts/2914051

10) http://resources.infosecinstitute.com/applications-threat-modeling/#gref

11) http://www.vulnerabilityassessment.co.uk/Penetration%20Test.html

Suggested List of Practical – SEMESTER VI

Course: **USCSP601**	**(Credits : 02 Lectures/Week:06)** **Practical of Elective-I**	

USCS601: Wireless Sensor Networks and Mobile Communication

Practical experiments require software tools like INET Framework for OMNeT++, NetSim , TOSSIM, Cisco packet tracer 6.0 and higher version.

1. Understanding the Sensor Node Hardware. (For Eg. Sensors, Nodes(Sensor mote), Base Station, Graphical User Interface.)
2. Exploring and understanding TinyOS computational concepts:- Events, Commands and Task.
   - nesC model
   - nesC Components
3. Understanding TOSSIM for
   - Mote-mote radio communication
   - Mote-PC serial communication
4. Create and simulate a simple adhoc network
5. Understanding, Reading and Analyzing Routing Table of a network.
6. Create a basic MANET implementation simulation for Packet animation and Packet Trace.
7. Implement a Wireless sensor network simulation.
8. Create MAC protocol simulation implementation for wireless sensor Network.
9. Simulate Mobile Adhoc Network with Directional Antenna
10. Create a mobile network using Cell Tower, Central Office Server, Web browser and Web Server. Simulate connection between them.

USCS602: Cloud Computing

1. Study and implementation of Infrastructure as a Service.
2. Installation and Configuration of virtualization using KVM.
3. Study and implementation of Infrastructure as a Service
4. Study and implementation of Storage as a Service
5. Study and implementation of identity management
6. Study Cloud Security management

7. Write a program for web feed.
8. Study and implementation of Single-Sing-On.
9. User Management in Cloud.
10. Case study on Amazon EC2/Microsoft Azure/Google Cloud Platform

USCS603: Cyber Forensics

1. Creating a Forensic Image using FTK Imager/Encase Imager :

- Creating Forensic Image
- Check Integrity of Data
- Analyze Forensic Image

2. Data Acquisition:

- Perform data acquisition using:
- USB Write Blocker + Encase Imager
- SATA Write Blocker + Encase Imager
- Falcon Imaging Device

3. Forensics Case Study:

- Solve the Case study (image file) provide in lab using Encase Investigator or Autopsy

4. Capturing and analyzing network packets using Wireshark (Fundamentals) :

- Identification the live network
- Capture Packets
- Analyze the captured packets

5. Analyze the packets provided in lab and solve the questions using Wireshark :

- What web server software is used by www.snopes.com?
- About what cell phone problem is the client concerned?
- According to Zillow, what instrument will Ryan learn to play?
- How many web servers are running Apache?
- What hosts (IP addresses) think that jokes are more entertaining when they are explained?

6. Using Sysinternals tools for Network Tracking and Process Monitoring :

- Check Sysinternals tools

- Monitor Live Processes
- Capture RAM
- Capture TCP/UDP packets
- Monitor Hard Disk
- Monitor Virtual Memory
- Monitor Cache Memory

7. Recovering and Inspecting deleted files

- Check for Deleted Files
- Recover the Deleted Files
- Analyzing and Inspecting the recovered files

Perform this using recovery option in ENCASE and also Perform manually through command line

8. Acquisition of Cell phones and Mobile devices
9. Email Forensics

- Mail Service Providers
- Email protocols
- Recovering emails
- Analyzing email header

10. Web Browser Forensics

- Web Browser working
- Forensics activities on browser
- Cache / Cookies analysis
- Last Internet activity

Course: **USCSP602**	**(Credits : 02 Lectures/Week:06)** **Practical of Elective-II**	

USCS604: Information Retrieval

Practical may be done using software/tools like Python / Java / Hadoop

1. Write a program to demonstrate bitwise operation.
2. Implement Page Rank Algorithm.
3. Implement Dynamic programming algorithm for computing the edit distance between

strings s1 and s2. (Hint. Levenshtein Distance)

4. Write a program to Compute Similarity between two text documents.
5. Write a map-reduce program to count the number of occurrences of each alphabetic character in the given dataset. The count for each letter should be case-insensitive (i.e., include both upper-case and lower-case versions of the letter; Ignore non-alphabetic characters).
6. Implement a basic IR system using Lucene.
7. Write a program for Pre-processing of a Text Document: stop word removal.
8. Write a program for mining Twitter to identify tweets for a specific period and identify trends and named entities.
9. Write a program to implement simple web crawler.
10. Write a program to parse XML text, generate Web graph and compute topic specific page rank.

USCS605: Digital Image Processing

Practical need to be performed using Scilab under Linux or Windows

1. 2D Linear Convolution, Circular Convolution between two 2D matrices
2. Circular Convolution expressed as linear convolution plus alias
3. Linear Cross correlation of a 2D matrix, Circular correlation between two signals and Linear auto correlation of a 2D matrix, Linear Cross correlation of a 2D matrix
4. DFT of 4x4 gray scale image
5. Compute discrete cosine transform, Program to perform KL transform for the given 2D matrix
6. Brightness enhancement of an image, Contrast Manipulation, image negative
7. Perform threshold operation, perform gray level slicing without background
8. Image Segmentation
9. Image Compression
10. Binary Image Processing and Colour Image processing

USCS606:Data Science

Practical shall be performed using R

1. Practical of Data collection, Data curation and management for Unstructured data (NoSQL)

2. Practical of Data collection, Data curation and management for Large-scale Data system (such as MongoDB)

3. Practical of Principal Component Analysis

4. Practical of Clustering

5. Practical of Time-series forecasting

6. Practical of Simple/Multiple Linear Regression

7. Practical of Logistics Regression

8. Practical of Hypothesis testing

9. Practical of Analysis of Variance

10. Practical of Decision Tree

Course: **USCSP603**	**(Credits : 01 Lectures/Week: 03)** **Project Implementation**	

Please Refer to Project Implementation Guidelines

Course: **USCSP604**	**(Credits : 01 Lectures/Week: 03)** **Practical of Skill Enhancement**	

USCS607 : Ethical Hacking

1. Use Google and Whois for Reconnaissance
2. a) Use CrypTool to encrypt and decrypt passwords using RC4 algorithm

   b) Use Cain and Abel for cracking Windows account password using Dictionary attack and to decode wireless network passwords
3. a) Run and analyze the output of following commands in Linux – ifconfig, ping, netstat, traceroute

   b) Perform ARP Poisoning in Windows
4. Use NMap scanner to perform port scanning of various forms – ACK, SYN, FIN, NULL, XMAS
5. a) Use Wireshark (Sniffer) to capture network traffic and analyze

   b) Use Nemesy to launch DoS attack
6. Simulate persistent cross-site scripting attack
7. Session impersonation using Firefox and Tamper Data add-on

8. Perform SQL injection attack 9. Create a simple keylogger using python 10. Using Metasploit to exploit (Kali Linux)

Project Implementation Guidelines

1. A learner is expected to carry out two different projects: one in Semester V and another in Semester VI.
2. A learner can choose any topic which is covered in Semester I- semester VI or any other topic with the prior approval from head of the department/ project in charge.
3. The Project has to be performed individually.
4. A learner is expected to devote around three months of efforts in the project.
5. The project can be application oriented/web-based/database/research based.
6. It has to be an implemented work; just theoretical study will not be acceptable.
7. A learner can choose any programming language, computational techniques and tools which have been covered during BSc course or any other with the prior permission of head of the department/ project guide.
8. A project guide should be assigned to a learner. He/she will assign a schedule for the project and hand it over to a learner. The guide should oversee the project progress on a weekly basis by considering the workload of 3 lectures as assigned.
9. The quality of the project will be evaluated based on the novelty of the topic, scope of the work, relevance to the computer science, adoption of emerging techniques/technologies and its real-world application.
10. A learner has to maintain a project report with the following subsections
    a) Title Page
    b) Certificate

    A certificate should contain the following information –

    - The fact that the student has successfully completed the project as per the syllabus and that it forms a part of the requirements for completing the BSc degree in computer science of University of Mumbai.
    - The name of the student and the project guide
    - The academic year in which the project is done
    - Date of submission,
    - Signature of the project guide and the head of the department with date along with the department stamp,

- Space for signature of the university examiner and date on which the project is evaluated.

c) Self-attested copy of Plagiarism Report from any open source tool.

d) Index Page detailing description of the following with their subsections:

- Title: A suitable title giving the idea about what work is proposed.
- Introduction: An introduction to the topic giving proper back ground of the topic.
- Requirement Specification: Specify Software/hardware/data requirements.
- System Design details : Methodology/Architecture/UML/DFD/Algorithms/protocols etc. used(whichever is applicable)
- System Implementation: Code implementation
- Results: Test Cases/Tables/Figures/Graphs/Screen shots/Reports etc.
- Conclusion and Future Scope: Specify the Final conclusion and future scope
- References: Books, web links, research articles, etc.

11. The size of the project report shall be around twenty to twenty five pages, excluding the code.
12. The Project report should be submitted in a spiral bound form
13. The Project should be certified by the concerned Project guide and Head of the department.
14. A learner has to make a presentation of working project and will be evaluated as per the Project evaluation scheme

Scheme of Examination

1. **Theory:**

I. **Internal 25 Marks :**

a) **Test – 20 Marks**

20 marks Test – Duration 40 mins
It will be conducted either using any open source learning management system like Moodle (Modular object-oriented dynamic learning environment)

OR

A test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

b) **5 Marks –** Active participation in routine class instructional deliveries
Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

II. **External 75 Marks as per University Guidelines**

11. **Practical and Project Examination:**

There will be separate Practical examination for Elective-I, II, Skill enhansement and project of these Elective-I 100, Elective-II: 100 and Skill Enhansement: 50 and Project Implementation: 50.

In the Practical Examination of Elective-I and II, the student has to perform practical on each of the subjects chosen. The Marking Scheme for each of the Elective is given below:

	Subject Code	**Experiment-I**	**Experiment-II**	**Total Marks**
Elective-I	**USCSP501/ USCSP601**	Experiment-40+Journal-5 +viva-5 Total:50M	Experiment-40+Journal-5+viva-5 Total:50M	100 M
Elective-II	**USCSP502/ USCSP602**	Experiment-40+Journal-5 +viva-5 Total:50M	Experiment-40+Journal-5+viva-5 Total:50M	100 M

Project Implementation	**USCSP503/ USCSP603**	****Project Evaluation Scheme**	**50M**
Skill Enhancement	**USCSP504/ USCSP604**	Experiment-40+Journal:5+viva-5 Total-50M	**50M**
Total Marks			**300M**

(Certified Journal is compulsory for appearing at the time of Practical Examination)

****Project Evaluation Scheme:**

Presentation	**Working of the Project**	**Quality of the Project**	**Viva**	**Documentation**
10Marks	10 Marks	10 Marks	10 Marks	10Marks

(Certified Project Document is compulsory for appearing at the time of Project Presentation)

****