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		