AC – 29/06/2021
Item No: 6.38

UNIVERSITY OF MUMBAI

Syllabus

For the

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

Course: Computer Science

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

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.

F.Y.B.Sc. Computer Science Syllabus

Choice Based Credit System (CBCS)

with effect from

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

Semester I - Practical

Course: USCSP1	Practical of USCSP101 + USCSP102 + USCSP103+USCSP104+USCSP105+USCSP106 (Credits : 6, Lectures/Week: 18)	
USCSP101	**Digital Systems & Architecture** 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	
USCSP102	**Introduction to Programming with Python** 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.	
USCSP103	**LINUX Operating System** 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 2. Becoming an Ubuntu power user a) Administering system and User setting b) Learning Unity keyboard c) Using the Terminal 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, etc5 Editors: vi, sed, awk 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.	

USCSP104	**Open Source Technologies** 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 • dentify any topic of your choice and contribute the missinginformation 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. • 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	
USCSP105	**Discrete Mathematics** 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 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	
USCSP106	**Descriptive Statistics** **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 Simple Correlation & Regression using R	

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.
