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 – 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 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.		

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.
