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 - Practical

Course: USCSP2	Practical of USCSP201 + USCSP202 + USCSP203+USCSP204+USCSP205+USCSP206 (Credits : 6, Lectures/Week: 18)	
USCSP 201	**Design & Analysis of Algorithms** 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.	
USCSP202	Advanced Python Programming 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	
USCSP203	**Introduction to OOPs using C++** 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	
USCSP204	**Database Systems** 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.	
USCSP205	**Calculus** 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.	
USCSP206	**Statistical Methods** 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: $\mu = \mu 0$) d. Test of significance for difference between two means. (Double mean Ho: $\mu 1 = \mu 2$) 8 Small sample tests based on t and F- a. t-test for significance of single mean, population variance being unknown (Single mean Ho : $\mu = \mu 0$) 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.	

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.
