

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

<b>Semester – II</b>				
<b>Course Code</b>	<b>Course Type</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures/Week</b>
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
USCS203	Introduction to OOPs using C++	2	3
<p><b>About the Course:</b>            The course aims to introduce a new programming paradigm called Object Oriented Programming. This will be covered using C++ programming language. C++ is a versatile programming language, which supports a variety of programming styles, including procedural, object-oriented, and functional programming. This makes C++ powerful as well as flexible. It can be used to develop software such as operating systems, databases, and compilers.</p>			
<p><b>Course Objectives:</b></p>			
<p><b>Learning Outcomes:</b>            After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <li><input type="checkbox"/> Work with numeric, character and textual data and arrays.</li> <li><input type="checkbox"/> Understand the importance of OOP approach over procedural language.</li> <li><input type="checkbox"/> Understand how to model classes and relationships using UML.</li> <li><input type="checkbox"/> Apply the concepts of OOPS like encapsulation, inheritance and polymorphism.</li> <li><input type="checkbox"/> Handle basic file operations.</li> </ul>			
Unit	Topics	No of Lectures	
I	<p><b>Introduction to Programming Concepts:</b> Object oriented programming paradigm, basic concepts of object oriented programming, benefits of object oriented programming, object oriented languages, applications of object oriented programming.</p> <p>Tokens-keywords, identifiers, constants-integer, real, character and string constants, backslash constants, features of C++ and its basic structure, simple C++ program without class, compiling and running C++ program.</p> <p><b>Data Types, Data Input Output and Operators:</b> Basic data types, variables, rules for naming variables, programming constants, the type cast operator, implicit and explicit type casting, cout and cin statements, operators, precedence of operators.</p> <p><b>Decision Making, Loops, Arrays and Strings:</b> Conditional statements-if, if...else, switch loops- while, do...while, for, types of arrays and string and string manipulations</p> <p><b>Unified Modeling Language (UML):</b> Introduction to UML &amp; class diagrams.</p> <p><b>Classes, Abstraction &amp; Encapsulation:</b> Classes and objects, Dot Operator, data members, member functions, passing data to functions, scope and visibility of variables in function.</p>	15	

<p style="text-align: center;"><b>II</b></p>	<p><b>Constructors and Destructors:</b> Default constructor, parameterized constructor, copy constructor, private constructor, destructors.</p> <p><b>Working with objects:</b>Accessor - mutator methods, static data and static function, access specifiers, array of objects.</p> <p><b>Polymorphism</b> - Binding-static binding &amp; overloading, constructor overloading function overloading, operator overloading, overloading unary and binary operators.</p> <p><b>Modelling Relationships in Class Diagrams:</b> Association, Aggregation-Composition and examples covering these principles</p>	<p style="text-align: center;"><b>15</b></p>
<p style="text-align: center;"><b>III</b></p>	<p><b>Inheritance:</b> Defining base class and its derived class, access specifiers, types of inheritance-single, multiple, hierarchical, multilevel, hybrid inheritance, friend function and friend class, constructors in derived classes.</p> <p><b>Modelling Relationships:</b> Generalization-Specialization and examples covering these principles</p> <p><b>Run time Polymorphism</b> - Dynamic Binding, Function overriding, virtual function, pure virtual function, virtual base class, abstract class.</p> <p><b>Pointers:</b> Introduction to pointers, * and &amp; operators, assigning addresses to pointer variables, accessing values using pointers, pointers to objects &amp; this pointer, pointers to derived classes</p> <p><b>File Handling:</b> File Stream classes, opening and closing file-file opening modes, text file handling, binary file handling.</p> <p><b>Applying OOP to solve real life applications:</b> To cover case studies like library management, order management etc. to design classes covering all relationships</p>	<p style="text-align: center;"><b>15</b></p>
<p><b>Textbooks:</b></p> <ol style="list-style-type: none"> <li>1. Object Oriented Programming with C++, Balagurusamy E., 8th Edition, McGraw Hill Education India.</li> <li>2. UML &amp; C++: A Practical Guide to Object Oriented Development, Lee/Tepfenhart, Pearson Education, 2<sup>nd</sup> Edition 2015</li> </ol> <p><b>Additional References:</b></p> <ol style="list-style-type: none"> <li>1. Mastering C++ by Venugopal, Publisher: McGraw-Hill Education, 2017</li> <li>2. Let Us C++ by Kanetkar Yashwant, Publisher: BPB Publications, 2020</li> <li>3. Object Oriented Analysis and Design by Timothy Budd TMH, 2001</li> </ol>		

<b>Course Code</b>	<b>Course Title</b>	<b>Credits</b>	<b>Lectures /Week</b>
<b>USCSP203</b>	<b>Introduction to OOPs using C++ - Practical</b>	<b>1</b>	<b>3</b>
1	Program to demonstrate 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		

## 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:

<b>All questions are compulsory.</b>			
<b>Question</b>	<b>Based on</b>	<b>Options</b>	<b>Marks</b>
Q.1	Unit I	<i>Any 4 out of 6</i>	20
Q.2	Unit II	<i>Any 4 out of 6</i>	20
Q.3	Unit III	<i>Any 4 out of 6</i>	20
Q.4	Unit I,II and III	<i>Any 5 out of 6</i>	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 subject carries 50 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.

-----