

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
USCS202	Advanced Python Programming	2	3
<p>About the Course: This course aims to explore and enable learners to master the skills of advanced topics in Python Programming. It helps learners develops advanced skills such as working with databases, matching patterns, implementing threads and exception handling and GUI in Python. It also highlights and why Python is a useful scripting language for all developers.</p>			
<p>Course Objectives:</p> <ul style="list-style-type: none"> <input type="checkbox"/> To learn how to design objectoriented programs with Python classes. <input type="checkbox"/> To learn about reading, writing and implementing other operation on files in Python. <input type="checkbox"/> To implement threading concept and multithreading on Python <input type="checkbox"/> To design GUI Programs and implement database interaction using Python. <input type="checkbox"/> To know about use of regular expression and handling exceptions for writing robust python programs. 			
<p>Learning Outcomes: After successful completion of this course, students would be able to</p> <ul style="list-style-type: none"> <input type="checkbox"/> Ability to implement OOP concepts in Python including Inheritance and Polymorphism <input type="checkbox"/> Ability to work with files and perform operations on it using Python. <input type="checkbox"/> Ability to implement regular expression and concept of threads for developing efficient program <input type="checkbox"/> Ability to implement exception handling in Python applications for error handling. <input type="checkbox"/> Knowledge of working with databases, designing GUI in Python and implement networking in Python 			
Unit	Topics	No of Lectures	
I	<p>Working with files: Files, opening and closing a file, working with text files containing strings, knowing whether a file exists or not, working with binary files, the „with“ statement, the seek() and tell() methods, random accessing of binary files, zipping and unzipping files, working with directories, running other programs from python program</p> <p>Regular expressions: What is a regular expression?, sequence characters in regular expressions, quantifiers in regular expressions, special characters in regular expressions, using regular expression on files, retrieving information from an html file,</p> <p>Threads in python: Difference between process and thread, types of threads, benefits of threads, creating threads, single tasking and multitasking, thread synchronization, deadlock in threads, daemon threads</p>	15	

	<p>Date and time in python: Date and time now, combining date and time, formatting dates and times, finding durations using “time delta”, comparing two dates, sorting dates, stopping execution temporarily, knowing the time taken by a program, calendar module</p>	
II	<p>Database in python: Using SQL with python, retrieving rows from a table, inserting rows into a table, deleting rows from a table, updating rows in a table, creating database tables through python, Exception handling in databases.</p> <p>Exceptions in python: Errors in a python program, compile & run-time errors, logical error, exceptions-exception handling, types of exceptions, the except block, the assert statement, user-defined exceptions, logging the exceptions</p> <p>Networking: Protocols,server-client architecture, tcp/ip and udp communication</p> <p>Graphical user interface: Creating a GUI in python, Widget classes, Working with Fonts and Colours, working with Frames, Layout manager, Event handling</p>	15
III	<p>OOPS in python: Features of Object Oriented Programming system (oops)- classes and objects, encapsulation, abstraction, inheritance, polymorphism, constructors and destructors</p> <p>Classes and objects: Creating a class, the self-variable, types of variables, namespaces, types of methods, instance methods, class methods, static methods, passing members of one class to another class, inner classes</p> <p>Inheritance and polymorphism: Inheritance in python, types of inheritance- single inheritance, multilevel inheritance, hierarchical inheritance, multiple inheritance, constructors in inheritance, overriding super class constructors and methods, the super() method, method resolution order (mro), polymorphism, duck typing, operator overloading, method overloading, method overriding,</p> <p>Abstract classes and interfaces: Abstract class, abstract method,interfaces in python, abstract classes vs. Interfaces</p>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Paul Gries , Jennifer Campbell, Jason Montojo, Practical Programming: An Introduction to Computer Science Using Python 3, Pragmatic Bookshelf, 3rd Edition, 2018 2. Programming through Python, M. T Savaliya, R. K. Maurya, G M Magar, Revised Edition, Sybgen Learning India, 2020 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Advanced Python Programming, Dr. Gabriele Lanaro, Quan Nguyen, SakisKasampalis, Packt Publishing, 2019 2. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018 3. Python: The Complete Reference, Martin C. Brown, McGraw Hill, 2018 4. Beginning Python: From Novice to Professional, Magnus Lie Hetland, Apress, 2017 5. Programming in Python 3, Mark Summerfield, Pearson Education, 2nd Ed, 2018 		

Course Code	Course Title	Credits	Lectures /Week
USCSP202	Advanced Python Programming – Practical	1	3
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 <ul style="list-style-type: none"> 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 handling.		
6	Write a GUI Program in Python to design application that demonstrates <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> a. Types of Methods b. Inheritance c. Polymorphism 		
10	Write a program to Python program to implement concepts of OOP such as <ul style="list-style-type: none"> a. Abstract methods and classes b. Interfaces 		

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	<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.
