

UNIVERSITY OF MUMBAI

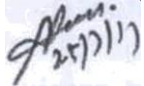
No. UG/ J>Tif 2017

CIRCULAR:-

A reference is **invited** to the Syllabi relating to the B.Sc. degree course, vide this office Circular No. UG/42 of 2016-17, dated 5th August , 2016 and the Principals of the affiliated Colleges in Science are hereby informed that the recommendation made by Ad-hoc-Board of Studies Ln Computer Science at its meeting held on **5/5/2017** has been accepted by the Academic Council at its meeting held on 11.5.2017 vide item No. 4.210 and that in accordance therewith, in revised syllabus as per the Credit Based Semester and Grading System for S.Y.B.Sc Computer Science (Sem III & IV) which is available on the University's website (www.mu.ac.in) and that the same has been brought into force with effect from the academic year 2016-17.

MUMBAI — 400 032

July, 2017


REGISTRAR

To,

The Prin.cipal of the affiliated Colleges in Science and the Head of Recognized Institutions concerned.

A.C/4.210/11.05.2017

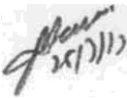
>>Io. UG/)bJ- A of 2017

.M AJ-40*0 032

२३th July, 2017

Copy forwarded with compliments for inTorr•.a.tion to

- 1) The Co-ordinator, Faculty of Science.,
- 2) The Offg. Director of Board of Examinations and Evaluation,
- 3) The Chairperson, Board of Studies in Botar.y,
- 4) The Director of Board of Studies Development.
- 5) The Professor-cum-Director, Ir.stitute of Distance and Open Leamlng.
- 6) The Co-Ordinator, University Cen.puierization Centre.



.... PTO

UNIVERSITY OF MUMBAI



**Syllabus for SemIII
Program: B.Sc.
Course: Computer Science**

(Credit Based Semester and Grading System with
effect from the academic year 2017-2018)

Preamble

The revised and restructured curriculum for the Three-year integrated course is systematically designed considering the current industry needs in terms of skills sets demanded under new technological environment. It also endeavours to align the programme structure and course curriculum with student aspirations and corporate expectations. The proposed curriculum is more contextual, industry affable and suitable to cater the needs of society and nation in present day context.

Second year of this course is about studying core computer science subjects. Theory of Computation course provides understanding of grammar, syntax and other elements of modern language designs. It also covers developing capabilities to design formulations of computing models and its applications in diverse areas.

The course in Operating System satisfies the need of understanding the structure and functioning of system. Programming holds key indispensable position in any curriculum of Computer Science. It is essential for the learners to know how to use object oriented paradigms. There is also one dedicated course Android Developer Fundamentals as a skill enhancement catering to modern day needs of Mobile platforms and applications. The syllabus has Database Systems courses in previous semesters. The course in Database Management Systems is its continuation in third semester. The course has objectives to develop understanding of concepts and techniques for data management along with covers concepts of database at advance level.

The course of Combinatorics and Graph Theory in third semester and the course of Linear Algebra in fourth semester take the previous courses in Mathematics. Graph theory is rapidly moving into the mainstream mainly because of its applications in diverse fields which include can further open new opportunities in the areas of genomics, communications networks and coding theory, algorithms and computations and operations research.

Introducing one of the upcoming concepts Physical Computing and IoT programming will definitely open future area as Embedded Engineer, involvement in IoT projects, Robotics and many more. The RasPi is a popular platform as it offers a complete Linux server in a tiny platform for a very low cost and custom-built hardware with minimum complex hardware builds which is easier for projects in education domain.

S.Y.B.Sc. (Semester III and IV)
Computer Science Syllabus
Credit Based Semester and Grading System
To be implemented from the Academic year 2017-2018

SEMESTER III			
Course	TOPICS	Credits	L / Week
USCS301	Theory of Computation	2	3
USCS302	Core JAVA	2	3
USCS303	Operating System	2	3
USCS304	Database Management Systems	2	3
USCS305	Combinatorics and Graph Theory	2	3
USCS306	Physical Computing and IoT Programming	2	3
USCS307	Skill Enhancement: Web Programming	2	3
USCSP301	USCS302+USCS303+USCS304	3	9
USCSP302	USCS305+USCS306+USCS307	3	9

SEMESTER IV			
Course	TOPICS	Credits	L / Week
USCS401	Fundamentals of Algorithms	2	3
USCS402	Advanced JAVA	2	3
USCS403	Computer Networks	2	3
USCS404	Software Engineering	2	3
USCS405	Linear Algebra using Python	2	3
USCS406	.NET Technologies	2	3
USCS407	Skill Enhancement: Android Developer Fundamentals	2	3
USCSP401	USCS401+ USCS402+ USCS403	3	9
USCSP402	USCS405+ USCS406+ USCS407	3	9

Suggested List of Practical- SEMESTER III

Course: USCSP301	(Credits : 03 Lectures/Week: 09) USCS302+ USCS303+USCS304	
USCS302: Core JAVA		
<ol style="list-style-type: none"> 1. Accept integer values for a, b and c which are coefficients of quadratic equation. Find the solution of quadratic equation. 2. Accept two n x m matrices. Write a Java program to find addition of these matrices. 3. Accept n strings. Sort names in ascending order. 4. Create a package: Animals. In package animals create interface Animal with suitable behaviors. Implement the interface Animal in the same package animals. 5. Demonstrate Java inheritance using extends keyword. 6. Demonstrate method overloading and method overriding in Java. 7. Demonstrate creating your own exception in Java. 8. Using various swing components design Java application to accept a student's resume. (Design form) 9. Write a Java List example and demonstrate methods of Java List interface. 10. Design simple calculator GUI application using AWT components. 		
USCS303: Operating System		
<i>Practical can be implemented either in JAVA or any other programming language.</i>		
<ol style="list-style-type: none"> 1. Process Communication: <ol style="list-style-type: none"> (i) Give solution to the producer–consumer problem using shared memory. (ii) Give solution to the producer–consumer problem using message passing. (iii) One form of communication in a Client–Server Systems environment is Remote method invocation (RMI). RMI is a Java feature similar to RPCs. RMI allows a thread to invoke a method on a remote object. Objects are considered remote if they reside in a different Java virtual machine (JVM). Demonstrate RMI program for adding/subtracting/multiplying/dividing two numbers. 2. Threads: <ol style="list-style-type: none"> (i) The Java version of a multithreaded program that determines the summation of a 		

non-negative integer. The Summation class implements the Runnable interface. Thread creation is performed by creating an object instance of the Thread class and passing the constructor a Runnable object.

- (ii) Write a multithreaded Java program that outputs prime numbers. This program should work as follows: The user will run the program and will enter a number on the command line. The program will then create a separate thread that outputs all the prime numbers less than or equal to the number entered by the user.
- (iii) The Fibonacci sequence is the series of numbers 0, 1, 1, 2, 3, 5, 8, ... Formally, it can be expressed as: $fib_0 = 0$, $fib_1 = 1$, $fib_n = fib_{n-1} + fib_{n-2}$ Write a multithreaded program that generates the Fibonacci sequence using either the Java,

3. Synchronization:

- (i) Give Java solution to Bounded buffer problem.
- (ii) Give solution to the readers–writers problem using Java synchronization.
- (iii) The Sleeping-Barber Problem: A barber shop consists of awaiting room with n chairs and a barber room with one barber chair. If there are no customers to be served, the barber goes to sleep. If a customer enters the barbershop and all chairs are occupied, then the customer leaves the shop. If the barber is busy but chairs are available, then the customer sits in one of the free chairs. If the barber is asleep, the customer wakes up the barber. Write a program to coordinate the barber and the customers using Java synchronization.

4. Implement FCFS scheduling algorithm in Java.

5. Implement SJF (with no preemption) scheduling algorithm in Java

6. Implement RR scheduling algorithm in Java

7. Write a Java program that implements the banker's algorithm

8. Write a Java program that implements the FIFO page-replacement algorithm.

9. Write a Java program that implements the LRU page-replacement algorithm.

10. Design a File System in Java.

USCS304: Database Management Systems

1. Creating and working with Insert/Update/Delete Trigger using Before/After clause.
2. Writing PL/SQL Blocks with basic programming constructs by including following:
 - a. Sequential Statements
 - b. unconstrained loop
3. Sequences:
 - a. Creating simple Sequences with clauses like START WITH, INCREMENT BY, MAXVALUE, MINVALUE, CYCLE | NOCYCLE, CACHE | NOCACHE, ORDER | NOORDER.
 - b. Creating and using Sequences for tables.
4. Writing PL/SQL Blocks with basic programming constructs by including following:
 - a. If...then...Else, IF...ELSIF...ELSE... END IF
 - b. Case statement
5. Writing PL/SQL Blocks with basic programming constructs for following Iterative Structure:
 - a. While-loop Statements
 - b. For-loop Statements.
6. Writing PL/SQL Blocks with basic programming constructs by including a GoTO to jump out of a loop and NULL as a statement inside IF
7. Writing Procedures in PL/SQL Block
 - a. Create an empty procedure, replace a procedure and call procedure
 - b. Create a stored procedure and call it
 - c. Define procedure to insert data
 - d. A forward declaration of procedure
8. Writing Functions in PL/SQL Block.
 - a. Define and call a function
 - b. Define and use function in select clause,
 - c. Call function in dbms_output.put_line
 - d. Recursive function
 - e. Count Employee from a function and return value back
 - f. Call function and store the return value to a variable
9. Writing a recursive Functions in PL/SQL Block
10. Study of transactions and locks

Evaluation Scheme

I. Internal Exam - 25 Marks

(i) Test – 20 Marks

20 marks Test – Duration 40 mins

It will be conducted either using any open source learning management system like Moodle (Modular object-oriented dynamic learning environment)

OR

A test based on an equivalent online course on the contents of the concerned course (subject) offered by or build using MOOC (Massive Open Online Course) platform.

(ii) 5 Marks – Active participation in routine class instructional deliveries

Overall conduct as a responsible student, manners, skill in articulation, leadership qualities demonstrated through organizing co-curricular activities, etc.

II. External Exam– 75 Marks

III. Practical Exam – 50 Marks

- Each course carry 50 Marks : 40 marks + 05 marks (journal) + 05 marks (viva)
- Minimum 75 % practical from each paper are required to be completed and written in the journal.

(Certified Journal is compulsory for appearing at the time of Practical Exam)
