UNIVERSITY OF MUMBAI

Syllabus for the F.Y.B.Sc.
Program: B.Sc.
Course: Computer Science

(Credit Based Semester and Grading System with effect from the academic year 2015–2016)

Preamble

The aim of the BSc Computer Science Syllabus has always been to lay the theoretical foundations of software and hardware equally supplemented by the practical techniques. With this strong foundation of computer science along with core subjects like Mathematics, Physics, Statistics etc, the computer science students are expected to contribute efficient solutions for the various problems that are given to them. Over this period of time, computer science students have proved this fact and have done well in Industries (mainly software) which have offered plenty of opportunities to them.

With the advancement in software industry and technological innovations, the industry demands from graduate and postgraduate students are changing. The syllabus is been designed to meet the industry expectations, to inspire the students to take-up higher education as well as research, to attract student over other courses and finally to fulfill the expectations of Credit system

The syllabus for these three years will be designed keeping these challenges in mind. The syllabus aims to cover core concepts of Computer Science and also to cover the latest technologies which can be accommodated at BSc level. One such step is that we would like to promote Open Source Technologies as much as possible.

All universities promote the use of ICT in education. With smart phones the power of computers has almost reached our palmtop. With so much of knowledge available on Internet, it is also the responsibility of teachers to make use of them in teaching, go beyond the books and make necessary changes in the approach and the contents. Students during these three years are expected to accumulate maximum knowledge and skills through this syllabus.

F.Y.B.Sc. Computer Science Syllabus
Credit Based System and Grading System
Academic year 2015-2016

SEMESTER I

Course	TOPICS	Credits	L / Week
USCS101	Fundamentals of Computer Organization & Embedded Systems	2	3
USCS102	Introduction to programming using Python	2	3
USCSP01	Practical of USCS101 + USCS102	2	6

SEMESTER II

Course	TOPICS	Credits	L / Week
USCS201	Introduction to Database Management Systems	2	3
USCS202	Advanced Programming using Python	2	3
USCSP02	Practical of USCS201 + USCS202	2	6

Semester I – Theory

Course: USCS101	TOPICS (Credits : 2 Lectures/Week: 3) Fundamentals of Computer Organization & Embedded Systems	
\multicolumn Objectives – The objective of this course is to make the learner ready to understand computer architecture & organization and microcontroller. To cover-up with the pre-requisites of embedded computing.		

Objectives – The objective of this course is to make the learner ready to understand computer architecture & organization and microcontroller. To cover-up with the pre-requisites of embedded computing.

Expected Learning Outcomes: Through this course there will be an enhancement to

1) Apply knowledge of computer architecture and organization appropriate to the discipline
2) Analyze given processing element, and identify and define the computing requirements.
3) Design, implement, and evaluate a microcontroller-based system, process, component, or program to meet desired needs.
4) Use current techniques, skills, and tools necessary for Low-Level computing.

Unit	Topics	Lectures
Unit I	**Introduction to computer organization:** **Introduction to computer internals and organization,** study of internal Architecture, understanding the system's Structure and Function. **System's technological Evolution with respect to Performance,** A Brief History of Computers systems, optimized Performance for processing, Time-line of Intel x86 Architecture, world of embedded Systems and the ARM architecture, Performance assessment (Clock Speed and Instructions per Second, Benchmarks) **Macroscopic view of Computer Function and Interconnection** Components of computers**,** Its Function, Interconnection: Structures & Buses. **Internal Memory:** Semiconductor memory, Organization, DRAM, SRAM, Types of ROM. **Cache Memory:** Characteristics of Memory Systems, Memory Hierarchy (Size, Speed, Cost), Introduction to Cache Memory & Its Principles.	15 L
Unit II	**Preliminaries for Embedded Systems:** **Processing Elements:** microprocessors and microcontrollers, Introduction to processing elements, basics of microprocessor & microcontrollers, History and evolution of Microcontrollers: 4-bit to 32-bit, Development systems for microcontrollers **Introduction to Number Systems** Binary, Hexadecimal, their conversions and arithmetic (ADDITION & Multiplication), Binary codes (ASCII, BCD, GRAY) **Operating System** Introduction to OS, Concept of Scheduling, Concept of Main Memory Management, RTOS structure and working. RTOS v/s GPOS **Reduced Instruction Set Computers (RISCs),** Instruction Execution Characteristics, Introduction to CISC.	15 L
Unit III	**Intro to microcontroller:** **Introduction to 8051 Architecture,** micro controller's internal hardware, Parallel I/O ports, pin-outs & circuits, Extended main memory, Internal Timers & counters, serial data I/O interconnection, interrupts management. **Embedded Hardware Design & Development** Components (Analog & Digital), EDA tools, PCB layout design and its Guidelines.	15 L

| | Input/Output: External Peripheral Devices, Concept of Device Drivers, Concept & components of I/O Modules, Types of Interfacing I/O peripherals (Programmed I/O, Interrupt-Driven I/O, Direct Memory Access), I/O Channels and Processors, Communication Interface: I^2C, SPI, UART, USB, WI-FI and FireWire. | |

Text book(s):

1) William Stallings, *Computer organization and architecture, designing for performance,* Pearson Education, Ninth Edition, 2012

2) Kenneth J. Ayala, *The 8051 microcontroller,* Cengage Learning, Third edition, 2013

3) Shibu K V, I*ntroduction to Embedded Systems,* McGraw Hill, 2011

Additional References:

1) Subrata Ghoshal, *8051 Microcontroller-Internals, Instructions, Programming & Interfacing*, Pearson Education, 2010

2) Muhammad Ali Mazidi, Rolin D. Mckinlay, Janice Gillispie Mazidi, *The 8051 Microcontroller: A Systems Approach,* Pearson Education, 2013

3) Ajay Deshmukh, *Microcontrollers: Theory and Applications,* Tata McGraw-Hill, 2005

Course: USCS102	TOPICS (Credits : 2 Lectures/Week: 3) Introduction to programming using Python	
colspan="3"	**Objectives – The objective of this course is to introduce various concepts of programming to the students using Python.** **Expected Learning Outcomes of this course** 1) Students should be able to understand the concepts of programming before actually starting to write new programs. . 2) Students should be able to understand what happens in the background when the programs are executed 3) Students should be able to develop logic for Problem Solving. 4) Students should be made familiar about the basic constructs of programming such as data, operations, conditions, loops, functions etc. 5) Students should be able to apply the problem solving skills using syntactically simple language i. e. **Python (version: 3.X or higher)**	
Unit I	**Introduction to Programming Languages :** What is program and programming paradigms, Programming languages-their classification and characteristics, language translators and language translation activities, Use of Algorithms/Flow Charts for problem solving **Building Blocks of Program:** Data, Data Types, Data Binding, Variables, Constants, Declaration, Operations on Data such as assignment, arithmetic, relational, logical operations, dry run, Evaluating efficiency of algorithms in terms of number of operations and variables used **Introduction to Python Programming**: Features, basic syntax, Writing and executing simple program, Basic Data Types such as numbers, strings, etc Declaring variables, Performing assignments, arithmetic operations, Simple input-output	15 L
Unit II	**Sequence Control –** Precedence of operators, Type conversion **Conditional Statements:** if, if-else, nested if –else **Looping**: for, while, nested loops	15 L

	Control statements: Terminating loops, skipping specific conditions **String Manipulation:** declaring strings, string functions **Manipulating Collections** Lists, Tuples	
Unit III	**Dictionaries –** Concept of dictionary, techniques to create, update & delete dictionary items. **Functions:** Defining a function, calling a function, Advantages of functions, types of functions, function parameters, Formal parameters, Actual parameters, anonymous functions, global and local variables **Modules:** Importing module, Creating & exploring modules, Math module, Random module, Time module,	15 L

Text books:

1) Charles Dierbach, *Introduction to Computer Science using Python*, Wiley, 2013

2) James Payne , *Beginning Python: Using Python 2.6 and Python 3,* Wiley India, 2010

Additional References:

1. Paul Gries , Jennifer Campbell, Jason Montojo, *Practical Programming: An Introduction to Computer Science Using Python 3*, Pragmatic Bookshelf, 2/E 2014

2. Adesh Pandey, *Programming Languages – Principles and Paradigms,* Narosa, 2008

Semester I – Practicals

USCSP01	**Practicals of USCS101 + USCS102 (Credits: 2, Pract/Week: 6)**
	Fundamentals of Computer Organization & Embedded Systems **(1)** Study and implementation of Basic Logic gates and Universal Logic gates using MMlogic / kit. **(2)** Study and implementation of Flip-flops (DFF,JKFF,SRFF,TFF) using kits or MMlogic. **(3)** Study and implementation of MUX and DEMUX (1:8,8:1) using kit or MMlogic. **(4)** Introduction to C programming (basic programs using if and switch) + introduce bitwise operations **(5)** Demonstration of loops in C program. (for, while and do-while) **(6)** Introduction to pointer and functions. (basic 1 program to demonstrate what pointer is , how to use , its need & to demonstrate function with call by value) **(7)** Demonstration of Interrupts: to print a name without using printf statement. **(8)** To read the contents of status register. **Python Programming** Using the Operating system (logging, creating – deleting folders, creating-deleting files, using editors etc.) **(1)** Installing python and setting up environment. Simple statements like printing the names, numbers, mathematical calculations, etc. **(2)** Simple programs containing variable declaration and arithmetic operations **(3)** Programs based on conditional constructs **(4)** Programs based on loops **(5)** Programs related to string manipulation **(6)** Programs related to Lists, Tuples **(7)** Programs related to dictionaries **(8)** Programs related to functions & modules

Semester II – Theory

Course: USCS201	TOPICS (Credits : 2 Lectures/Week: 3) Introduction to Database Management Systems	
Objectives – The objective of this course is to introduce the concept of the DBMS with respect to the relational model, to specify the functional and data requirements for a typical database application and to understand creation, manipulation and querying of data in databases. **Expected Learning Outcomes of the course:** 1) Students should be able to evaluate business information problem and find the requirements of a problem in terms of data. 2) Students should be able to design the database schema with the use of appropriate data types for storage of data in database. 3) Students should be able to create, manipulate, query and back up the databases.		
Unit I	**Introduction of DBMS –** Database, DBMS – Definition, Overview of DBMS, File processing system vs DBMS, Limitation of file processing system, Advantages of DBMS, Levels of abstraction, Data independence, DBMS Architecture, Users of DBMS, **Data models** - Object Based Logical Model, Record Based Logical Model (relational, hierarchical, network), **Entity Relationship Model -** Entities, attributes, entity sets, relations, relationship sets , Additional constraints (key constraints, participation constraints, weak entities, aggregation / generalization, Conceptual Design using ER (entities VS attributes, Entity Vs relationship, binary Vs ternary, constraints beyond ER), Entity Relationship Diagram (ERD) **Relational data model**– Domains, attributes, Tuples and Relations, Relational Model Notation, Characteristics of Relations, Relational Constraints - primary key, referential integrity, unique constraint, Null constraint, Check constraint	15 L
Unit II	**ER to The Relational Model** - Entity to Table, Relationship to tables with and without key constraints. **Introduction to Functional Dependencies and Normalization –** 1NF, 2NF, 3NF **Relational Algebra** operations (selection, projection, set operations union, intersection, difference, cross product, Joins –conditional, equi join and natural joins, division) **MySQL DDL Statements** - Creating Databases, Using Databases, MySQL datatypes, Creating Tables (with integrity constraints – primary key, default, check, not null), Altering Tables, Renaming Tables, Dropping Tables, Truncating Tables, Backing Up and Restoring databases **DML Statements** – Viewing the structure of a table insert, update, delete, **Select** – all columns, specific columns, unique records, conditional select, in clause, between clause, limit, aggregate functions (count, min, max, avg, sum), group by clause, having clause **DCL Statements** – savepoint, commit and rollback	15 L

Unit III	**MySQL Functions** – String Functions (concat, instr, left, right, mid, length, lcase/lower, ucase/upper, replace, strcmp, trim, ltrim, rtrim), Math Functions (abs, ceil, floor, mod, pow, sqrt, round, truncate) Date Functions (adddate, datediff, day, month, year, hour, min, sec, now, reverse) **MySQL Joining Tables** – inner join, outer join (left outer, right outer, full outer) **Subqueries** – sub queries with IN, EXISTS, sub queriesrestrictions, Nested sub queries, ANY/ALL clause, correlated sub queries **Database Protection:**Security Issues, Threats to Databases, Security Mechanisms, Role of DBA, Discretionary Access Control **MySQL** – **Views** (creating, altering dropping, renaming and manipulating views)**DCL Statements** (creating/dropping users, privileges introduction, granting/revoking privileges, viewing privileges)	15 L

Text books:

1) Ramez Elmasri & ShamkantB.Navathe, *Fundamentals of Database Systems*, Pearson Education, Sixth Edition, 2010.
2) Joel Murach, *Murach's MySQL*, Murach, 2012

Additional References:

1. Robert Sheldon, Geoff Moes, *Begning MySQL*, Wrox Press, 2005.
2. Ramakrishnam, Gehrke, *Database Management Systems*, McGraw-Hill, 2007

Course: USCS202	**TOPICS (Credits : 2 Lectures/Week: 3)** **Advanced Programming using Python**	
colspan	**Objective – The objective of this subject is to explore Python programming language and get a grip on structured programming. It should also give the idea to students through GUI programming and database connectivity about software development activities.** **Expected Learning Outcomes** 1) Students should be able to understand how to read/write to files using python. 2) Students should be able to catch their own errors that happen during execution of programs. 3) Students should get an introduction to the concept of pattern matching. 4) Students should be made familiar with the concepts of GUI controls and designing GUI applications. 5) Students should be able to connect to the database to move the data to/from the application. 6) Students should know how to search and sort data and how to determine efficiency of algorithms used.	
Unit I	**Python File Input-Output**: Opening and closing file, Various types of file modes, reading and writing to files, manipulating directories **Exception Handling** – What is exception, Various keywords to handle exception such try, catch, except, else, finally, raise – **Regular Expressions –** Concept of regular expression, various types of regular expressions, using match function	15 L
Unit II	**GUI Programming in Python (using Tkinter/wxPython/Qt) -** What is GUI, Advantages of GUI, Introduction to GUI library,	15 L

	Layout management, Events and bindings, Font, Colors, drawing on canvas (line, oval, rectangle, etc.) Widget such as : Frame, Label, Button, Checkbutton, Entry, Listbox, Message, Radiobutton, Text, Spinbox etc	
Unit III	**Database connectivity in Python –** Installing mysql connector, accessing connector module module, using connect, cursor, execute & close functions, reading single & multiple results of query execution, executing different types of statements, executing transactions, understanding exceptions in database connectivity **Algorithm, Searching and Sorting –** Searching and sorting techniques, Efficiency of algorithms	15 L

Text books:

1) Paul Gries , Jennifer Campbell, Jason Montojo, *Practical Programming: An Introduction to Computer Science Using Python 3*, Pragmatic Bookshelf, 2/E 2014

2) James Payne , *Beginning Python: Using Python 2.6 and Python 3,* Wiley India, 2010

Additional References:

1) A. Lukaszewski, *MySQL for Python: Database Access Made Easy,* Pact Publisher, 2010

Semester II - Practicals

USCSP02	**Practicals of USCS201 + USCS202 (Credits: 2, Pract/Week: 6)**

	Introduction to Database Management Systems
	(1) Practical No. 1
	• 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
	• Saving (Commit) and Undoing (rollback)
	(2) Practical No. 2
	• Altering a Table
	• Dropping/Truncating/Renaming Tables
	• Backing up / Restoring a Database
	(3) Practical No. 3
	• Simple Queries
	• SIMPLE Queries with Aggregate functions
	• Queries with Aggregate functions (group by and having clause)
	(4) Practical No. 4
	• Queries involving
	• Date Functions
	• String Functions
	• Math Functions
	   (On previously created tables and/or the dual table)
	(5) Practical No. 5
	• Join Queries
	    o Using 2 related tables
	    o More than 2 related tables

	(6) Practical No. 6
	   • Sub Queries
	(7) Practical No. 7
	   • Views
	   • Creating Views (with and without check option)
	   • Dropping views
	   • Selecting from a view
	(8) Practical No. 8
	   • Creating a savepoint
	   • Commit & Roll back
	   • Granting and revoking permissions

	Python Programming (1) Programs to read & write file. (2) Program to demonstrate exception handling (3) Program to demonstrate the use of regular expressions (4) Program to draw shapes (5) Program to show GUI controls and processing-I (6) Program to show GUI controls and processing-II (7) Program to show database connectivity (8) Programs to do searching and sorting

Evaluation Scheme

I. Internal Exam - Marks – 25

(i) Test – 20 Marks

20 marks Test – Duration 40 mins
It will be conducted either using any open source learning management system such as 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 - Marks – 75
1. Duration - 2.5 Hours.
2. Theory question paper pattern:-

All questions are compulsory.			
Question	Based on		Marks
Q.1	Unit I		[20]
Q.2	Unit II		[20]
Q.3	Unit III		[20]
Q.4	Unit I, II and III		[15]

All questions shall be compulsory with internal choice within the questions.
Each Question may be sub divided into sub questions as a, b, c, d & e, etc & the allocation of Marks depends on the weightage of the topic.

III. Practical Exam – 100 marks
(Certified Journal is compulsory for appearing for practical exam)

Practicals of USCS101 – 40 marks + 5 marks (journal) + 5 marks (viva)
Practicals of USCS102 - 40 marks + 5 marks (journal) + 5 marks (viva)