PART B

B.SC. COMPUTER SCIENCE COMPLEMENTARY ELECTIVE COURSES

[FOR B.SC.MATHEMATICS/B.SC.STATISTICS/B.SC.PHYSICS/B.SC. ELECTRONICS PROGRAMMES]

WORK AND CREDIT DISTRIBUTION

(2019 ADMISSION ONWARDS)

COURSE CODE	COURSE TITLE	SEMESTER	HOURS PER WEEK	CREDIT	EXAM HOURS	MARKS (INTERNAL + EXTERNAL)
1C01CSC	INTRODUCTION TO COMPUTERS AND PROGRAMMING	1	2	2	3	8+32
1C01CSC	LAB 1: PROGRAMMING IN C, WEB PROGRAMMING AND PYTHON PROGRAMMING	1	2	0	-	-
2C02CSC	PROGRAMMING IN C	2	2	2	3	8+32
2C02CSC	LAB 1: PROGRAMMING IN C, WEB PROGRAMMING AND PYTHON PROGRAMMING	2	2	0	-	-
3C03CSC	WEB TECHNOLOGY WITH DATA BASE MANAGEMENT SYSTEM	3	3	2	3	8+32
3C03CSC	LAB 1: PROGRAMMING IN C, WEB PROGRAMMING AND PYTHON PROGRAMMING	3	2	0	-	-
4C04CSC	COMPUTATION USING PYTHON	4	3	2	3	8+32
4C05CSC	LAB 1: PROGRAMMING IN C, WEB PROGRAMMING AND PYTHON PROGRAMMING*	4	2	4	3	8+32

TOTAL 200 MARKS

- PRACTICAL DONE IN ALL THE 4 SEMESTER

EVALUATION

ASSESSMENT	WEIGHTAGE
EXTERNAL	4
INTERNAL	1

CONTINUOUS EVALUATION FOR THEORY

COMPONENT	WEIGHTAGE	REMARKS
COMPONENT1: TEST	75%	MINIMUM OF 2 TESTS SHOULD BE CONDUCTED. MARKS FOR THE TEST COMPONENT SHOULD BE CALCULATED AS THE AVERAGE OF THE BEST TWO MARKS OBTAINED IN THE TESTS CONDUCTED.
COMPONENT 2: ASSIGNMENT/ SEMINAR/VIVA	25%	ANY ONE COMPONENT

PATTERN OF QUESTION PAPER FOR END SEMESTER EVALUATION

Part A	**Short Answer**	**5 Questions x 1 Mark = 5 Marks**
	Answer all questions	5 Questions x 1 Mark = 5 Marks
Part B	**Short Essay**	**6 Questions x 2 Marks = 12 Marks**
	Answer any 4 questions	4 Questions x 2 Marks = 8 Marks
Part C	**Essay**	**5 Questions x 3 Marks = 15 Marks**
	Answer any 3 questions	3 Questions x 3 Marks = 9 Marks
Part D	**Long Essay**	**4 Questions x 5 Marks = 20 Marks**
	Answer any 2 questions	2 Questions x 5 Marks = 10 Marks
Total Marks Including Choice: 52		
Maximum Marks for the Course: 32		

CONTINUOUS EVALUATION FOR PRACTICAL

COMPONENT	WEIGHTAGE	REMARKS
COMPONENT 1: LAB SKILLS, OBSERVATION NOTE AND PUNCTUALITY	25% FOR LAB SKILL 25% FOR OBSERVATION NOTE AND PUNCTUALITY	OBSERVATION NOTE IS MANDATORY. MARKS SHOULD BE GIVEN CONSIDERING OBSERVATION NOTE LAB SKILLS AND PUNCTUALITY.
COMPONENT1: TEST	50%	MODEL EXAMINATION SHOULD BE CONDUCTED BEFORE EXTERNAL EXAM AND CONSIDERED FOR INTERNAL MARK

END SEMESTER EVALUATION FOR PRACTICAL

COMPONENT	PART A	PART B
Code Writing	7	7
Execution & Output	8	8
Record	2	
Total Marks	**32**	

COMPLEMENTARY ELECTIVE COURSE I: INTRODUCTION TO COMPUTERS AND PROGRAMMING

SEMESTER	COURSE CODE	HOURS PER WEEK	CREDIT	EXAM HRS
1	1C01CSC	2	2	3

COURSE OUTCOME

CO1: Familiarize with the hardware components of a digital computer
CO2: Understand the basic idea of how data is represented in computers
CO3: Familiarize with types of software
CO4: Ability to design algorithmic solutions to problems

Unit I: Introduction to Computers

Characteristics of Computers, Computer System Hardware, Basic Concepts of CPU, ALU, Registers, Control Unit and System Bus, Components Inside a Computer Cabinet (Motherboard, BIOS, CMOS Chip, Ports and Interfaces, Expansion Slots, Memory Chips, Storage Devices, Processor - Basic functions), Computer Memory Representation, Memory Hierarchy, Basic Concepts of Cache Memory, Primary Memory (RAM and ROM), Secondary Memory Types (Working principle is not required).

(10 Hrs)

Unit II: Number System and Codes

Decimal, Binary, Hexa-Decimal and Octal Number Systems, Conversion Between Number Systems, Binary Arithmetic, Complements of Binary Numbers (1's Complement and 2's Complement), Signed Numbers, Floating Point Numbers, Binary Coded Decimal (8421 BCD Code, Applications, BCD Addition), Gray Code, ASCII Code, Unicode

(8 Hrs)

Unit III: Types of Software and Networking

System Software, Operating System (Functions of Operating Systems), Application Software, Software Acquisition (Retail, OEM, Demo, Shareware, Freeware, Open-Source Software), Computer Networks (Importance, Types of Networks – LAN, MAN, WAN).

(8 Hrs)

Unit IV: Introduction to Programming

Types of Computer Languages (Machine Language, Assembly Language, High-level Language), Basic Concepts of Compiler, Assembler, Interpreter, Linker and Loader.

Program Development Life Cycle, Algorithm, Flowcharts, Program Control Structures (Sequential, Selection, Loop), Programming Paradigms (Structured Programming, Basic Idea of Object-Oriented Programming), Characteristics of a Good Program

(10 Hrs)

Books for Study:

1. Anita Goel, Computer Fundamentals, Pearson
2. Thomas L. Floyd, Digital Fundamentals, 11th Edition, Pearson

Books for Reference:

1. Rajaraman V and Adabala N, Fundementals of Computers, PHI
2. Brian W Kernighan, D is for Digital: What a well-informed person should know about computers and communications, CreateSpace Independent Publishing Platform
3. Stewart Venit and Elizabeth Drake, Prelude to Programming (6th Edition), Pearson

Marks including choice:

Unit	Marks
I	17
II	13
III	9
IV	13

COMPLEMENTARY ELECTIVE COURSE II: PROGRAMMING IN C

SEMESTER	COURSE CODE	HOURS PER WEEK	CREDIT	EXAM HRS
2	2C02CSC	2	2	3

COURSE OUTCOME

CO1: Understand the building blocks of C programming language
CO2: Familiarize with program control structures in C
CO3: Learn procedural programming using functions
CO4: Understand user defined data types

Unit I: Introduction to C

C Character Set, Constants, Variables, Keywords, Instructions in C (Type Declaration, Arithmetic, Integer and Float Conversions), Operators in C (Arithmetic, Relational, Logical, Increment/Decrement, Assignment, Bitwise), Operator Precedence, Data Types (int, char, float, double, void), Compiling and Running C Programs in Linux.

(7 Hrs)

Unit II: Inputs and Control Statements

Formatted Console I/O Functions (printf, scanf), Escape Sequences, Unformatted Console I/O Functions (getch, putch, gets, puts), Decision control structures (Different forms of if statement), Conditional Operator, Case Control Structure (switch), Loop control structure (while, do-while, for), break and continue statements.

(10 Hrs)

Unit III: Functions and Pointers

User defined Functions (Advantages, Definition, Calling and Prototype), Library Functions, Pointers (Introduction to Pointers, Pointer Notation, Pointer Declaration and Initialization, Accessing Variable through Pointer), Call by Value and Call by Reference, Recursion

(10 Hrs)

Unit IV: Arrays, Strings and Structures

Arrays (Introduction, One Dimensional Arrays, Two Dimensional Arrays), Strings, Standard Library String Functions (strlen, strcpy, strcat, strcmp), Two-Dimensional Array of Characters. Storage Classes in C, Structures (Declaration, Initialization, Accessing Structure Elements), Array of Structures, Array Within Structure, Renaming Data Types with Typedef, C Preprocessors (#define, #include).

(9 Hrs)

Books for Study:

1. Yashavant P. Kanetkar, Let Us C, 16th Edition, BPB

Books for Reference:

1. Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Prentice Hall of India
2. E. Balaguruswamy, Programming in ANSI C, Tata McGraw-Hill
3. Byron Gottfried, Schaum's Outline of Programming with C, McGraw-Hill

Marks including choice:

Unit	Marks
I	10
II	16
III	16
IV	10

COMPLEMENTARY ELECTIVE COURSE III: WEB TECHNOLOGY WITH DATABASE MANAGEMENT SYSTEM

SEMESTER	COURSE CODE	HOURS PER WEEK	CREDIT	EXAM HRS
3	3C03CSC	3	2	3

COURSE OUTCOME

CO1: Develop skills to design a web page using HTML
CO2: Understand HTML Forms and CSS Styling
CO3: Develop skills to develop database and retrieve data using SQL
CO4: Learn basics of server-side programming with PHP

Unit I:HTML Basics

Introduction to WWW and HTML, Steps for hosting a website, Structure of HTML, HTML elements and attributes, Headings, Paragraphs, Formatting tags, line breaks, Comments, Links, Images, Lists, HTML5 Semantic Elements (header, footer, nav, section, article, nav, aside), HTML Tables.

(14 Hrs)

Unit II:HTML Forms and CSS

HTML Forms (input, select, textarea, button, datalist), Input types (text, password, submit, radio, checkbox, date, email), Input attributes (value, readonly, disabled, maxlength, autocomplete, list, min, max, placeholder), HTML5 form validation (required and pattern attribute of input type), Applying style to html using CSS (Inline, Internal and External CSS, Colors, Fonts, Borders, Padding, Applying style using class and id attribute)

(12 Hrs)

Unit III: Database Management System

Database Management System (Introduction, Simplified DBMS structure, advantages of DBMS, Database Administrators, Designers, End Users, System Analysts and Application Programmers), Relational Data Model (Domains, Attributes, Tuples, Relations), Relational Data Model Constraints (Domain Constraints, Key Constraints)

SQL Data Definition and Basic Data Types, Schema, DDL Statements (Create, Alter, Drop), Specifying Key Constraints in SQL, DML (Select, Insert, Update, Delete), Ordering Tuples, Renaming Attributes, Substring Pattern Matching and Arithmetic Operators, Aggregate Functions in SQL, Group By and Having, Joins (Inner and Outer)

(18 Hrs)

Unit IV: Introduction to PHP

Introduction to PHP, PHP basics (Variable, data types, Constants, Operators), Flow control (if, switch, while, for), Functions, Strings, Arrays, Form Handling (GET and POST methods), Connecting php to a database.

(10 Hrs)

Books for Study:

1. Julie C. Meloni, HTML and CSS in 24 Hours, Sams Teach Yourself (Updated for HTML5 and CSS3), Ninth Edition
2. RamezElmasri, Shamkant B. Navathe, Fundamentals of Database Systems, 7th Edition, Pearson
3. https://www.w3schools.com/php/

Books for Reference:

1. Powell, Thomas A. HTML & CSS: The Complete Reference. McGraw Hill Education; 5 edition.
2. Silberschatz, Abraham, Henry F. Korth, and ShashankSudarshan. Database system concepts. McGraw-Hill.
3. PHP: The Complete Reference, Steven Holzner, McGraw Hill Education
4. https://www.w3schools.com/css/
5. https://www.w3schools.com/html/

Marks including choice:

Unit	Marks
I	12
II	12
III	20
IV	8

COMPLEMENTARY ELECTIVE COURSE IV: COMPUTATION USING PYTHON

SEMESTER	COURSE CODE	HOURS PER WEEK	CREDIT	EXAM HRS
4	4C04CSC	3	2	3

COURSE OUTCOME

CO1: Learn Python for expressing computation
CO2: Familiarize with functions and modules in python
CO3: Understand object-oriented programming concepts
CO4: Learn the techniques for data visualization in python

Unit I: Basic Elements and Control Statements

Features of Python, Different Methods to Run Python, Basic Elements (Objects, Expressions, Numerical Types, Strings, Variables), Comments, Indentation in Python, Input and Output in Python, import function, Operators in Python, Branching (if, else, elif), Iteration (while, for), range and enumerate functions, Tuples, Lists, Sets, Dictionaries, Built-in methods of lists, sets and dictionaries, Mutable and Immutable Objects.

(16 Hrs)

Unit II: Functions, Modules and Exception Handling

Functions Definition, Function Calling, Function Arguments (Required, Keyword, Default), Recursion, Modules, Built-in Modules (math, statistics), Creating Modules, File Handling (Opening, Closing, Writing, Reading), Exceptions, Built-in Exceptions (IndexError, OverflowError, ZeroDivisionError, RuntimeError), Exception Handling.

(16 Hrs)

Unit III: Object Oriented Programming

Class Definition, Object Creation, Built-in Attribute Methods, Encapsulation, Data Hiding, Inheritance, Multi-Level Inheritance, Polymorphism (Method Overriding, Operator Overloading)

(10 Hrs)

Unit IV: Arrays and Data Visualization

Arrays in Python, Numpy Module, ndarray, Creating Arrays (array, zeros, ones, empty, linspace, arrange, random), Two-Dimensional Array, Indexing, Slicing, Iterating, Copying, Splitting, Shape Manipulation (reshape, transpose, resize), Arithmetic Operations on Arrays.

Data Visualization in Python (matplotlib Module, pyplot, plot(), hist, scatter, bar charts, Formatting, figure(), subplot(), text(), xlabel(), ylabel(), title(), Plotting Simple Mathematical Functions (sin x, x^2)

(12 Hrs)

Books for Study:

1. Taming Python By Programming, Dr. Jeeva Jose, Khanna Publishing
2. Introduction to Computation and Programming Using Python with Application to Understanding Data - John V. Guttag, PHI (2016)
3. https://www.numpy.org/devdocs/user/quickstart.html
4. https://matplotlib.org/users/pyplot_tutorial.html

Books for Reference:

1. https://www.tutorialspoint.com/python/
2. Introduction to Computer Science using Python - Charles Dierbach, Wiley (2015)
3. Python for Education by Ajith Kumar B P
4. https://docs.python.org/3/tutorial/index.html
5. Introduction to Computer Science and Programming Using Python Provided by Massachusetts Institute of Technology (MITx) - Available at : (https://www.edx.org/course/introduction-to-computer-science-and-programming-using-python-2)

Marks including choice:

Unit	Marks
1	15
2	15
3	10
4	12

COMPLEMENTARY ELECTIVE COURSE V: LAB 1 – PROGRAMMING IN C, WEB PROGRAMMING AND PYTHON PROGRAMMING

SEMESTER	COURSE CODE	HOURS PER WEEK	CREDIT	EXAM HRS
4	4C05CSC	2*	4	3

*Lab will be conducted for 2 hours each in I, II, III and IV semesters

COURSE OUTCOME

CO1: Achieve skills to use C language for problem solving
CO2: Understand SQL and basic web programming
CO3: Achieve skills to use Python for problem solving

Part I: C Programming

1. Write a program to receive an angle in degrees and check whether sum of the squares of sines and cosines of the angle is equal to 1. (Hint: Convert the angle in degrees to radians and apply mathematical functions).
2. Write a C program to check whether a year entered through the keyboard is leap year or not.
3. Write a program to reverse the digits of a positive integer number up to 5 digits. Display an error message if any other number is entered.
4. Write a program to enter numbers till the user wants. At the end, it should display the count of positive, negative and zeros entered.
5. Given the value of n, write a program to generate n Fibonacci numbers.
6. Create a menu driven calculator using switch statement. The menu should contain options for Addition, Subtraction, Multiplication, Division and Exit. The program should end only when the user enters the choice as Exit.
7. Create function which takes an integer value as parameter and returns 1 if the number is prime and 0 otherwise. Write a program which uses this function to generate first 100 prime numbers.
8. Write a program using recursion to find the factorial of a number.
9. Write a program to sort n numbers in ascending/descending order.
10. Write a program to check whether a string is palindrome or not.
11. Write a program to add two matrices. Display an error message if the matrices cannot be added due to incompatibility.

12. Create a structure student with membersroll_no, name and year_of_admn. Write a program to read n students into an array of the structure student. Write a function which takes year as argument and displays the names of students who joined that year. Get an input year from the user and display the student list using this function. (Hint: Make student array and number of students as global variables).

Part II: DBMS and Web Programming

To be updated

Part II: Python Programming

To be updated