

ST JOSEPH'S UNIVERSITY

BENGALURU-27



**DEPARTMENT OF COMPUTER SCIENCE AND COMPUTER
APPLICATION**

SYLLABUS FOR UNDERGRADUATE PROGRAMME - BSc

For Batch 2021-2024

DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

(BSc) (2021-2024)

SUMMARY OF CREDITS

<u>Semester 1</u>	Code Number	Title	No. of Hours of Instructions	Number of Hours of teaching per week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS121	Computer Fundamentals and Programming in C	52	04	04	40	60	100
Practical	CS1P1	C Programming Lab	52	04	02	25	25	50
Total Number of credits:			06					
<u>Semester 2</u>	Code Number	Title	No. of Hours of Instructions	Number of teaching hrs. /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS221	Data Structures Using C	52	04	04	40	60	100
Practical	CS2P1	Data Structures Lab	52	04	02	25	25	50
Total Number of credits:			06					

<u>Semester 3</u>	Code Number	Title	No. of Hours of Instructions	Number of teaching hrs. / week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS322	Object Oriented Programming Concepts and Programming in JAVA	52	04	04	40	60	100
Practical	CS3P1	Java Lab	52	04	02	25	25	50
Total Number of credits:			06					

<u>Semester 4</u>	Code Number	Title	No. of Hours of Instructions	Number of teaching hrs. /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS422	Database Management Systems and Software Engineering	52	04	04	40	60	100
Practical	CS4P1	DBMS Lab	52	04	02	25	25	50
Total Number of credits:			06					

<u>Semester 5</u>	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS5123	Programming in Python	42	03	03	40	60	100
Practical	CS5P1	Python Lab	52	04	02	25	25	50
Theory	CS5223	Computer Networks	42	03	03	40	60	100
Practical	CS5P2	Computer Networks Lab	52	04	02	25	25	50
Total Number of credits:			10					

<u>Semester 6</u>	Code Number	Title	No. of Hours of Instructions	Number of teaching Hrs. /week	Number of credits	Continuous Internal Assessment (CIA) Marks	End Semester Marks	Total marks
Theory	CS6123	Internet Technologies and C#	42	03	03	40	60	100
Practical	CS6P1	Internet Technologies and C# Lab	52	04	02	25	25	50
Theory	CS6223	Operating System	42	03	04	40	60	100
Practical	CS6P2	Major Project Lab	52	04	02	25	25	50
Total Number of credits:			10					

EXAMINATION AND ASSESSMENTS

THEORY

1. IA Weight age 40 %
2. End Semester Examination Weight age 60%

PRACTICAL/PROJECTS

1. IA Weight age 50 %
2. End Semester Examination Weight age 50%

Title	Credits	CA Marks	SE Marks	Total marks	Time Duration for ESE
Core Subjects	3 or 4	40	60	100	2 Hrs
Department Electives	3 or 4	40	60	100	2 Hrs
Open Electives	3	40	60	100	2 Hrs
Practicals	2	25	25	50	2 Hrs

QUESTION PAPER PATTERN CORE/DEPARTMENT ELECTIVE

The question papers of the theory examinations should follow the pattern specified below:

Section	Marks for each question	Number Of Questions		Total Marks
		Total	Should Answer	
A	2	5	5	10
B	4	7	5	20
C		4	3	30
(a+b)	10			

Total Marks 60

OPEN ELECTIVE

60 MCQS – One mark each.

INTERNAL ASSESSMENT FORMAT

THEORY:

1. First Activity (should be multiple activities): 10 marks
2. Mid Semester Test: 20 marks (Test will be for 30 marks. It will be converted to 20)
3. Second Activity (should be multiple activities): 10 marks
4. Third Activity only for the first semester students who had scored less than 5 out of 10. Will be conducted after activity 2. Maximum marks shall be 5.

Total : 40 marks

MID SEMESTER TEST QUESTION PAPER PATTERN

Section	Marks for each question	Number Of Questions		Total Marks
		Total	Should Answer	
A	2	6	5	10
B	5	5	4	20
				30

PRACTICALS:

Every practical class the student should be assessed.

1. **Writing the observation book** **10 marks**
 2. **Executing the programs and viva voce** **5 marks**
 3. **Record writing** **5 marks**
- Total** **25 marks**

PIA - 25 marks

PROJECT LAB

Presentation / demo must be carried out in all the lab sessions in the whole semester for internal assessment of the project. In each lab session a student is evaluated for 25 marks.

Presentation /Demo-15

Viva Voce-10

Semester: I

Course Code: CS121	Course Title: Computer Fundamentals and Programming in C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours:52	Formative Assessment Marks:40
Exam Marks: 100	Exam Duration: 03 Hrs

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Course Content

Content	Hours
Unit - 1	
Fundamentals of Computers: Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organisation of a Digital Computer; Memories -primary memory, secondary memory, cache memory. Number Systems – different types, conversion from one number system to another; Computer Codes – BCD, Gray Code, ASCII and Unicode; Boolean Algebra – Boolean Operators with Truth Tables; Types of Software – System Software and Utility Software; Computer Languages - Machine Level, Assembly Level & High Level Languages, Translator Programs – Assembler, Interpreter and Compiler; Planning a Computer Program - Algorithm, Flowchart and Pseudo code with Examples.	8
Unit - 2	
Introduction to C Programming: Over View of C; History and Features of C; Structure of a CProgram with Examples; Creating and Executing a C Program; Compilation process in C. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, andvariables; Data types; Declaration & initialization of variables; Symbolic constants. Input and output with C: Formatted I/O functions - <i>printf</i> and <i>scanf</i> , control stings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functionsto read anddisplay single character and a string - <i>getchar</i> , <i>putchar</i> , <i>gets</i> and <i>puts</i> functions.	10

Unit - 3	
C Operators & Expressions: Arithmetic operators; Relational operators; Logical operators; Assignment operators; Increment & Decrement operators; Bitwise operators; Conditional operator; Special operators; Operator Precedence and Associativity; Evaluation of arithmetic expressions; Type conversion.	12
Control Structures: Decision making Statements - <i>Simple if, if_else, nested if_else, else_if ladder, Switch-case, goto, break & continue</i> statements; Looping Statements - Entry controlled and Exitcontrolled statements, <i>while, do-while, for</i> loops, Nested loops.	
Unit - 4	
Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; TwoDimensional arrays - Declaration, Initialization and Memory representation. Strings: Declaring & Initializing string variables; String handling functions - <i>strlen, strcmp, strcpy and strcat</i> ; Character handling functions - <i>toascii, toupper, tolower, isalpha, isnumeric</i> etc. User Defined Functions: Need for user defined functions; Format of C user defined functions; Components of user defined functions - return type, name, parameter list, function body, return statement and function call; Categories of user defined functions - With and without parameters and return type	12
Unit - 5	
Pointers in C: Understanding pointers - Declaring and initializing pointers, accessing address and value of variables using pointers; Pointers and Arrays; Pointer Arithmetic; Advantages and disadvantages of using pointers; User defined data types: Structures - Structure Definition, Advantages of Structure, declaring structure variables, accessing structure members, Structure members initialization, comparing structure variables, Array of Structures; Unions - Union definition; difference between Structures and Unions.	10

Text Books

1. Pradeep K. Sinha and Priti Sinha: Computer Fundamentals (Sixth Edition), BPB Publication
2. E. Balgurusamy: Programming in ANSI C (TMH)

References

1. Kamthane: Programming with ANSI and TURBO C (Pearson Education)
2. V. Rajaraman: Programming in C (PHI - EEE)
3. S. ByronGottfried: Programming with C (TMH)
4. Kernighan & Ritchie: The C Programming Language (PHI)
5. Yashwant Kanitkar: Let us C
6. P.B. Kottur: Programming in C (Sapna Book House)

BLUE PRINT

Unit Nos	No. of Hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	10	10
Unit 2	8	9
Unit 3	12	19
Unit 4	12	19
Unit 5	10	19
TOTAL	52	76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CS1P1	Course Title: C Programming Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration:03 Hrs

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Confidently operate Desktop Computers to carry out computational tasks
- Understand working of Hardware and Software and the importance of operating systems
- Understand programming languages, number systems, peripheral devices, networking, multimedia and internet concepts
- Read, understand and trace the execution of programs written in C language
- Write the C code for a given problem
- Perform input and output operations using programs in C
- Write programs that perform operations on arrays

Practice Lab

The following activities be carried out/ discussed in the lab during the initial period of the semester.

1. Basic Computer Proficiency
 - a. Familiarization of Computer Hardware Parts
 - b. Basic Computer Operations and Maintenance.
 - c. Do's and Don'ts, Safety Guidelines in Computer Lab
2. Familiarization of Basic Software – Operating System, Word Processors, Internet Browsers, Integrated Development Environment (IDE) with Examples.
3. Type Program Code, Debug and Compile basic programs covering C Programming fundamentals discussed during theory classes.

Programming Lab

Part A:

1. Write a C Program to read radius of a circle and to find area and circumference
2. Write a C Program to read three numbers and find the biggest of three
3. Write a C Program to demonstrate library functions in *math.h*
4. Write a C Program to check for prime
5. Write a C Program to generate n primes
6. Write a C Program to read a number, find the sum of the digits, reverse the number and check it for palindrome
7. Write a C Program to read numbers from keyboard continuously till the user presses 999 and to find the sum of only positive numbers
8. Write a C Program to read percentage of marks and to display appropriate message (Demonstration of else-if ladder)
9. Write a C Program to find the roots of quadratic equation (demonstration of switch-case statement)
10. Write a C program to read marks scored by n students and find the average of marks (Demonstration of single dimensional array)
11. Write a C Program to remove Duplicate Element in a single dimensional Array
12. Program to perform addition and subtraction of Matrices

Part B:

1. Write a C Program to find the length of a string without using built in function
2. Write a C Program to demonstrate string functions.
3. Write a C Program to demonstrate pointers in C
4. Write a C Program to check a number for prime by defining *isprime()* function
5. Write a C Program to read, display and to find the trace of a square matrix
6. Write a C Program to read, display and add two m x n matrices using functions
7. Write a C Program to read, display and multiply two m x n matrices using functions
8. Write a C Program to read a string and to find the number of alphabets, digits, vowels, consonants, spaces and special characters.
9. Write a C Program to Reverse a String using Pointer
10. Write a C Program to Swap Two Numbers using Pointers
11. Write a C Program to demonstrate student structure to read & display records of n students.
12. Write a C Program to demonstrate the difference between structure & union.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course

Semester: II

Course Code: CS221	Course Title: Data Structures using C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 40
Exam Marks: 100	Exam Duration: 03 Hours

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Describe how arrays, records, linked structures, stacks, queues, trees, and graphs are represented in memory and used by algorithms
- Describe common applications for arrays, records, linked structures, stacks, queues, trees, and graphs
- Write programs that use arrays, records, linked structures, stacks, queues, trees, and graphs
- Demonstrate different methods for traversing trees
- Compare alternative implementations of data structures with respect to performance
- Describe the concept of recursion, give examples of its use
- Discuss the computational efficiency of the principal algorithms for sorting and searching

Course Content

Content	Hours
Unit - 1	
Introduction to data structures: Definition; Types of data structures - Primitive & Non-primitive, Linear and Non-linear; Operations on data structures. Dynamic memory allocation: Static & Dynamic memory allocation; Memory allocation and de-allocation functions - <i>malloc</i> , <i>calloc</i> , <i>realloc</i> and <i>free</i> . Algorithm Specification, Performance Analysis, Performance Measurement-Asymptotic notations.	10
Unit - 2	
Arrays: Basic Concepts – Definition, Declaration, Initialization, Operations on arrays; Types of arrays; Arrays as abstract data types (ADT); Representation of Linear Arrays in memory; Traversing linear arrays; Inserting and deleting elements; Sorting – Selection sort, Bubble sort, Quick sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays;	10
Unit - 3	

Stacks: Basic Concepts – Definition and Representation of stacks; Operations on stacks; Applications of stacks; Infix, postfix and prefix notations; Conversion from infix to postfix using stack; Evaluation of postfix expression using stack; Application of stack in function calls.	10
---	----

Unit – 4	
Queues: Basic Concepts – Definition and Representation of queues; Types of queues- Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues; Linked list: Basic Concepts – Definition and Representation of linked list, Types of linked lists - Singly linked list, Doubly linked list, Header linked list, Circular linked list; Representation of Linked list in Memory; Operations on Singly linked lists – Traversing, Searching, Insertion, Deletion; Memory allocation; Garbage collection.	12
Unit – 5	
Trees: Definition; Tree terminologies – node, root node, parent node, ancestors of a node, siblings, terminal & non-terminal nodes, degree of a node, level, edge, path, depth; Binary tree: Type of binary trees - strict binary tree, complete binary tree, binary search tree and heap tree; Array and pointer representation of binary tree. Traversal of binary tree; preorder, inorder and postorder traversal;	10

Text Books

1. Ellis Horowitz and Sartaj Sahni: Fundamentals of Data Structures

References

1. Tanenbaum: Data structures using C (Pearson Education)
2. Kamathane: Introduction to Data structures (Pearson Education)
3. Y. Kanitkar: Data Structures Using C (BPB)
4. Kottur: Data Structure Using C
5. Padma Reddy: Data Structure Using C
6. Sudipa Mukherjee: Data Structures using C – 1000 Problems and Solutions (McGraw Hill Education, 2007)

BLUE PRINT

Unit Nos	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	10	10
Unit 2	10	9
Unit 3	10	19
Unit 4	12	19
Unit 5	10	19
TOTAL	52	76
Maximum marks for the paper (Excluding bonus questions)		
=60		

Course Code: CS2P1	Course Title: Data Structures Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 44	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hrs

Programming

LabPart A:

1. Write a C Program to find GCD using recursive function
2. Write a C Program to display Pascal Triangle using binomial function
3. Write a C Program to generate n Fibonacci numbers using recursive function.
4. Write a C Program to implement Towers of Hanoi.
5. Write a C Program to implement dynamic array, find smallest and largest element of the array.
6. Write a C Program to create two files to store even and odd numbers.
7. Write a C Program to create a file to store student records.
8. Write a C Program to read the names of cities and arrange them alphabetically.
9. Write a C Program to sort the given list using selection sort technique.
10. Write a C Program to sort the given list using bubble sort technique.

Part B:

1. Write a C Program to sort the given list using insertion sort technique.
2. Write a C Program to sort the given list using quick sort technique.
3. Write a C Program to sort the given list using merge sort technique.
4. Write a C Program to search an element using linear search technique.
5. Write a C Program to search an element using recursive binary search technique.
6. Write a C Program to implement Stack.
7. Write a C Program to convert an infix expression to postfix.
8. Write a C Program to implement simple queue.
9. Write a C Program to implement linear linked list.
10. Write a C Program to display traversal of a tree.

Evaluation Scheme for Lab Examination

III SEMESTER

Course Code: CA 322	Course Title: Object Oriented Programming Concepts and Programming in JAVA
Course Credits: 04	Hours / Week : 04
Total Contact Hours: 52	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 2 hrs.

Course Out comes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand the features of Java and the architecture of JVM
Write, compile, and execute Java programs that may include basic data types and control flow constructs and how typecasting is done.
- Identify classes, objects, members of a class and relationships among them needed for a specific problem and demonstrate the concepts of polymorphism and inheritance.
- The students will be able to demonstrate programs based on interfaces and threads and explain the benefits of JAVA"s Exceptional handling mechanism compared to other Programming Language
Write, compile, execute Java programs that include GUIs and event driven programming and also programs based on files

Course Content

Content	Hours
Unit -1	
Introduction to Java : Basics of Java programming, Data types, Variables, Operators, Control structures including selection, Looping, Java methods, Overloading, Math class, Arrays in java.	6
Unit -2	
Object Oriented Programming Concepts: Concept of programming paradigm, procedural paradigm and draw backs, object oriented paradigm concepts, OOP features – inheritance, polymorphism, encapsulation, abstraction and others (with examples), comparison of object oriented paradigm and other paradigms. OOP as a way of viewing world – Members and methods, Responsibilities, Classes and Instances, Summary of Object-Oriented concepts, Introducing classes, Methods and Classes, Constructors, Finalize, Visibility modifiers, Inbuilt classes like String, Character, String Buffer, File, this reference.	10
Unit -3	

Inheritance and Polymorphism: Inheritance in java, Super and subclass, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	10
Unit -4	
Event and GUI programming: Event handling in java, Event types, Mouse and key events, GUI Basics, Panels, Frames, Layout Managers: Flow Layout, Border Layout, Grid Layout, GUI components like Buttons, Check Boxes, Radio Buttons, Labels, Text Fields, Text Areas, Combo Boxes, Lists, Scroll Bars, Sliders, Windows, Menus, Dialog Box, Applet and its life cycle, Introduction to swing, Exception handling mechanism. Contd.	10
Unit -5	
I/O Programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files.	6
Unit -6	
Multithreading and Exception handling in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to Java Beans and Network Programming.	10

Text Books:

1. "Introduction to Java Programming" by Daniel Liang
2. Programming with Java, By E Balagurusamy – A Primer, Fourth Edition, Tata McGraw Hill Education Private Limited.
3. Core Java Volume I–Fundamentals, By Cay S. Horstmann, Prentice Hall
4. Object Oriented Programming with Java :Somashekara, M.T., Guru, D.S., Manjunatha, K.S

Reference Books:

1. Java 2-The Complete Reference–McGraw Hill publication.
2. Java - The Complete Reference, 7th Edition, By Herbert Schildt McGraw Hill publication.

BLUEPRINT

Unit Nos	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	6	9
Unit 2	10	10
Unit 3	10	19
Unit 4	10	19
Unit 5	6	19
Unit 6	10	19
TOTAL		76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CA3P1	Course Title: JAVA Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 03 Hours

Course Outcomes (COs):

- After completing this course, student will be able to:
- Implement Object Oriented programming concept using basic syntax of control Structures.
- Identify classes, objects, members of a class and the relationships among them needed for finding the solution for the specific problem.
- Demonstrate show to achieve reusability using inheritance.
- Demonstrate understanding and use of interfaces, packages, different exception handling mechanisms and concept of multithreading for robust faster and efficient application development.
- Identify and describe common user interface components to design GUI in Java using Applet & AWT along with response to events.

Practice Lab List

1. Program to print the following triangle of numbers

```
1
12
123
1234
12345
```

2. Program to simple java application, to print the message, "Welcome to java".

3. Program to display the month of a year. Months of the year should be held in an array.

4. Program to find the area of rectangle.

5. Program to demonstrate a division by zero exception

Program to create a user defined exception say Pay out of Bounds.

Programming Lab

PART A: Java Fundamentals OOPs in Java

1. Program to assign two integer values to X and Y. Using the „if“ statement the Output of the program should display a message whether X is greater than Y.
2. Program to list the factorial of the numbers 1 to 10. To calculate the factorial value, use while loop.(Hint Fact of 4=4*3*2*1).
3. Program to add two integers and two float numbers. When no arguments are supplied, give a default value to calculate the sum. Use function overloading.
4. Program to perform mathematical operations. Create a class called Add Sub with methods to add and subtract. Create another class called Mul Div that extends from Add Sub class to use the member data of the super class. Mul Div should have methods to multiply and divide A main function should access the methods and perform the mathematical operations.
5. Program with class variable that is available for all instances of a class. Use static variable declaration. Observe the changes that occur in the object's member variable values.
6. Program
 - i. To find the area and circumference of the circle by accepting the radius from the user.
 - ii. To accept a number and find whether the number is Prime or not.
7. Program to create a student class with following attributes;
Enrollment No: Name, Mark of sub1, Mark of sub2, mark of sub3, Total Marks. The three marks must be calculated only when the student passes in all three subjects. The pass mark for each subject is 50.If a candidate fails in any one of the subject, his total mark must be declared as zero. Using this condition write a constructor for this class. Write separate functions for accepting and displaying student details. In the main method create an array of three student objects and display the details.
 - In a college first year class are having the following attributes Name of the class (BCA, B.Com, BSc), Name of the staff No of the students in the class, Array of students in the class
 - Define a class called first year with above attributes and define a suitable constructor. Also write a method called best Student () which process a first-year object and return the student with the highest total mark. In the main method define a first-year object and find the best student of this class.
8. Program to define a class called employee with the name and date of appointment. Create ten employee objects as an array and sort them as per their date of appointment.ie, print them as per their seniority.

9. Create a package „student. Full time. BCA „in your current working directory
 - Create a default class student in the above package with the following attributes: Name, age, sex.
 - Have methods for storing as well as displaying.

PARTB: Exception Handling & GUI Programming.

1. Program to catch Negative Array Size Exception. This exception is caused when the array is initialized to negative values.
2. Program to handle Null Pointer Exception and use the “finally” method to Display a message to the user.
3. Program which create and displays a message on the window.
4. Program to draw several shapes in the created window.
5. Program to create an applet and draw grid lines.
6. Program which creates a frame with two buttons father and mother. When we click the father button the name of the father, his age and designation must appear. When we click mother similar details of mother also appear.
7. Create a frame which displays your personal details with respect to a button click
8. Create a simple applet which reveals the personal information of yours.
9. Program to move different shapes according to the arrow key pressed.
10. Demonstrate the various mouse handling events using suitable example.
11. Program to create menu bar and pull-down menus.

Note: Student has to execute a minimum of 10 programs in each part to complete the Lab course.

SEMESTER IV

Course Code: CS422	Course Title: DATABASE MANAGEMENT SYSTEMS AND SOFTWARE ENGINEERING
Course Credits:04	Hours/Week:04
Total Contact Hours: 52	Formative Assessment Marks:40
Exam Marks:60	Exam Duration: 02 Hrs.

Course Outcomes (COs):

After completing this course satisfactorily, a student will be able to:

- Understand database concepts and database management system software.
- Model an application's data requirements using conceptual modeling tools like ER diagrams.
- Write SQL commands to create tables and indexes, insert/update/delete data, and Query data in a relational DBMS.
- Understand the importance of software life cycle.
- Understand various aspects of design, coding, testing and reusability in Software development.

Course Content

Content	Hours
DATA BASE MANAGEMENT SYSTEM	
Unit -1	
INTRODUCTION: Data, database, DBMS, Disadvantages of File oriented systems, Advantages of DBMS, Database users, Database Languages, Characteristics of Database, Role of DBA, Data Abstraction (Views) – Logical, Conceptual & Physical, Data independence – physical and logical independence. DATA MODELS: Introduction to Data Models: E-R model, Relational model, network model and hierarchical model.	10
Unit -2	
RDBMS: Relational database concepts, attribute, tuple, types of attributes, single, multi-valued, stored, derived etc., keys, primary, index, candidate, alternate, foreign, Relationships, Relational algebra operations, union, intersection, difference, Cartesian product, selection, projection, join, division, relational calculus, Normalization and its properties, I, II, III Normal forms and BCNF.	10

Unit - 3	
DDL and DML in SQL: DDL commands - create table/views/index, drop, alter, DML commands - select, insert, delete, update, etc., DCL commands - grant, revoke, commit, TCL commands, SQL - query, sub-query, nested query, Joins - natural, inner, outer join.	10
Unit - 4	
SOFTWARE ENGINEERING: Software and Software Engineering: Defining Software, Software Application Domains, Software Engineering, Software Process, Software Engineering Practice, Software Myths, Agile Development: Agility, Agility and the cost of change, Agile Process, Extreme Programming and Other Agile Process. Models. Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Developing the use cases, Building the Requirements Model, Negotiating Requirements and Validating Requirements.	12
Unit - 5	
Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, Design Concepts: The Design Process, Design Concepts, The Design Model, Architectural Design, Component-Level Design, User Interface Design, Pattern-Based Design, Quality Concepts: Software Quality, Review Techniques, Software Quality Assurance Software Testing Strategies: A Strategic Approach to Software Testing, Strategic Issues, Test Strategies for Conventional Software, System Testing, The Art of Debugging, Software Testing Fundamentals, White box Testing, Block-Box Testing.	10

Text Books:

1. "Remez Elmasri and Shamkant B. Navathe, "Fundamentals of Database Systems", 5th Edition, Pearson Education, 2007.
2. Abrahamsi. Silberschatz, Henry. F. Korth, S. Sudarshan, "Database System Concepts", 6th Edition, McGraw Hill, 2012.
3. Sundarraman, Oracle 9i programming A Primer,1/e Pearson Education.
4. Roger S. Pressman, "Software Engineering, A Practitioner"s approach", 7th Edition, McGRAW-HILL Publication, 2010.
5. Pankaj Jalote, "An integrated approach to Software Engineering", 3rd Edition, Narosa Publishing House, 2013.

Reference Books:

1. C.J.Date, "Introduction to Database Systems", Eight Edition, Addison Wesley, 2003.
2. Karate, "Introduction to Database Management System", Pearson India, 2004
3. Ian Sommerville, "Software Engineering", 9th Edition, Pearson Education Ltd, 2010

BLUEPRINT

Unit No.	No. of hours	Total marks for which questions are to be asked (including bonus questions)
Unit 1	10	10
Unit 2	10	9
Unit 3	10	19
Unit 4	12	19
Unit 5	10	19
TOTAL	52	76
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code:CS4P1	Course Title: DATABASE LAB
Course Credits:02	Hours/Week:04
Total Contact Hours:52	Formative Assessment Marks:25
ExamMarks:25	Exam Duration: 03 Hours

Course Outcomes (COs):

This course will enable students to

- To understand basic database concepts, applications, data models, schemas and instances.
- Describe the basics of SQL and construct queries using SQL. Emphasize the importance of normalization in databases.

SQL Programming

- Design, develop, and implement the specified queries for the following problems using Oracle, MySQL, MS SQL Server, or any other DBMS under LINUX/Windows environment.
- Create Schema and insert at least 5 records for each table. Add appropriate database constraints.
 1. Perform the following:
 - i) Creating a Database,
 - ii) Viewing all databases,
 - iii) Viewing all Tables in a Database,
 - iv) Creating Tables (With and Without Constraints),
 - v) Delete Table
 - vi) Rename Table.
 2. Write SQL Queries involving:
 1. Date Functions,
 2. String Functions and
 3. Math Functions.
 3. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG and populate the table STATION with a few rows:
 - i) Write a SQL query to look at table STATION in undefined order
 - ii) Write a SQL query to select Northern stations (latitude > 39.7)
 - iii) Write a SQL query to select only ID, CITY, and STATE columns
 - iv) Write a SQL query to select only ID, CITY, and STATE columns where Longitude >45.

4. Create a table STATION to store information about weather observation stations with fields: ID (Primary Key), CITY, STATE, LAT, LONG. Duplicate ID fields are not allowed. Populate the table STATION with a few rows.
Create another table called STATS to store normalized temperature and precipitation data:
 - ID field must match some STATION table ID (so that name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
 - i) Populate the table STATS with some statistics for January and July.
 - ii) Write a SQL query to look at table STATS in undefined order.
 - iii) Write a SQL query to look at table STATS, picking up location information by joining with table STATION on the ID column.
 - iv) Write a SQL query to look at the table STATS, ordered by month and greatest rainfall, with columns rearranged.
5. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows. Create another table called STATS to store normalized temperature and precipitation data.
 - ID field must match some STATION table ID (so name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
 - i) Write a SQL query to look at temperatures for July from table STATS, lowest temperatures first, picking up city name and latitude by joining with table STATION on the ID column
 - ii) Write a SQL query to show MAX and MIN temperatures as well as average rainfall for each station
 - iii) Write a SQL query (with sub query) to show stations with year-round average temperature above 50 degrees.
 - Rows are selected from the STATION table based on related values in the STATS table.
6. Create table called STATS to store normalized temperature and precipitation data.
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
 - i) Create a view (derived table or persistent query) to convert Fahrenheit to Celsius and inches to centimeters
 - ii) Add new column rainfall_centimeter to the table STATS.
 - iii) Insert values into rainfall_centimeter from the view

- iv) Delete Column rainfall from table STATS
7. Create table called STATS to store normalized temperature and precipitation data:
- Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- i) Write a SQL query to look at table STATS in a metric light (through the new view).
- ii) Write a SQL metric query restricted to January below-freezing (0 Celsius) data, sorted on rainfall.
8. Create table called STATS to store normalized temperature and precipitation data.
- i) Update all rows of table STATS to compensate for faulty rain gauges known to read 0.01 inches low
- ii) Update one row, ID 44's July temperature reading, to correct a data entry error
- iii) Make the above changes permanent
- iv) Undo that update
9. Create a table STATION to store information about weather observation stations with fields: ID(Primary Key), CITY, STATE, LAT, LONG. No duplicate ID fields allowed. Populate the table STATION with a few rows.
- Create another table called STATS to store normalized temperature and precipitation data.
- ID field must match some STATION table ID (so name and location will be known).
 - Allowable ranges will be enforced for other values.
 - No duplicate ID and MONTH combinations.
 - Temperature is in degrees Fahrenheit.
 - Rainfall is in inches.
- i) Delete data from STATION table where longitude is >90
- ii) Delete July data from STATS table where longitude is >90
- iii) Increase the size for the column CITY with the following information:-
- | COLUMNNAME | DATATYPE(SIZE) |
|------------|----------------|
| ----- | ----- |
| CITY | VARCHAR (25) |
- iv) Modify the column name of LONG to LONGITUDE present in the STATION table and verify the result.
10. Create Table INSTRUCTOR with the following fields: InstructID, InstructName, Department, Salary.
- i) Find instructors whose salary is more than the salary of any employee from department „Physics“.

- ii) Find the instructor name and department name of all instructors working in a department with any instructor whose name contains the letter "S".
- iii) Find the name and department name, of instructors whose salary is more than all their colleagues' salaries in the same department.

11. Create table INSTRUCTOR with the following fields: InstuctID, InstructName, Department, Salary.

Create another table called STUDENT with the following fields: StudentID, StudentName, Department, InstuctID

- i) Find the names of all instructors whose salary is greater than at least one instructor in the Finance department.
- ii) Find all instructors whose salary is less than the salary of all instructors in the Computer Science department and whose department name is not Computer Science.
- iii) Find the student name and department name of all student who study in a department with any student whose name contains the letter "S".

12. Consider the following schema for a Library Database:

BOOK (Book_id, Title, Publisher_Name, Pub_Year)

BOOK_AUTHORS (Book_id, Author_Name)

PUBLISHER (Publisher_Name , Address, Phone)

BOOK_COPIES (Book_id, Branch_id, No-of_Copies)

BOOK_LENDING (Book_id, Branch_id, Card_No, Date_Out, Due_Date)

LIBRARY_BRANCH (Branch_id, Branch_Name, Address).

- i) Draw the E_R Diagram for the Library database.
- ii) Write SQL query to Retrieve details of all books in the library: id, title, name of publisher, authors, number of copies in each branch.
- iii) Write an SQL query to compute the total number of books based on publisher name order by year.

13. Consider the schema for Movie Database:

ACTOR(Act_id, Act_Name, Act_Gender)

DIRECTOR(Dir_id, Dir_Name, Dir_Phone)

MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id)

MOVIE_CAST(Act_id, Mov_id, Role)

RATING(Mov_id, Rev_Stars)

- i) Draw the E_R Diagram for the Movie database.
- ii) Write an SQL query to find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. (Use Inner Join)

- iii) Write an SQL query to retrieve all the actors and any movies they have acted in (Use left Join)
- iv) Write an SQL query to retrieve all the directors and any movies they have directed.(Use right outer join)
- (v) Write an SQL query to retrieve all the directors name and all themovies (use Full join)

SEMESTER V

Course Code: CS5123	Course Title: Programming in Python
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 02Hrs

Course Objectives:

1. The course is designed to provide essential programming components like variables, conditional logic, looping, and functions.
2. To understand and begin to implement code. Collaborate with user input to develop entertaining and engaging programs.

Course Outcomes (COs):

This course focuses on structure and components of Python program. It emphasis on designing object-oriented program with Python classes. At the end of this course the student will be able to:

1. Examine Python syntax and semantics and be fluent in the use of Python flow control and functions.
2. Get the knowledge in solving the complex problems with optimized solutions.
3. Understand the concept of object-oriented programming paradigm and its applications.
4. Understand the data manipulations and built-in functions.
5. Integrate and connect various real time projects in achieving multi-disciplinary fields.

Content	Hours
Unit - 1	
Basics of Python program Printing text, printing the result of a calculation, Strings in Python, The Escape Character, Variables and Types, Numeric Data Types in Python, Numeric Operators, Expressions, Integer division, Operator Precedence. Programming examples for each concept. String and Slice: The String Data Type, Negative Indexing in Strings, Slicing, Slicing with Negative Numbers, Using a Step in a Slice, Slicing Backwards, String Operators, String Replacement Fields, String Formatting. Programming examples for each concepts.	8
Unit - 2	
Program Flow Control & Functions Conditional blocks using if, else and elif, for loops in, for loop using ranges, use of while loops in python, Loop manipulation using pass, continue, break and else. Organizing python codes using functions, organizing python projects into modules, importing own module as well as external modules, Understanding Packages.	8
Unit - 3	

String, List, Dictionary & File handling: Understanding in-built string methods, List manipulation and Dictionary manipulation. Programming using string, list and dictionary in-built functions. Reading files in python, writing files in python, Understanding read and write functions, manipulating file pointer using seek, Programming using file operations.	9
Unit - 4	
OO Programming & Regular Expressions: Concept of class, object and instances, constructor, class attributes and destructors, Inheritance, overlapping and overloading operators, Adding and retrieving dynamic attributes of classes. Pattern matching and searching, Power of pattern searching using regex in python, Pattern finding programs using regular expression.	9
Unit - 5	
Exception Handling & Database Interaction: Avoiding code break using exception handling, Safe guarding file operation using exception handling, Handling and helping developer with error code. SQL Database connection using python, Creating and searching tables, Reading, storing and updating information on database.	8

REFERENCES:

1. "Python Programming: A Complete Guide for Beginners to Master, Python Programming Language" by Brian Draper
2. "Python Programming for Beginners: Python Programming Language Tutorial" by Joseph Joyner

BLUEPRINT

Title of the paper: Programming in Python

Chapter	Number of Hours	Total marks for which the question are to be asked (including bonus questions)
UNIT I	8	6
UNIT II	8	6
UNIT III	9	23
UNIT IV	9	23
UNIT V	8	20
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code:CS5P1	Course Title: Python Programming Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks:25	Exam Duration: 02Hours

Course Outcomes (COs):

This course will enable students to

- Understand the menu driven program and its approach in making the program more interactive.
- Understand the need of Object-Oriented Programming Paradigm.
- Analysis the various features like built library and data structure to manage the various types of data.

Python Program List

(Do not use inbuilt functions wherever possible)

1. Create a Menu Driven Program using Python that takes a number as input and prints the following:
 - a. Reverse of the given number
 - b. Sum of Digits
 - c. Palindrome or Not
 - d. Series "1 + 2 +.... +n="
 - e. Smallest Divisor of the given number
2. Create a Menu Driven Program using Python that takes a string as input and prints the following:
 - a. Reverse the string
 - b. Length of the string
 - c. Number of Vowels in a String
 - d. Number of Uppercase Letters and Lower-Case Letters in a String
 - e. New string that does not have any duplicate characters
3. Create a Menu Driven Program using Python that takes a List of Numbers from the user and does the following:
 - a. Print the Second Largest Number in the List
 - b. Swap the first and last items of that list and print
 - c. Remove the duplicate members in that list and print.
 - d. Print the number Occurring Odd Number of Times in a List
 - e. Print the list which has the square of the numbers in the given List.
4. Create a Menu Driven Program using Python that takes two lists from the user and does the following:
 - a. Print the Union of the Lists
 - b. Print the Intersection of the Lists
 - c. Print the Merged and sorted List
 - d. Print the sum of all the numbers in both the lists
 - e. Print the average of all numbers in both the lists

5. Create a Menu Driven Program using Python that does the following:
 - a. Creates and Prints a Dictionary with numbers and their squared values till 10
 - b. Print the sum of all values in the Dictionary
 - c. Ask a number from the user and remove it from the Dictionary and Print
 - d. Print all items in the dictionary whose values are above 50

6. Create a Menu Driven Program using Python that takes a number from the user does the following using Recursion:
 - a. Print Fibonacci Series
 - b. Print Factorial of that number
 - c. Print the Sum of the digits in that number
 - d. Print if the given number is Prime Number or Not

7. Create a Menu Driven Program using Python that does the following file operations:
 - a. Create a File named 'zen_of_python.txt and write the data that comes from import this into the created file (Use Exclusive Creation).
 - b. Print all the data in that file
 - c. Print total number of words in that file.
 - d. Capitalize First Letter of Every word in that file
 - e. Print the contents of the file in reverse order

8. Create a Menu Driven Program using Python that does the following:
 - a. Print the current Date and Time (DD-MM-YYYY HH:MM: SS AM/PM)
 - b. Print all the Sundays of a specified year
 - c. Print new Date after adding 28 days to the current Date
 - d. Print the Day of the given Date
 - e. Calculate Age by taking the Date of Birth

9. Create a Menu Driven Program in Python that takes a string and checks if it follows the regular expressions given below:
 - a. Matches a string that has an 'a' followed by anything, ending in 'b'
 - b. Print only a 4-digit number from a message as string
 - c. Convert a date of yyyy-mm-dd format to dd-mm-yyyy format
 - d. Find all five characters long word in a string
 - e. Split a string with multiple delimiters

10. Create a User Defined Exception for Age, that is raised if the given number is not above 18 and below 100. Use this exception in a program where it is raised programmatically after accepting a Number.

11. Create a Menu Driven Program in Python that prints the following pattern for the given number: (Refer Pattern List in Google Classroom)
 - a. Pattern #2: Inverted Pyramid of Numbers
 - b. Pattern #6: Reverse Pyramid of Numbers
 - c. Pattern #8: Unique Pyramid Pattern of Digits.
 - d. Pattern #11: Pyramid of Horizontal Tables,
 - e. Pattern #15: Downward Triangle Pattern of Stars

12. Create a Menu Driven Program in Python that does the following:

- a. Accept a sentence and reverse each word in it
- b. Take a string input and sort them based on the number of characters in each word and mention the count of letters in a word at the end.
- c. Take a number as string type and convert it into an integer without using inbuilt functions
- d. Take a String and append a digit after every character with total count of that character in that string (ignore case)

13. You are given input file input_digits_P1.txt and input_words_P1.txt which contains Comma separated random integers. E.g. 2, 3, 10, 8, 7, 20,11,25,30,17,12.

- Read the integers from the input_digits_P1.txt file and calculate the Sum of the Odd and Even numbers.
- Create a file with the name sum_of_odd_numbers.txt and store the sum of odd Numbers in that file. E.g. 63
- Create a file with the name sum_of_even_numbers.txt and store the sum of even numbers in that file.e.g. 82
- In the same program read the words from input_words_P1.txt (E.g.: Apple, Ball, Cat, Dog, Egg) separated by comma and generate the below output using the digits read in Output:
- 2 Apple, 10 Ball, 8 Cat, 20 Dog, 30 Egg, 12 Apple Save the final output in a file final_output.txt.

14. Create a Menu Driven Program using Python that performs the following using the Database Structure mentioned below: Table Name: Student Fields: student_id, student_name, student_course, student_email, student_dob

- a) Add a new Student and Display the Details
- b) Edit a Particular Student Detail and display the details.
- c) Show All Student Details
- d) Show One Student Detail e) Delete a Student Record.

15. a. Write a Python program that matches a string that has an *a* followed by zero or more *b*'s.

b. Write a Python program to find the sequences of one upper case letter followed by lower case letters.

c. Write a Python program to find the substrings within a string.

d. Write a Python program to convert a date of yyyy-mm-dd format to dd-mm-yyyy format.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

Course Code: CS5223	Course Title: Computer Networks
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2Hrs

Course Objectives:

1. To build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Introduce the student to network security concepts
4. Preparing the student for entry in advanced courses in computer networking.

Course Outcomes:

Here the students will be able to

1. Understand the concepts of Data Communication.
2. Study the functions of OSI Layers.
3. Familiarize with the Transmission Media, Flow Control and Error Detection and Correction.
4. Understand fundamental concepts in Routing, Addressing & working of Transport Protocols.
5. Gain familiarity with common networking & Application Protocols
6. Understand the formats of Wired and Wireless LANs.

CONTENT	Hours
UNIT-1	
<p>INTRODUCTION Communication Network and services, Approaches to Network Design, Network Functions, Key factors in Communication Network, Network Topology, Applications of networks, Line configuration, Layered Architecture and Applications – Examples of Layering, OSI Reference Model, TCP/IP , Telnet, FTP and IP Utilities</p> <p>Self-study: Digital representation of data</p>	7

UNIT-2	
<p>TRANSMISSION SYSTEMS: Digital transmission Systems – Twisted Pair , Coaxial Cable, Optical Fiber, Multiplexing – frequency – Division , Time – Division , Wavelength Division Multiplexing, Switching- Circuit , Packet and Message. Error Correction and detection, Two – dimensional parity checks, Internet Checksum, standardized Polynomial codes, Cyclic redundancy check, Error detecting capability of a polynomial code, Hamming distance and coding.</p> <p>Self-study: Unguided media - Radio wave, Microwave, Infrared Transmission systems.</p>	8
UNIT-3	
<p>PEER –TO-PEER PROTOCOLS: Peer-to peer Protocols and service models, Data Link Control, ARQ Protocols stop and wait, Go–back-N, Selective Repeat, Transmission efficiency of ARQ Protocols, Other adaptation functions, - Sliding window flow control Timing Recovery in Synchronous Services, Reliable Stream Service.</p>	10
UNIT-4	
<p>LOCAL AREA NETWORKS AND MEDIUM ACCESS CONTROL PROTOCOLS: Multiple access communications: Local Area network – LAN Structure, MAC Sub layer, Logical link control layer, Random Access protocols ALOHA, Slotted ALOHA, CSMA, CSMA/CD, Scheduling approaches to medium access control – Reservation Systems, polling , Token passing rings, Channelization – FDMA, TDMA, CDMA. LAN Standard – Ethernet and IEEE 802.3 LAN Standard, IEEE 802. 5 LAN standard, FDDI, Wireless LAN’s : IEEE 802.11 Standards</p> <p>Self-study: IEEE 802.4 standard</p>	7
UNIT-5	
<p>ROUTING AND SECURITY: Routing algorithms classification, Routing tables, shortest path routing algorithms, Flooding , Hierarchical routing , Distance vector routing Link state routing, congestion control algorithms. Network security, ITU-T security Architecture, Three aspects of security, Security attacks, Security services, Security Mechanisms.</p> <p>Self-study: Cryptography, Symmetric and asymmetric encryption.</p>	10

Text Books:

1. Stallings, "Data and Computer Communications", 7th Edition, Pearson Education, 2012 Reference

References:

1. Andrew S Tanenbaum, "Computer Networks", 5th Edition, Pearson Education.
2. Behrouz Forouzan, Introduction to Data Communication & Networking TMH, 2019.
3. Larry & Peterson & Bruce S Davis; Computer networks Second Edition, Morgan Kaufman, 2000.
4. Michael E Whiteman and Herbert J Mattord; "Principles of Information Security", Vikas Publishing House, New Delhi, 2003.
5. Alfred J. Menezes, Paul. C. Van Oorschot, and Scott A. Vanstone "Handbook of Applied Cryptography", CRC press, Lib of Congress -2018.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	7	15
Unit 2	8	18
Unit 3	10	15
Unit 4	7	15
Unit 5	10	15
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CS5P2	Course Title: Computer Networks Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2Hrs

Objectives:

The student should be made

1. Exposed to learn the concepts using the network simulator
2. Learn to implement the algorithms on Error correction and detection.
3. Learn to use network security Cryptool.

Course Outcomes:

Here the students will be able to

1. Understand the concepts of Computer networks.
2. Study the basic network commands and network configuration commands.
3. Familiarize with the Transmission Media, Flow Control and Error Detection & Correction.
4. Understand fundamental concepts in Routing, Addressing & working of Transport Protocols.
5. Gain familiarity with network security.

LIST OF EXPERIMENTS

1. Study of network devices
Repeater, Hub, Switch, Router, Digitizer, Bridge, Gateway.
2. Study of IP address, Sub netting and Super netting
3. Connect the computers in Local area networks.
4. Study of basic network commands and network configuration
4. Write a socket program for the implementation of echo.
5. Write a program for error detecting code using CRC-CCITT (16- bits).
6. Write a program for error detecting code using 2D Parity check.
7. Write a program to find the shortest path between vertices using Dijkstra's Algorithm.
8. Using TCP/IP sockets, write a client – server program to make the client send the file name and to make the server send back the contents of the requested file if present.
9. Write a program for congestion control using leaky bucket algorithm.
10. Write a program for congestion control using Token bucket algorithm.
11. Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.
12. Write a program to implement Token Passing algorithm.
13. Write a program to encrypt and decrypt a Password.

14. Implement the substitution mono alphabetic technique by using Caesar Cipher algorithm.
15. Write a program to implement Play fair Cipher algorithm.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

SEMESTER VI

Course Code: CS6123	Course Title: Internet Technologies and C#
Course Credits: 03	Hours/Week: 03
Total Contact Hours:42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration: 2Hrs

Course Objectives

1. Determine logical alternatives with C# decision structures utilizing iteration, class methods, fields, delegate and properties
2. To provide the familiarity in the concept of developing console, window and web application.
3. To gain knowledge about the methodologies behind C# and ASP.Net and helps the students to develop Dot Net based application using ADO.NET and SQL

Course Outcomes (COs):

On successful completion of this course, students should be able to:

1. Understand the concept of .Net Framework and C# language fundamentals.
2. Evaluate advanced C# OOPs concept and the .NET framework contents.
3. Set up various navigation techniques for integrating web pages using ASP.NET
4. Create your own website, enhanced by using Master pages and Themes.
5. Create database-oriented application, Perform DML Operation / small projects with database interaction.

CONTENT	Hours
UNIT 1	
Introduction: .NET Definition, Characteristics, Application, Difference between C and C#, .NET framework, benefits of .NET approach, benefits of C# and .NET, .NET Architecture-Common Language Runtime, MSIL, .NET Framework Classes. Advantages of Managed Code ,Garbage Collection ,Security, program structure , command line argument, Literals, variables and data types , constant variables, scope of variables, boxing and unboxing, Enumerations.	8
UNIT 2	

<p>Features of C# and OOP: Operators in C#, expression, Decision making and looping statements in c# Methods in C# -declaring methods, nesting of methods, methods parameters, the output parameters, Static data members, Static member function, Arrays - variable size arrays, array list class, String handling . Delegate- delegate declaration, delegate methods, delegates instantiation, delegate invocation, using delegates. Overview of OOP , Defining a class, Adding Variables and Methods, Member Access Modifiers, Creating Objects, Accessing Class Members,</p>	9
---	---

UNIT 3

<p>Object Oriented C#: Types of Constructors, This Reference, Inheritance and polymorphism: classical inheritance, containment inheritance, defining a subclass, visibility control, defining subclass constructors, multilevel inheritance, hierarchical inheritance, overriding methods, hiding methods, abstract classes, abstract methods, sealed classes: Preventing inheritance, sealed methods, polymorphism: Operator overloading. Interfaces: Multiple Inheritance: defining an interface, extending an interface, implementing interface, interface & inheritance. Errors and Exception Handling, Exceptions Classes User Defined Exceptions</p>	9
---	---

UNIT 4

<p>Introduction to ASP.NET: Code behind Web Form-Separations of content & Business logic-Life Cycle of a Web Form Page-Stages NET- ASP.NET. Creating A web application using ASP.NET ,Components of an ASP.NET , Deploying ASP.NET application, Custom Control: Introduction to Server Controls, HTML Controls, Validation Controls, User control, Data Binding Controls, Configuration, Personalization, Session State, Adding controls to a web form, Buttons, Text Box , Labels, Checkbox, Radio Buttons, List Box, etc. Self-study : Difference Between ASP and ASP.NET</p>	8
--	---

UNIT 5

<p>Introduction to Data Access: Comparison between ADO & ADO.NET—The difference between Connection Model & Disconnected Model – difference between the DataSet and RecordSet- The Dataset Model. Accessing Data using ADO.NET: datasetDataAdapter DataRelation. Self-study: APPLICATION DEVELOPMENT ON .NET: Creating our own window forms with events and controls, menu creation, accessing data with ADO.NET.</p>	8
---	---

TEXT BOOKS:

1. Simon Robinson, Christian Nagel, Karli Watson, Jay Glynn, Morgan Skinner and Bill Evjen, Professional C#, Wiley – dreamtech India Pvt. Ltd., 3rd Edition.
2. C# 8.0 and .NET, Core 3.0 is a book written by Mark J 4th Edition.
3. NET(Core Reference) Microsoft® Visual C#® 2005: The Language by Donis Marshall
4. Programming in C#, E. Balagurusamy, 2017, McGraw Hill Education 5. C# 4.0 complete reference by Herbert Schildt, from McGraw Hill Education
5. Beginning ASP.NET in C#, Matthew MacDonald.
6. HANDS ON WITH ASP.NET MVC - COVERING MVC 6, Rahul Sahay. Designing Microsoft ASP.NET Application , Jonathan Goodyear, Brain Peek.

Reference books:

1. Crouch, ASP.NET and VB.NET Web Programming, Pearson Education
2. Richard Blair, Mathew Renolds, Beginning VB.NET 2003, 3rd edition, Wrox Publication
3. Bill Evjen, Billy, Hollis, et al, Professional VB.NET 2003, 3rd edition, Wrox Publication
4. Deitel and Deitel, Visual Basic.NET How to Program, Pearson Education, 2nd edition Greg Buczek, ASP.NET Developer's Guide, Tata McGraw-Hill, 2002.
5. Chris Ullman, John Kauffman, Beginning ASP.NET 1.1 with VB.NET 2003, Wrox Publication
6. ADO.NET Professional, Wrox Publication
7. Alex Homer, Dave Sussman, Professional ASP.NET 1.1, Wrox Publication
8. .NET Framework, OREILY Publication.

BLUEPRINT

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	15
Unit 2	9	15
Unit 3	9	15
Unit 4	8	17
Unit 5	8	16
TOTAL	42	78
Maximum marks for the paper (Excluding bonus questions) =60		

Course Code: CS 6P1	Course Title: Internet Technologies and C# Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours:52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2Hrs

Course Objectives

1. This course will enable students to design and develop Windows-based business applications using .NET technologies that meet commercial programming standards.
2. Students will be able to implement advanced c# concept.
3. Knowledge of .NET environments.

Course Outcomes (COs):

This course will enable students to

1. Design and develop Windows-based business applications using .NET technologies that meet commercial programming standards.
2. Write programs using object-oriented programming techniques including classes, objects, inheritance, and polymorphism
3. Write code to access and manipulate databases

Part A

1. Write a program to demonstrate the working of command line arguments
 - a) Sum n natural number using constant.
 - b) Sum of series using read only.
2. Write a program to demonstrate the working of boxing and unboxing.
 - a) kilometer to meter
 - b) data type to object type.
3. Write a program to demonstrate the working of scope of the variables for the following access modifier.
 - a) Public
 - b) Protected
 - c) Private
 - d) Internal
 - e) Protected internal.
4. Write a program to demonstrate the working of different control statement used in c# with support to a generate shopping bill will contain bill number, name of clerk preparing the bill, each item with quantity and price and total amount to be paid. Total items in the bill are varying can be accommodated in a single bill. There is an array describing each item with a price. The price is to be picked up from that.
5. Write a program to demonstrate the working of looping statement used in c# to select an option from the menu. Upon selecting a menu item, we will be taken to concerned options. Here, if we don't select any options, it just hangs out. So the condition(menu) must be displayed, before its usage.

6. Given below is a hypothetical table showing rates of Income Tax for male citizens below the age of 65 years:

Taxable Income (T1) in

Does not exceed 1,60,000

Is greater than 1,60,000 and less than or equal to 5,00,000

Is greater than 5,00,000 and less than or equal to 8,00,000

Is greater than 8,00,000

Income Tax in

Nil

$(T1 - 1,60,000) * 10\%$

$[(T1 - 5,00,000) * 20\%] + 34,000$ $[(T1 - 8,00,000) * 30\%] + 94,000$

Write a program to input the age, gender (male or female) and Taxable Income of a person. If the age is more than 65 years or the gender is female, display –wrong category|. If the age is less than or equal to 65 years and the gender is male, compute and display the Income Tax payable as per the table given above.

7. Write a program to demonstrate the working of

a) To input the length and breadth of a rectangle and find its diagonal. using method

b) Write a program to pass 2 integer numbers as parameters. If either of the two numbers is 0, display invalid entry and the program should end, if it is valid entry, divide the larger number with the smaller number and display the result.

8. Write a program to demonstrate the working of

a) Swapping of two values using ref keyword.

b) Method overloading for displaying int and character data.

9. Write a program to demonstrate the working of nesting of method to reverse a number using nested method for new modifier.

10. Write a program to read the data using one- dimensional array and determine the following.

a) Total marks obtained.

b) The highest marks and lowest marks obtained subject wise.

11. Write a program to demonstrate the working multi-dimensional array for matrix multiplication.

12. Write a program to demonstrate the working of jagged array to input and display elements in the calendar format.

13. a) Write a program in C# Sharp to read a string through the keyboard and sort it using bubble sort.

b) Create a project to demonstrate the working of immutable string operation and find the time taken for the immutable string manipulation.

14. Create a project to demonstrate the working of ATM transaction.

15. Create a project to implement phone book.

16. Create a project to demonstrate the working of shopping mall.

Evaluation Scheme for Lab Examination

Assessment Criteria	Marks
Writing	8
Execution	10
Viva Voice	7
Total	25

Course Code:CS6223	Course Title: Operating System
Course Credits: 03	Hours/Week: 03
Total Contact Hours: 42	Formative Assessment Marks: 40
Exam Marks: 60	Exam Duration:2 Hrs.

Course Objectives:

1. To give an overview of operating system, services provided, different types of OS and components of OS.
2. To understand the concept of process and how processes are synchronized and scheduled, with an understanding of deadlock handling.
3. To have a comprehension of the various methods of memory management techniques.
4. To gain an insight into secondary memory management aspects of file and disk structure and scheduling.

Course Outcomes:

On successful completion of the course, students will be able to

1. Understand various services that are offered by the operating system at different levels.
2. Have an understanding of the practical uses of operating systems in a wide range of fields.
3. Acquire the basic understanding required to comprehend the architecture and components of operating systems.
4. Illustrate the use of different process scheduling algorithm and synchronization techniques to detect and avoid deadlocks.
5. Gain knowledge of a variety of memory management strategies, virtual memory aspects, as well as various device and file management algorithms for secondary storage management.

CONTENT	HOURS
UNIT 1:	
INTRODUCTION: Definition, functions, operating system structure, types- simple batch, Buffering & spooling, multiprogramming, time - sharing, Real time system, distributed systems, parallel systems protection, system components, system calls.	08
UNIT 2:	
PROCESS MANAGEMENT: Process concept [program, process and threads], states of a process, process control block, operations on a process, Independent and Cooperating process, Inter-process communication, Scheduling, scheduling Criteria, scheduling queues ,types of schedulers – short term , long term, medium term schedulers , pre-emptive and non-preemptive scheduling , scheduling algorithms- FCFS, SJF, Round Robin, Priority scheduling, Multilevel scheduling , multilevel feedback algorithms	12
UNIT 3:	
PROCESS SYNCHRONIZATION AND DEADLOCK MANAGEMENT: Process synchronization – semaphores, critical section problems, classic problem of synchronization, Introduction to Deadlock, deadlock characterization, methods for handling deadlock, deadlock prevention, avoidance and detection using resource allocation graph and wait for graph, Recovery from deadlock.	08
UNIT 4:	
MEMORY MANAGEMENT: Memory Management: Logical and Physical address space, Swapping, Contiguous allocation, Paging, Segmentation, Virtual memory, Demand paging, Page replacement Algorithms-First in first out, Least Recently Used algorithm, thrashing.	08
UNIT 5:	
SECONDARY STORAGE MANAGEMENT: File system – definition, file attributes, file types, file structure, access methods, directory structure, file protections. Disk structure, allocation methods, need for disk scheduling, disk scheduling algorithms - SCAN, C-SCAN (circular scan). Self-study : FCFS (First Come, First Serve), SSTF (Shortest Seek Time First),	06

Text Books:

1. Operating System concepts by Abraham Silberschatz, Peter Galvin, Greg Gagne, Seventh Edition.

References:

1. Operating Systems: Internals and Design Principles, by William Stallings, seventh edition
2. Operating System Concepts by James L Peterson, second Edition.
3. Operating System Design and Implementation by Andrew S Tenenbaum. Third edition.

BLUEPRINT:

Chapter	No. of hours	Total marks for which questions are to be asked (Including bonus questions)
Unit 1	8	12
Unit 2	12	16
Unit 3	8	18
Unit 4	8	18
Unit 5	6	18
TOTAL	42	82
Maximum marks for the paper (Excluding bonus questions) = 60		

Course Code: CS6P2	Course Title: Major Project Lab
Course Credits: 02	Hours/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 25
Exam Marks: 25	Exam Duration: 2 Hrs

Course Objective:

1. Identify real world problems and challenges that need IT based solutions.
2. Demonstrate an ability to work in teams.
3. Improve the team building, communication and management skills of the students.
4. Use all concepts of IT in creating a solution for a problem.
5. Report and present the findings of the study conducted in the preferred domain.

Course Outcomes:

Outcomes of the Major Project:

Project Analysis and Design will cover following

1. Problem definition
2. Analyze Systems requirements.
3. Plan the systems design phase of the SDLC.
4. Decide the s/w requirement specifications and h/w requirement specifications.
5. Perform and evaluate feasibility studies like cost-benefit analysis, technical feasibility, time feasibility and operational feasibility for the project.
6. Work on data collection methods for fact finding.
7. Construct and evaluate data dictionaries/ decision trees/ decision table.
8. Create and evaluate graphical tools as systems flow charts, entity-relationship (er) diagrams and state transition diagrams.
9. Construct and evaluate data flow diagrams.

Project Development and Implementation will cover following

1. Perform coding for the project.
2. Design and evaluate system outputs.
3. Design and evaluate systems inputs.
4. Design and evaluate validity checks for input data.
5. Design and evaluate user interfaces.
6. Documentation of requirements
7. Prepare documentation of project
8. Perform various testing techniques/strategies.
9. Generate various reports in project.
10. Deploy the project on machine/lab/real time environment
11. Define the future scope and further enhancement of the system.

Project Guidelines

1) Understanding the significance of Project

Most of the students are under an impression that if good layout is proposed then it will help them in scoring good marks but the quality of project is analyzed by proper write-ups. It provides an opportunity for learners to demonstrate originality and to plan and organize the project work and put the practical approach of all the topics studied in the entire curriculum.

2) Meaning of Project

A project is a study of factual information for comprehending and applying the various concepts of the course into practice. Its main purpose is not to generalize but to study the situation with a practical orientation.

3) No Objection Certificate

If the project is carried out in a company or organization, then a certificate for no objection of same needs to be presented. It should mention that the organization has no objection in publishing the findings of the project study. The certificate should contain the name of authority with signature and company stamp and should be given on company's letterhead and duly signed by authorized signatory.

GUIDELINES FOR BSc PROJECT

- The project work should normally include software development.
- The students are expected to work on real-life project. However, it is not mandatory for a student to work on a real-life project.
- Not more than four students are permitted to work on a project.
- The project may be done in the university campus/concern study institute or in an approved sponsoring organization (industry/ research and development laboratories / educational institution / software company related to the proposed topic.
- The student can formulate a project problem with the help of her/his lab supervisors / teachers and if approved, the students commence working on it.
- A candidate is required to present the progress of the project work during the semester as per the schedule.

4) PROJECT SYNOPSIS FORMAT

The project proposal should be prepared and approved in consultation with supervisor. The project proposal should clearly state the project objectives and the environment of the proposed project to be undertaken. The project proposal should contain complete details in the following form:

1. Title of the project
2. Introduction and objectives of the project
3. Analysis (DFD, ER diagrams, class diagrams, time line etc. As per the project requirements).
4. A complete structure which includes:
 - Name of modules and their description
 - Database / data structures description
 - Process logic of each module (flow chart)
 - Reports generation. (Report format)
5. Tools / platform, hardware and software requirement specifications
6. Organization/ Company Certificate

5) Technical Specifications of Project Report

Length:

The length of the report should be between 80- 100 pages including the cover page, summary, table of contents, list of figures, list of tables, and acknowledgement.

Script and Page Format:

The report should be typed using a Word Processor on standard A4 (210 mm x 297 mm) paper size. A conventional font, size 12-point for Normal text, 14 for headings, 16 for Chapter heading, and line spacing of 1.5 mm should be used.

Margins:

Left-hand margins should have a width of not less than 38 mm to facilitate binding. The right hand, the top, and the bottom should be 25 mm. Each page must be typed in one side, leaving a wide margin.

Paper and Print Quality Paper and print quality:

Paper and Print Quality Paper and print quality are important for successful legibility. The report can be printed on a standard quality paper, (e.g., photocopy paper)

Pagination:

Positioning of page numbers should be at the bottom of the pages. Pages starting from the summary until the last list of tables should be numbered using Latin numbers (I, II, III, IV, ...).

Pages starting from the Introduction until the appendices should be numbered using numbers (1,2, 3,...). Pages with figures and tables or illustrations must be also numbered.

Binding:

The report should be hard bound.

Number of copies:

Two copies should be submitted – self copy, College copy(Group).

Project Evaluation

Presentation / demo must be carried out in all the lab sessions in the whole semester for internal assessment of the project. In each lab session a student is evaluated for 25 marks.

Presentation /Demo-15

Viva Voce-10

End Semester assessment must be carried out based on documentation, demo and Viva

Assessment Criteria	Marks
Presentation/Demo	10
Documentation	5
Viva Voce	10
Total	25

Lab Manual for 10 weeks

Weeks Descriptions

Week 1: Detail briefing of project, Group formation

Week 2: No objection certificate, Synopsis and timeline chart /Gantt Submission

Week 3: Requirement Analysis

Week 4: System Design

Week 5: UI design for minimum 4 pages.

Week 6: Database Specification

Week 7: UI design and Coding

Week 8: Implementation, Testing and maintenance

Week 9: Demo of the Project

Week 10: Documentation for Supervisor Signature