

BENGALURU NORTH UNIVERSITY
KOLAR - 563103

National Education Policy - 2020
(Semester Scheme)

Model Curriculum Structure for Bachelor of Science
(Basic and Honours) Programmes with Computer Science
as Major and Minor Courses.

Model Syllabus for I and II Semester
and
Open Elective Courses in Computer Science.

Revised w.e.f.

Academic Year 2021 - 2022 and onwards

Model Curriculum Structure

Program: BSc (Basic and Honors)

Subject: Computer Science

1. Computer Science as MAJOR with another Subject as MINOR (Table IIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)/ Vocational Courses (VC)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	4 4	VC-1: Any one from Vocational Courses, Group – 1*	3
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	4 4	VC-2: Any one from Vocational Courses, Group – 2* Internship:	3 2
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

2. Computer Science as MAJOR with another Subject also as MAJOR (Table IIIA of Model Curriculum)

Sem	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week		Discipline Specific Elective Courses (DSE)	Hour of Teaching/ Week
		Theory	Lab		
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4		
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4		
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4		
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4		
5	DSC-5: Programming in PYTHON DSC-6: Computer Networks DSC-5Lab: PYTHON Programming lab DSC-6Lab: Computer Networks Lab	3 3	4 4		
6	DSC-7: Internet Technologies DSC-8: Operating System Concepts DSC-7Lab: JAVA Script, HTML, CSS Lab DSC-8Lab: C# Programming Lab	3 3	4 4		
7	DSC-9: Computer Graphics and Visualization DSC-10: Design and Analysis of Algorithms DSC-11: Software Engineering DSC-9Lab: Computer Graphics and Visualization Lab DSC-10Lab: Algorithms Lab	3 3 3	4 4	DSE-1: Any one from Discipline Specific Elective Courses, Group – 1** DSE-2: Any one from Discipline Specific Elective Courses, Group – 2** Research Methodology:	3 3 3
8	DSC-12: Artificial Intelligence and Applications DSC-13: Computer Organization and Architecture DSC-14: Data Warehousing and Data Mining DSC-12Lab: AI Lab	3 3 3	4	DSE-3: Any one from Discipline Specific Elective Courses, Group – 3** Research Project:	3 6

3. Computer Science as MINOR with another Subject as MAJOR (As per Table IIA of Model Curriculum)

Semester	Discipline Specific Core Courses (DSC)	Hour of Teaching/ Week	
		Theory	Lab
1	DSC-1: Computer Fundamentals and Programming in C DSC-1Lab: C Programming Lab	4	4
2	DSC-2: Data Structures using C DSC-2Lab: Data structures Lab	4	4
3	DSC-3: Object Oriented Programming Concepts and Programming in JAVA DSC-3Lab: JAVA Lab	4	4
4	DSC-4: Database Management Systems DSC-4Lab: DBMS Lab	4	4
5	DSC-5: Programming in PYTHON DSC-5Lab: PYTHON Programming lab	3	4
6	DSC-6: Internet Technologies DSC-6Lab: JAVA Script, HTML, CSS Lab	3	4

* Vocational Courses:

Group-1: <ul style="list-style-type: none"> • DTP, CAD and Multimedia • Hardware and Server Maintenance • Web Content Management Systems • E-Commerce • Web Designing 	Group-1: <ul style="list-style-type: none"> • Health Care Technologies • Digital Marketing • Office Automation • Multimedia Processing • Accounting Package
---	---

** Discipline Specific Elective Courses:

Group-1: <ul style="list-style-type: none"> • IoT • Cyber Law and Cyber Security • Web Programming - PHP and MySQL • Clouds, Grids, and Clusters • Software Testing 	Group-2: <ul style="list-style-type: none"> • Information and Network Security • Data Compression • Discrete Structures • Opensource Programming • Multimedia Computing • Big Data 	Group-3: <ul style="list-style-type: none"> • Data Analytics • Storage Area Networks • Pattern Recognition • Digital Image Processing • Parallel Programming • Digital Signal Processing
---	---	---

Model Syllabus for BSc (Basic and Honors), Semesters I and II

Semester: I

Course Code: DSC-1	Course Title: Computer Fundamentals and Programming in C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 30
Exam Marks: 70	Exam Duration: 03

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; 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 C Program with Examples; Creating and Executing a C Program; Compilation process in C. C Programming Basic Concepts: C Character Set; C tokens - keywords, identifiers, constants, and variables; 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 strings and escape sequences, output specifications with <i>printf</i> functions; Unformatted I/O functions to read and display 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 Exit controlled statements, <i>while, do-while, for</i> loops, Nested loops.	
Unit - 4	
<p>Arrays: One Dimensional arrays - Declaration, Initialization and Memory representation; Two Dimensional arrays - Declaration, Initialization and Memory representation.</p> <p>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.</p> <p>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;</p>	12
Unit - 5	
<p>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.</p> <p>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.</p>	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)

Course Code: DSC-1Lab	Course Title: C Programming Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04

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

Evaluation Scheme for Lab Examination

Assessment Criteria		Marks
Program – 1 from Part B	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		40

Semester: II

Course Code: DSC-2	Course Title: Data Structures using C
Course Credits: 04	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 30
Exam Marks: 70	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. Algorithm Specification, Performance Analysis, Performance Measurement Recursion: Definition; Types of recursions; Recursion Technique Examples - Fibonacci numbers, GCD, Binomial coefficient nC_r , Towers of Hanoi; Comparison between iterative and recursive functions.	10
Unit - 2	
Arrays: Basic Concepts – Definition, Declaration, Initialisation, 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, Selection sort, Insertion sort; Searching - Sequential Search, Binary search; Iterative and Recursive searching; Multidimensional arrays; Representation of multidimensional arrays; Sparse matrices.	10
Unit - 3	
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> . 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 – 4	
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. Queues: Basic Concepts – Definition and Representation of queues; Types of queues - Simple queues, Circular queues, Double ended queues, Priority queues; Operations on Simple queues;	10
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 representation of binary tree. Traversal of binary tree; <i>preorder</i> , <i>inorder</i> and <i>postorder</i> traversal; Reconstruction of a binary tree when any two of the traversals are given.	10

Text Books

1. Satraj Sahani: 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))

Course Code: DSC-2Lab	Course Title: Data Structures Lab
Course Credits: 02	Hour of Teaching/Week: 04
Total Contact Hours: 52	Formative Assessment Marks: 10
Exam Marks: 40	Exam Duration: 04

Programming Lab

Part 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

Assessment Criteria		Marks
Program – 1 from Part A	Flowchart / Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Program -2 from Part B	Flowchart/Algorithm	02
	Writing the Program	05
	Execution and Formatting	08
Viva Voice based on C Programming		05
Practical Record		05
Total		40

Open Electives in Computer Science:

(For BA, BSc, BCom, BSW, BBA, BBM students studying Core Courses other than ComputerScience/ Computer Applications)

- Office Automation
- C Programming Concepts
- Multimedia Processing
- Python Programming Concepts
- R Programming
- E-Content Development
- E-Commerce
- Web Designing
- Computer Animation
- Accounting Package

Syllabus for BSc (Basic and Honors), Semesters III and IV

S.No	Cours e	Semester	Subject	Recommendation
1	B.Sc	III	DSC-3: Object Oriented Programming Concepts and Programming in JAVA	It was proposed to approve the scheme and syllabus for UG Computer Science & BCA (NEP) 3 rd and 4 th Semester from the academic year 2022-2023 onwards. The Board read through the Syllabus. The Board considered and approved the scheme and syllabus for UG Computer Science & BCA (NEP). After discussion, reviewed and fruitful suggestions were incorporated.
		III	DSC-3 Lab: JAVA Lab	
		IV	DSC-4: Database Management Systems	
		IV	DSC-4 Lab: DBMS Lab	
2	B.Sc. & other Subject Students	III / IV	Skill Enhancement Course: Artificial Intelligence	

Sem	Discipline Specific Core Courses (DSC)	Hour / Week		DS Elective Courses	Hours/ Week
		Theory	Lab		
III	DSC-3: Object Oriented Programming Concepts and Programming in JAVA	4			
	DSC-3 Lab: JAVA Lab		4		
IV	DSC-4: Database Management Systems	4			
	DSC-4 Lab: DBMS Lab		4		

Syllabus for BSc (Basic and Honors), Semesters III and IV

Semester: III

Course Title: Object Oriented Programming Concepts and Programming in Java	Course code: DSC3
Total Contact Hours: 52	Course Credits: 04+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the object-oriented concepts and JAVA.
- Write JAVA programs using OOP concepts like Abstraction, Encapsulation, Inheritance and Polymorphism.
- Implement Classes and multithreading using JAVA.
- Demonstrate the basic principles of creating Java applications with GUI.

DSC3: Object Oriented Programming Concepts and Programming in Java

Unit	Description	Hours
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.	09
2	Objects and Classes: Basics of objects and classes in java, Constructors, Finalizer, Visibility modifiers, Methods and objects, Inbuilt classes like String, Character, String Buffer, File, this reference.	10
3	Inheritance and Polymorphism: Inheritance in java, Super and sub class, Overriding, Object class, Polymorphism, Dynamic binding, Generic programming, Casting objects, Instance of operator, Abstract class, Interface in java, Package in java, UTIL package.	09
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, Exceptional handling mechanism.	12
5	I/O programming: Text and Binary I/O, Binary I/O classes, Object I/O, Random Access Files. Multithreading in java: Thread life cycle and methods, Runnable interface, Thread synchronization, Exception handling with try catch-finally, Collections in java, Introduction to JavaBeans and Network Programming	12

References:

1. Programming with Java, By E Balagurusamy – A Primer, 4th Edition, McGraw Hill Publication.
2. Core Java Volume I – Fundamentals, By Cay S. Horstmann, Prentice Hall.
3. Object Oriented Programming with Java: Somashekara M.T., Guru, D.S., Manjunatha K.S, 1st Edition, PHI Learning 2017.
4. Java 2 - The Complete Reference, Herbert Schildt, 5th Edition, McGraw Hill Publication, 2017.
5. Java - The Complete Reference, Herbert Schildt, 7th Edition, McGraw Hill Publication, 2017.

Semester: III

Course Title: JAVA Lab	Course code: DSC3 Lab
Total Contact Hours: 52	Course Credits: 02

Course Outcomes (COs):**At the end of the course, students will be able to:**

- Implement Object Oriented programming concept using basic syntaxes of control Structures
- Identify classes, objects, members of a class and the relationships among them needed for finding the solution to specific problem
- Demonstrates how 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

1. Program to print the following triangle of numbers

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

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.
6. 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 AddSub with methods to add and subtract. Create another class called MulDiv that extends from AddSub class to use the member data of the super class. MulDiv 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
 - a. To find the area and circumference of the circle by accepting the radius from the user.
 - b. 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. Total of 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 subjects 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.
8. In a college first year class are having the following attributes Name of the class (BCA, BCom, BSc), Name of the staff No of the students in the class, Array of students in the class
9. 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
10. 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.

11. Create a package „student. Fulltime. BSC „in your current working directory

- a. Create a default class student in the above package with the following attributes:Name, age, sex.
- b. Have methods for storing as well as displaying

PART B: 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 amessage to the user.
3. Program which create and displays a message on the window
4. Program to draw several shapes in the created window 32
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 clickmother 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. Program to create a window when we press M or m the window displays Good Morning,A or a the window displays Good After Noon E or e the window displays Good Evening,N or n the window displays Good Night
11. Demonstrate the various mouse handling events using suitable example.
12. 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

Evaluation Scheme for Lab Examination:

Assessment Criteria		Marks
Program – 1 from Part A	Writing the Program	03
	Execution and Formatting	07
Program -2 from Part B	Writing the Program	03
	Execution and Formatting	07
Viva Voice		05
Total		25

Semester: IV

Course Title: Database Management System	Course code: DSC4
Total Contact Hours: 52	Course Credits: 04+02
Formative Assessment Marks: 40	Duration of SEE/Exam: 03 Hours
Summative Assessment Marks: 60	

Course Outcomes (COs):

At the end of the course, students will be able to:

- Explain the various database concepts and the need for database systems.
- Identify and define database objects, enforce integrity constraints on a database using DBMS.
- Demonstrate a Data model and Schemas in RDBMS.
- Identify entities and relationships and draw ER diagram for a given real-world problem.
- Convert an ER diagram to a database schema and deduce it to the desired normal form.
- Formulate queries in Relational Algebra, Structured Query Language (SQL) for database manipulation.
- Explain the transaction processing and concurrency control techniques.

DSC7: Database Management System (DBMS)

Unit	Description	Hours
1	Database Architecture: Introduction to Database system applications. Characteristics and Purpose of database approach. People associated with Database system. Data models. Database schema. Database architecture. Data independence. Database languages, interfaces, and classification of DBMS.	10
2	E-R Model: Entity-Relationship modeling: E – R Model Concepts: Entity, Entity types, Entity sets, Attributes, Types of attributes, key attribute, and domain of an attribute. Relationships between the entities. Relationship types, roles and structural constraints, degree and cardinality ratio of a relationship. Weak entity types, E -R diagram.	10
3	Relational Data Model: Relational model concepts. Characteristics of relations. Relational model constraints: Domain constraints, key constraints, primary & foreign key constraints, integrity constraints and null values. Relational Algebra: Basic Relational Algebra operations. Set theoretical	12

	operations on relations. JOIN operations Aggregate Functions and Grouping. Nested Sub Queries-Views. Introduction to PL/SQL & programming of above operations in PL/SQL	
4	Data Normalization: Anomalies in relational database design. Decomposition. Functional dependencies. Normalization. First normal form, Second normal form, Third normal form. Boyce-Codd normal form.	09
5	Query Processing Transaction Management: Introduction Transaction Processing. Single user & multiuser systems. Transactions: read & write operations. Need of concurrency control: The lost update problem, Dirty read problem. Types of failures. Transaction states. Desirable properties (ACID properties) of Transactions. Concurrency Control Techniques: Locks and Time stamp Ordering. Deadlock & Starvation.	11

References:

1. Fundamentals of Database Systems, Ramez Elamassri, Shankant B. Navathe, 7th Edition, Pearson, 2015
2. An Introduction to Database Systems, Bipin Desai, Galgotia Publications, 2010.
3. Introduction to Database System, C J Date, Pearson, 1999.
4. Database Systems Concepts, Abraham Silberschatz, Henry Korth, S.Sudarshan, 6th Edition, McGraw Hill, 2010.
5. Database Management Systems, Raghu Rama Krishnan and Johannes Gehrke, 3rd Edition, McGraw Hill, 2002

Practicals:

Student would be able to create tables, execute queries and PL/SQL programs.

1. Execute a single line query and group functions.
2. Execute DDL Commands.
3. Execute DML Commands
4. Execute DCL and TCL Commands.
5. Implement the Nested Queries.
6. Implement Join operations in SQL
7. Create views for a particular table
8. Implement Locks for a particular table
9. Write PL/SQL procedure for an application using exception handling.
10. Write PL/SQL procedure for an application using cursors.
11. Write a PL/SQL procedure for an application using functions
12. Write a PL/SQL procedure for an application using package

Skill Enhancement Course: SEC for B.Sc. & Other Subject Students**SEC Model-2****Semester: III/IV**

Course Title: Artificial Intelligence	Course Credits: 2
Total Contact Hours: 13 hours of theory and 26 hours of practical	Duration of ESA: 01 Hour
Formative Assessment Marks: 20 marks	Summative Assessment Marks: 30 marks

Course Outcomes (COs):

At the end of the course, students will be able to:

- Appraise the theory of Artificial intelligence and list the significance of AI.
- Discuss the various components that are involved in solving an AI problem.
- Illustrate the working of AI Algorithms in the given contrast.
- Analyze the various knowledge representation schemes, Reasoning and Learning techniques of AI.
- Apply the AI concepts to build an expert system to solve the real-world problems.

Course Content

Unit - 1	
Overview of AI: Definition of Artificial Intelligence, Philosophy of AI, Goals of AI, Elements of AI system, Programming a computer without and with AI, AI Techniques, History of AI.	05
Intelligent Systems: Definition and understanding of Intelligence, Types of Intelligence, Human Intelligence vs Machine Intelligence.	
Unit – 2	
AI Applications: Virtual assistance, Travel and Navigation, Education and Healthcare, Optical character recognition, E-commerce and mobile payment systems, Image based search and photo editing.	05
AI Examples in daily life: Installation of AI apps and instructions to use AI apps.	
Unit – 3	
Robotics: Introduction to Robotics, Difference in Robot System and Other AI Program, Components of a Robot.	03

Laboratory Activities:

- **Amazon Alexa:**
<https://play.google.com/store/apps/details?id=com.amazon.dee.app&hl=en&am p;gl=US>
- **Google Lens:**
<https://play.google.com/store/search?q=google+lens&c=apps&hl=en&gl=US>
- **Image to Text to Speech ML OCR:**
https://play.google.com/store/apps/details?id=com.mlscanner.image.text.speech&hl=en_IN&gl=US
- **Google Pay:**
https://play.google.com/store/apps/details?id=com.google.android.apps.nbu.paisa.user&hl=en_IN&gl=US

26

Grammarly:

https://play.google.com/store/search?q=grammarly&c=apps&hl=en_IN&gl=

☐ **Google Map:**

<https://play.google.com/store/search?q=google+maps&c=apps&hl=en&gl=US>

☐ **FaceApp:**

https://play.google.com/store/apps/details?id=io.faceapp&hl=en_IN&gl=US

☐ **Socratic:**

☐ https://play.google.com/store/apps/details?id=com.google.socratic&hl=en_IN&gl=US

Google Fit: Activity Tracking:

☐ https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=en_IN&gl=US

☐ **SwiftKey Keyboard:**

<https://swiftkey-keyboard.en.uptodown.com/android>

☐ **E-commerce App:**

https://play.google.com/store/apps/details?id=com.jpl.jiomart&hl=en_IN&gl=US

Text Books:

1. Wolfgang Ertel, "Introduction to Artificial Intelligence", 2nd Edition, Springer International Publishing 2017.
2. Michael Negnevitsky, "Artificial Intelligence A Guide to Intelligent Systems", 2nd Edition, Pearson Education Limited 2005.

References:

1. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_tutorial.pdf
2. Kevin Knight, Elaine Rich, Shivashankar B. Nair, "Artificial Intelligence", 3rd Edition, July 2017.

Reference Links:

1. Voice Assistant: <https://alan.app/blog/voiceassistant-2/>
2. Browse with image: <https://www.pocket-lint.com/apps/news/google/141075-what-is-google-lens-and-how-does-it-work-and-which-devices-have-it>
3. OCR: <https://aws.amazon.com/what-is/ocr/>
4. Mobile Payment system: <https://gocardless.com/en-us/guides/posts/how-do-mobile-payment-systems-work/>
5. Grammarly: <https://techjury.net/blog/how-to-use-grammarly/#gref>
6. Travel & Navigation: <https://blog.google/products/maps/google-maps-101-ai-power-new-features-io-2021/>
7. AI in photo editing: <https://digital-photography-school.com/artificial-intelligence-changed-photo-editing/>
8. AI in education: <https://www.makeuseof.com/what-is-google-socratic-how-does-it-work/>
9. AI in health and fitness: <https://cubettech.com/resources/blog/implementing-machine-learning-and-ai-in-health-and-fitness/>
10. E-commerce and online shopping: <https://medium.com/@nyxonedigital/importance-of-e-commerce-and-online-shopping-and-why-to-sell-online-5a3fd8e6f416>

Open Elective for III & IV Semester

Python Programming Concepts

Course Title: Python Programming Concepts	Course Credits: 3 (3L+0T+0P)
Semester: III	Duration of SEE: 03 Hour
Total Contact Hours: 42	SEE: 60 Marks IA: 40 Marks

Course Outcomes (COs):

- Explain the fundamentals of Computers.
- Explain the basic concepts of Python Programming.
- Demonstrate proficiency in the handling of loops and the creation of functions.
- Identify the methods to create and store strings.

Unit I Fundamentals of Computers

10 Hrs

Introduction to Computers - Computer Definition, Characteristics of Computers, Evolution and History of Computers, Types of Computers, Basic Organization of a Digital Computer; 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.

Unit II Python Basics

10 Hrs

Introduction to Features and Applications of Python; Python Versions; Installation of Python; Python Command Line mode and Python IDEs; Simple Python Program. Identifiers; Keywords; Statements and Expressions; Variables; Operators; Precedence and Association; Data Types; Indentation; Comments; Built-in Functions- Console Input and Console Output, Type Conversions; Python Libraries; Importing Libraries with Examples; Illustrative programs.

Unit III

08 Hrs

Python Control Flow: Types of Control Flow; Control Flow Statements- if, else, elif, while loop, break, continue statements, for loop Statement; range() and exit () functions; Illustrative programs.

Unit IV

08 Hrs

Python Functions: Types of Functions; Function Definition- Syntax, Function Calling, Passing Parameters/arguments, the return statement; Default Parameters; Command line Arguments; Key Word Arguments; Illustrative programs.

Unit V

06 Hrs

Strings: Creating and Storing Strings; Accessing Sting Characters; the str() function; Operations on Strings- Concatenation, Comparison, Slicing and Joining, Traversing; Format Specifiers; Escape Sequences; Raw and Unicode Strings; Python String Methods; Illustrative programs.

References

1. Computer Fundamentals (BPB), P. K. Sinha & Priti Sinha
2. Think Python How to Think Like a Computer Scientist, Allen Downey et al., 2nd Edition, Green Tea Press. Freely available online 2015.
@<https://www.greenteapress.com/thinkpython/thinkCSpy.pdf>
3. Introduction to Python Programming, Gowrishankar S et al., CRC Press, 2019.
4. <http://www.ibiblio.org/g2swap/byteofpython/read/>
5. http://scipy-lectures.org/intro/language/python_language.html
6. <https://docs.python.org/3/tutorial/index.html>

Open Elective

PRINCIPLES OF INTERNET

Course Title: PRINCIPLES OF INTERNET	Course Credits: 3 (3L+0T+0P)
Semester: IV	Duration of SEE: 03 Hour
Total Contact Hours: 42	SEE: 60 Marks IA: 40 Marks

Course Outcomes (COs):

- Able to explain understand the internet standards and recent web technologies like Conferencing, newsgroup etc.
- Able to implement, compile, test and run HTML program
- Able to explain understand web Browsers.

UNIT I

09 Hrs

Basics of Networks – Advantages and disadvantages- Types of networks - LAN, WAN, MAN - Topologies of Networks – Bus Topology, Ring Topology, Star Topology, Mesh Topology, Tree Topology.

UNIT II

06 Hrs

Internet Basics – Advantages and disadvantages – Internet Applications
– IP Address- Internet Services– URL- E-mail - E-mail features – Email Providers.

UNIT III

09 Hrs

Introduction to Internet Protocol –TCP –UDP-FTP – The Telnet –The Usenet – Internet Chat.

UNIT IV

09 Hrs

WWW Overview – Web pages -The Web Servers – Web Browsers– Search Engines-
Introduction to Firewalls – Uses of Firewalls – Types of Firewalls.

UNIT V

09 Hrs

Art of creating a website – Hypertext and HTML – Features – Elements – Attributes- Structure of HTML– Basic tags in HTML – Formatting tags in HTML – HTML Lists –Types of Lists.

Text Book:

1. Raj Kamal , “Internet and Web Technologies”, Mc GrawHill Education,2007.

References:

1. Joe krayank & Joe Habraken, “Internet 6 in 1”, Prentice Hall of India Private Limited, New Delhi, 1998.
2. “Internet Complete”, BPB publications, New Delhi, 1998.

