SCHEME OF EXAMINATION AND COURSE OF STUDY

of subject

Computer Science

Under

National Education Policy of India 2020 (NEP 2020)

for

Bachelor of Science (Hons.)

(Computer Science)

(***w.e.f. Session 2022- 23***)

DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

GURUKULA KANGRI (DEEMED TO BE UNIVERSITY)
HARIDWAR

MAY 2022

DEPARTMENT OF COMPUTER SCIENCE

GURUKULA KANGRI (DEEMED TO BE UNIVERSITY)

SCHEME OF EXAMINATION

COURSE TYPE	PAPER CODE	PAPER NAME	COURSE HOURS			C	EVALUTION SCHEME		
			L	T	P		CIA	ESE	TOTAL
SEMESTER - I									
DSC	BCS-C101	Object Oriented Programming in C++	4	-	-	4	30	70	100
	BCS-C151	Object Oriented Programming in C++ Lab	-	-	4	2	30	70	100
SEMESTER - II									
DSC	BCS-C201	Data Structures and File Processing	4	-	-	4	30	70	100
	BCS-C251	Data Structures and File Processing Lab	-	-	4	2	30	70	100
SEMESTER - III									
DSC	BCS-C301	Database Management System	4	-	-	4	30	70	100
	BCS-C351	Database Management System Lab	-	-	4	2	30	70	100
SEC-1	BCS-S302	Web Technologies	2	-	-	2	30	70	100
	BCS-S352	Web Technologies Lab	-	-	4	2	30	70	100
SEMESTER – IV									
DSC	BCS-C401	Design and Analysis of Algorithms	4	-	-	4	30	70	100
	BCS-C451	Design and Analysis of Algorithms Lab	-	-	4	2	30	70	100
SEC-2	BCS-S402	PHP Programming	2	-	-	2	30	70	100
	BCS-S452	PHP Programming Lab	-	-	4	2	30	70	100
SEMESTER – V									
DSE	Choose any one theory paper and its corresponding lab.								
	BCS-E501	Operating Systems	4	-	-	4	30	70	100
	BCS-E551	Operating Systems Lab	-	-	4	2	30	70	100
	BCS-E502	Artificial Intelligence	4	-	-	4	30	70	100
	BCS-E552	Artificial Intelligence Lab	-	-	4	2	30	70	100
	BCS-E503	Cryptography	4	-	-	4	30	70	100
	BCS-E553	Cryptography Lab	-	-	4	2	30	70	100
SEC-3	Choose any one theory paper and its corresponding lab.								
	BCS-S504	Unix/ Linux Programming	2	-	-	2	30	70	100
	BCS-S554	Unix/ Linux Programming Lab	-	-	4	2	30	70	100
	BCS-S505	Android Programming	2	-	-	2	30	70	100
	BCS-S555	Android Programming Lab	-	-	4	2	30	70	100
SEMESTER – VI									
DSE	Choose any one theory paper and its corresponding lab.								
	BCS-E601	Information Security	4	-	-	4	30	70	100
	BCS-E651	Information Security Lab	-	-	4	2	30	70	100
	BCS-E602	Graph Theory	4	-	-	4	30	70	100
	BCS-E652	Graph Theory Lab	-	-	4	2	30	70	100
	BCS-E603	Computer Graphics	4	-	-	4	30	70	100

COURSE TYPE	PAPER CODE	PAPER NAME	COURSE HOURS			C	EVALUTION SCHEME		
			L	T	P		CIA	ESE	TOTAL
	BCS-E653	Computer Graphics Lab	-	-	4	2	30	70	100
SEC-4	Choose any one theory paper and its corresponding lab.								
	BCS-S604	Java Programming	2	-	-	2	30	70	100
	BCS-S654	Java Programming Lab	-	-	4	2	30	70	100
	BCS-S605	Internet Technologies	2	-	-	2	30	70	100
	BCS-S655	Internet Technologies Lab	-	-	4	2	30	70	100
SEMESTER – VII									
DSC	BCS-C701	Software Engineering	4	-	-	4	30	70	100
	BCS-C751	Software Engineering Lab	-	-	4	2	30	70	100
	BCS-C702	Python Programming	4	-	-	4	30	70	100
	BCS-C752	Python Programming Lab	-	-	4	2	30	70	100
	BCS-O703	Online Course*	-	-	-	4	-	-	100
DSC	BCS-C761	Industrial Training/ Research Project/ Dissertation	-	-	-	6	-	-	100
SEMESTER – VIII									
DSC	BCS-C801	Computer Networks	4	-	-	4	30	70	100
	BCS-C851	Computer Networks Lab	-	-	4	2	30	70	100
	BCS-C802	Computer System Architecture	5	1	-	6	30	70	100
DSE	Choose any one theory paper and its corresponding lab.								
	BCS-E803	Machine Learning	4	-	-	4	30	70	100
	BCS-E853	Machine Learning Lab	-	-	4	2	30	70	100
	BCS-E804	Soft Computing	5	1	-	6	30	70	100
	BCS-E805	Theory of Computation	5	1	-	6	30	70	100
DSC	BCS-C861	Industrial Training/ Research Project/ Dissertation	-	-	-	6	-	-	100

* This course has to be completed by the students from NPTEL/ SWAYAM/ MOOCs/etc. online platform. They should undergo the online course completely, submit assignments, projects, etc. and appear for the final exam conducted by the online instructor. The student can select the course from the approved list of online courses by the Department of Computer Science, GK(DU), Haridwar for the Odd semester. The awarded certificate must be submitted for the award of credits in this course.

L- Lecture **T** – Tutorial **P** - Practical **C**- Credits

DSC - Discipline Specific Core
DSE - Discipline Specific Elective
SEC - Skill Enhancement Course
CIA - Continuous Internal Assessment
ESE - End Semester Examination

Programme Educational Objectives (PEOs)

The graduates of the **BSc Hons. in Computer Science** program will be:

PEO1 Prepared to be employed in IT industries and be engaged in learning,

PEO2 Prepared to be responsible professionals in their domain of interest.

PEO3 Able to apply their technical knowledge as practicing professionals or engaged in higher education.

PEO4 Able to work efficiently as an individual and in a professional team environment.

Programme Outcomes (POs)

PO1 Ability to apply knowledge of computing, mathematics, and basic sciences that may be relevant and appropriate to the domain

PO2 Ability to build the necessary skill set and analytical abilities for developing computer-based solutions for real life problems.

PO3 Ability to communicate scientific information in a clear and concise manner both orally and in writing.

PO4 Developing self-motivating and inspiring team members to engage with the team objectives by using management skills.

PO5 Ability to think, acquire knowledge and skills through logical reasoning and to inculcate the habit of self-learning.

PO6 Ability to identify unethical behavior and adopting objective, unbiased and truthful actions in all aspects of their programme.

PO7 Ability to augment the recent developments in the field of IT and to develop a research culture and Implementation the policies to tackle the burning issues at global and local level.

PO8 Ability to identify opportunities, entrepreneurship vision and use of innovative ideas to create value and wealth for the betterment of the individual and society.

Programme Specific Outcomes (PSOs)

PSO1 Exhibit fundamental concepts of Data Structures, Databases, Operating Systems, Computer networks, Theory of Computation, Advanced Programming, and Software Engineering.

PSO2 Ability to Demonstrate an ability to design, develop, test, debug, deploy, analyze, troubleshoot, maintain, manage and secure software.

PSO3 Ability to explore technical understanding in varied ranges of Computer Science and experience a beneficial environment in cultivating skills for prosperous career and higher studies.

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C101	Object- Oriented Programming in C++	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Students are expected to have some basic knowledge about computers, some knowledge in programming language is preferred.

Course Objectives:

- To design, analyze and evaluate computer programs using the C++ programming language.
- To apply object-oriented programming principles and techniques using C++.

Course Outcomes:

CO1	Describe the procedural and object-oriented paradigm with concepts of streams, classes, functions, data and objects.
CO2	Understand dynamic memory management techniques using pointers, constructors, destructors, etc
CO3	Describe the concept of function overloading, operator overloading, virtual functions and polymorphism.
CO4	Classify inheritance with the understanding of early and late binding, usage of exception handling, generic programming.
CO5	Demonstrate the use of various OOPs concepts with the help of programs.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to C and C++** History of C and C++, Overview of Procedural Programming and Object-Orientation Programming, using main () function, Compiling and Executing Simple Programs in C++.	2
2.	**Data Types, Variables, Constants, Operators and Basic I/O** Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), Formatted and Console I/O (printf(), scanf(), cin, cout), Using Basic Header Files (stdio.h, iostream.h, conio.h etc).	3
3.	**Expressions, Conditional Statements and Iterative Statements** Simple Expressions in C++ (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch-case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)	4
4.	**Functions and Arrays** Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Inline Functions, return data type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Command Line Arguments/Parameters in Functions, Functions with variable number of Arguments. Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, accessing individual elements in an	6

	Array, manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two-Dimensional Array, Working with Rows and Columns), Introduction to Multi-dimensional arrays	
5.	**Derived Data Types (Structures and Unions)** Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, manipulating individual members of structures and unions, Array of Structures, Individual data members as structures, Passing and returning structures from functions, Structure with union as members, Union with structures as members.	3
6.	**Pointers and References in C++** Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables), Pointers to Pointers, Pointers to structures, Passing pointers as function arguments, Returning a pointer from a function, using arrays as pointers, Passing arrays to functions. Pointers vs. References, Declaring and initializing references, Using references as function arguments and function return values	6
7.	**Using Classes in C++** Principles of Object-Oriented Programming, Defining & Using Classes, Class Constructors, Constructor Overloading, Function overloading in classes, Class Variables & Functions, Objects as parameters, Specifying the Protected and Private Access, Copy Constructors, Overview of Template classes and their use.	8
8.	**Overview of Function Overloading and Operator Overloading** Need of Overloading functions and operators, Overloading functions by number and type of arguments, looking at an operator as a function call, Overloading Operators (including assignment operators, unary operators)	5
9.	**Inheritance, Polymorphism and Exception Handling** Introduction to Inheritance (Multi-Level Inheritance, Multiple Inheritance), Polymorphism (Virtual Functions, Pure Virtual Functions), Basics Exceptional Handling (using catch and throw, multiple catch statements), Catching all exceptions, Restricting exceptions, Rethrowing exceptions.	8
10.	**File I/O, Preprocessor Directives** Opening and closing a file (use of fstream header file, ifstream, ofstream and fstream classes), Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files, Understanding the Preprocessor Directives (#include, #define, #error, #if, #else, #elif, #endif, #ifdef, #ifndef and #undef), Macros	3
	Total Lectures	48
Suggested Text Book(s)		
1.	H. Schildt C++, "The Complete Reference Book", McGraw Hill.	
2.	E. Balaguruswamy, "Object Oriented Programming with C++", McGraw Hill.	
3.	J. R. Hubbard, "Programming with C++", Schaum’s Outlines, McGraw Hill.	
4.	R. Albert and T. Breedlove, C++, "An Active Learning Approach", Jones and Bartlett India Ltd.	
Suggested Reference Book(s)		

1.	Stroustrap B., “The C++ Programming Language”, Addison Wesley.
2.	Bruce Eckel, “Thinking in C++”, Pearson.

Other Useful Resource(s)

1.	https://onlinecourses.nptel.ac.in/noc16_cs17/preview
2.	https://onlinecourses.nptel.ac.in/noc17_cs25/announcements
3.	https://www.tutorialspoint.com/cplusplus/
4.	http://www.cplusplus.com/doc/tutorial

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	2	1	-	3	3	**2.1**
CO2	3	3	2	3	3	2	2	3	**2.6**
CO3	3	3	2	1	1	-	3	3	**2.0**
CO4	3	3	2	3	3	2	2	3	**2.6**
CO5	3	3	3	2	1	1	3	3	**2.4**
AVG.	**3.0**	**3.0**	**2.2**	**2.2**	**1.8**	**1.0**	**2.6**	**3.0**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	2	3	**2.7**
CO2	3	2	3	**2.7**
CO3	3	2	3	**2.7**
CO4	3	2	3	**2.7**
CO5	3	2	3	**2.7**
AVG.	**3.0**	**2.0**	**3.0**	**2.7**

SUBJECT: COMPUTER SCIENCE							
DCS	BCS-C151	Object- Oriented Programming in C++ Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Programming concepts

Course Objectives:

- To design, analyze and evaluate computer programs using the C++ programming language.
- To apply object-oriented programming principles and techniques using C++.

Course Outcomes:

CO1	Develop solutions for a range of problems using objects and classes.
CO2	Demonstrate the implementation of constructors, destructors and operator overloading.
CO3	Apply fundamental algorithmic problems including type casting, inheritance, and polymorphism.
CO4	Understand generic programming, templates, file handling.

Lab Exercises

1. Write a menu driven program to perform the following:
    - Calculate the factorial of a given number.
    - Calculate the power of y to x, x and y given by the user.
    - Calculate the nth term of the Fibonacci series.
    - Print the nth prime number.
    - Calculate the sum of the digits of a number.
    - Reverse a number (without treating it as a string).

2. Create an array of structure called employee that contains employee code, employee name, designation and salary. Get the data for any number of employees as per user's choice. Display the details of only those employees who earn more than Rs.20000.

3. An electricity board charges the following rates to domestic users to discourage large consumption of energy:
    - For the first 100 units - ₹ 4.60 per unit
    - For the next 200 units - ₹ 5.80 per unit
    - Beyond 300 units - ₹ 6.90 per unit
    - All the users are charged a minimum of ₹ 250.00. If the total amount is more than ₹ 600 then an additional surcharge of 15% is added.

4. Write a program to read the names of users and number of units consumed and print out the charges with names.

5. Write a program to store sale (in rupees) of three salesmen of five products in a matrix. Calculate and display the average sale of each salesman as well as average sale of each product.

6. Write a menu driven program to perform the following operations on matrices. The matrix is an object with the following operations defined over it. The constructor takes N, the size of the matrix and returns the object.

a. Addition of two matrices.
b. Subtraction of two matrices.
c. Multiply two matrices.
d. Determine the transpose of a matrix.

7. Write a menu driven program to do the following computations, using recursion
   a. Multiplication of two integer numbers given by the user.
   b. Greatest Common Divisor (GCD) of any two given numbers.
   c. nth term of the Fibonacci series.

8. Write a program to maintain accounts in a bank using an array of objects. The program should allow the following four operations :
   a. Create a new account
   b. Withdrawal from an account.
   c. Deposit in an account
   d. Show the balance of the account

9. Define a class Mystring with functions for the following string operations:
   a. Concatenate two strings
   b. Compute the length of a string
   c. Compare two strings
   d. Copy one string on to another
   e. Count the number of occurrences of a substring in a string
   f. Replace a substring
   g. Delete a substring

10. Write a program to create a text file and display the contents using basic file handling operations.

11. Write a program to simulate an arithmetic calculator for integers. The program should be able to produce the last result calculated and the number of arithmetic operations performed so far. Any wrong operation is to be reported. (demonstrate the use of static variable and static function)

12. Write a program that uses a function to check whether a given number is divisible by another number or not. However if the second number is missing, the function checks whether the given number is prime or not

13. Create a class complex having real and imaginary part of complex number as data members. Write a default constructor and parameterized constructor to initialize the complex numbers and write methods to add, subtract, multiply and display complex numbers

14. Write a menu driven program to compare and swap the private data members of two objects from two different classes. Display the result accordingly

15. Create two classes DM and DB which store the value of distances in metres, centimetres and feet, inches respectively. Write a program that can read values for the class objects and add one object of DM with another of DB. The object that stores the results may be a DM or DB object, depending on the units in which the results are required.

16. rite a program that calculates the area of a circle, rectangle and triangle using function overloading. Accept the dimensions of the figure as command line parameters.

17. Create a class Complex having real and imaginary part of a complex numbers as data members. Overload the binary operators +, - and * to perform d on the complex number objects. Write methods for input and output of complex numbers. Overload the binary operators using friend function.

18. Define the given class hierarchy (Three levels) and write functions.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	2	1	3	2	3	2	2	**2.0**
CO2	1	3	2	3	2	3	2	2	**2.3**
CO3	1	3	2	3	2	3	3	3	**2.5**
CO4	2	3	3	3	2	3	3	3	**2.8**
AVG.	**1.3**	**2.8**	**2.0**	**3.0**	**2.0**	**3.0**	**2.5**	**2.5**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**3.0**	**2.5**	**2.7**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C201	Data Structures and File Processing	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Programming knowledge of C/ C++

Course Objectives:

- To impart the basic concepts of data structures.
- To understand concepts about searching and sorting techniques
- To understand basic concepts about stacks, queues, lists, trees and graphs.
- To enable them to write algorithms for solving problems with the help of fundamental data structures.

Course Outcomes:

CO1	Implement linear and binary searches.
CO2	Write the algorithms for Stacks, Queues and singly, doubly & circular linked lists.
CO3	Write the algorithms for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort.
CO4	Implement BST, Graph search and traversal algorithms.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** Basic Terminologies: Elementary Data Organizations, Data Structure Operations: insertion, deletion, traversal etc.; Analysis of an Algorithm, Asymptotic Notations, Time-Space trade off. Searching: Linear Search and Binary Search Techniques and their complexity analysis.	4
2.	**Stacks:** ADT Stack and its operations: Algorithms and their complexity analysis, Applications of Stacks: Expression Conversion and evaluation – corresponding algorithms and complexity analysis.	4
3.	**Queues:** ADT queue, Simple Queue, Circular Queue, Priority Queue. Operations on each types of Queues: Algorithms and their analysis.	4
4.	**Linked Lists:** Singly linked lists: Representation in memory, Traversing, Searching, Insertion into, Deletion; Linked representation of Stack and Queue, Header nodes, Doubly linked list: operations on it and algorithmic analysis; Circular Linked Lists.	8
5.	**Trees:** Basic Tree Terminologies, Different types of Trees: Binary Tree, Threaded Binary Tree, Binary Search Tree, AVL Tree	10
6.	**Sorting and Hashing:** Objective and properties of different sorting algorithms: Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap Sort; Performance and Comparison among all the methods.	10
7.	**File Organizations:** Sequential, indexed sequential, direct, inverted, multi-list, directory systems, Indexing using B-tree, B+ tree and their variants, hashing – hash function, collision handling methods.	8
	Total Lectures	48

Suggested Text Book(s):

1.	M.T. Goodrich, R. Tamassia and D. Mount, Data Structures and Algorithms in C++, John Wiley and Sons, Inc.
2.	Ellis Horowitz, Sartaj Sahni, Fundamentals of Data Structures, , Computer Science Press

Suggested Reference Book(s):

1.	T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, Introduction to Algorithms, 2nd Ed., Prentice- Hall.
2.	Robert L. Kruse and A.J. Ryba, Data Structures and Program Design in C++, Prentice Hall, Inc., NJ.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106102064/
2.	http://www.nptelvideos.in/2012/11/data-structures-and-algorithms.html
3.	https://onlinecourses.nptel.ac.in/noc18_cs25/preview
4.	https://nptel.ac.in/courses/106103069/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	3	1	2	2	3	2	2	**2.1**
CO2	2	3	2	3	2	3	2	2	**2.4**
CO3	2	3	2	2	2	3	3	3	**2.5**
CO4	2	3	2	2	2	3	3	3	**2.5**
AVG.	**2.0**	**3.0**	**1.8**	**2.3**	**2.0**	**3.0**	**2.5**	**2.5**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**3.0**	**2.5**	**2.7**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C251	Data Structures and File Processing Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Programming knowledge of C/ C++

Course Objectives:

- To impart the basic concepts of data structures.
- To understand concepts about searching and sorting techniques
- To understand basic concepts about stacks, queues, lists, trees and graphs.
- To enable them to write algorithms for solving problems with the help of fundamental data structures.

Course Outcomes:

CO1	Implement basic data structures such as arrays and linked list.
CO2	Demonstrate fundamental algorithmic problems including Tree Traversals and Graph traversals.
CO3	Implement various searching and sorting algorithms.
CO4	Demonstrate the implementation of various operations on stack and queue and their application.

Lab Exercises

1. Write programs to implement the following searching techniques.
   a) Linear search
   b) Binary search
2. Write programs to implement the following sorting techniques to arrange a list of integers in ascending order.
   a) Bubble sort
   b) Insertion sort
   c) Selection sort
3. Write programs to implement the following sorting techniques to arrange a list of integers in ascending order.
   a) Quick sort
   b) Merge sort
4. Write programs to
   a) Design and implement Stack and its operations using List.
   b) Design and implement Queue and its operations using List.
5. Write programs for the following:
   a) Uses Stack operations to convert infix expression into postfix expression.
   b) Uses Stack operations to convert infix expression into postfix expression.
   c) Uses Stack operations for evaluating the postfix expression.
6. Write a program to handle the following operations on a singly linked list :
   a) Create a list
   b) Add a node before a given node
   c) Add a node after a given node

d) Delete a node
e) Count the no of nodes
f) Concatenate two lists

7. Design, develop and execute a program to implement doubly linked list where each node consists of integers. The program should support following functions.
   a) Create a doubly linked list
   b) Insert a new node
   c) Delete a node if it is found, otherwise display appropriate message
   d) Display the nodes of doubly linked list

8. Write a program to implement a circular queue using array (perform insert, delete, empty and full queue operations).

9. Using array representation for a polynomial, design, develop and execute a program to add two polynomials and then print the resulting polynomial.

10. Write Python programs to implement the following graph traversal algorithms:
    a) Depth first search.
    b) Breadth first search.

11. Write a program to perform the following:
    a) Create a binary search tree.
    b) Traverse the above binary search tree recursively in pre-order, post-order and in-order.
    c) Count the number of nodes in the binary search tree.

12. Design, develop and execute a program to create a max heap of integers by accepting one element at a time and by inserting it immediately in to heap. Use the array representation of heap. Display the array at the end of insertion phase.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	3	1	2	2	3	2	2	**2.1**
CO2	2	3	2	3	2	3	2	2	**2.4**
CO3	2	3	2	2	2	3	3	3	**2.5**
CO4	2	3	2	2	2	3	3	3	**2.5**
AVG.	**2.0**	**3.0**	**1.8**	**2.3**	**2.0**	**3.0**	**2.5**	**2.5**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**3.0**	**2.5**	**2.7**

SUBJECT: COMPUTER SCIENCE							
DCS	BCS-C301	Database Management System	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Introduction to Computer Programming, Data Structures.

Course Objectives:

- To learn data models, conceptualize and depict a database system using ER diagram.
- To understand the internal storage structures in a physical DB design.
- To know the fundamental concepts of transaction processing techniques.

Course Outcomes:

CO1	For a given specification of the requirement design the databases using E-R method and normalization.
CO2	For a given specification construct the SQL queries.
CO3	For a given transaction-processing system, determine the transaction atomicity, consistency, isolation, and durability.
CO4	Implement the isolation property, including locking, time stamping based on concurrency control and Serializability of scheduling.

Course Contents

UNIT	Contents	Lectures Required
1.	**Database system architecture:** Data Abstraction, Data Independence, Data Definition Language (DDL), Data Manipulation Language (DML). **Data models:** Entity-relationship model, network model, relational and object-oriented data models, integrity constraints, data manipulation operations.	6
2.	**Relational query languages:** Relational algebra, Tuple and domain relational calculus, SQL3, DDL and DML constructs, Open source and Commercial DBMS MYSQL, ORACLE, DB2, SQL server.	8
3.	**Relational database design**: Domain and data dependency, Armstrong's axioms, Normal forms, Dependency preservation, Lossless design.	10
4.	**Query processing and optimization:** Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms.	7
5.	**Storage strategies:** Indices, B-trees, hashing.	4
6.	**Transaction processing:** Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, Multi-version and optimistic Concurrency Control schemes, Database recovery.	7
7.	**Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection.	6
	Total Lectures	48

Suggested Text Book(s):

1.	Elmasri, Navathe, Fundamentals of Database Systems, Pearson Education.
2.	Henry F Korth, Abraham Silberschatz, S. Sudurshan, Database system concepts, McGraw-Hill.

Suggested Reference Book(s):

1.	Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to design, Implementation and Management, Pearson Education.
2.	Bipin C Desai, An Introduction to Database Systems, Galgotia Publications Pvt Limited.
3.	C.J.Date, An Introduction to Database Systems, Pearson Education.

Other Useful Resource(s)

1.	https://www.youtube.com/watch?v=EUzsy3W4I0g&list=PL9426FE14B809CC41
2.	https://www.tutorialspoint.com/dbms/database_normalization.htm
3.	https://www.igi-global.com/journal/journal-database-management/1072
4.	https://www.tutorialspoint.com/dbms/dbms_hashing.htm

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	2	2	3	2	2	**1.8**
CO2	1	3	2	3	2	3	2	2	**2.3**
CO3	2	3	2	2	2	3	3	3	**2.5**
CO4	2	3	2	2	2	3	3	3	**2.5**
AVG.	**1.5**	**2.5**	**1.8**	**2.3**	**2.0**	**3.0**	**2.5**	**2.5**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**3.0**	**2.5**	**2.7**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C351	Database Management System Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Concepts of DBMS.

Course Objectives:

- Introduce ER data model, database design and normalization.
- Learn SQL basics for data definition and data manipulation.

Course Outcomes:

CO1	Implement Basic DDL, DML and DCL commands
CO2	Understand Data selection and operators used in queries and restrict data retrieval and control the display order
CO3	Write sub queries and understand their purpose
CO4	Use Aggregate and group functions to summarize data and Join multiple tables using different types of joins
CO5	Understand the PL/SQL architecture and write PL/SQL code for procedures, triggers, cursors, exception handling etc..

Lab Exercises

1. Create a database having two tables with the specified fields, to computerize a library system of Computer Science Department.

   LIBRARYBOOKS (Accession number, Title, Author, Department, PurchaseDate, Price)

   ISSUEDBOOKS (Accession number, Borrower)

   a) Identify primary and foreign keys.
   b) Create the tables and insert at least 5 records in each table.
   c) Delete the record of book titled 'Database System Concepts',
   d) Change the Department of the book titled Discrete Maths to Computer Science.
   e) List all books that belong to Computer Science department.
   f) List all books that belong to Computer Science department and are written by author 'Navathe'.
   g) List all computer (Department='Computer Science') that have been issued.
   h) List all books which have a price less than 500 or purchased between 01/01/2021 and 01/01/2022.

2. Create a database having three tables to store the details of students of Computer Department.

   STUDENT (Roll_number, StudentName, DateOfBirth, Address, 12_th_Marks, Phone_number)
   PAPER (PaperCode, Papername)
   DETAILS (RollNumber, PaperCode, Attendance, Marks_in_home_examination).

   a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and more than 60% marks in paper 2.

c) List all students who live in 'Delhi' and have marks greater than 60 in paper 1.

d) Find the total attendance and total marks obtained by each student.

e) List the name of student who has got the highest marks in paper 2.

3. Create the following tables and answer the queries given below:

CUSTOMER (CustID, email, Name, Phone, ReferrerID)
BICYCLE (BicycleID, DatePurchased, Color, CustID, ModelNo)
BICYCLEMODEL (ModelNo, Manufacturer, Style)
SERVICE (StartDate, BicycleID, EndDate)

a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.

b) List all the customers who have the bicycles manufactured by manufacturer 'Honda'.

c) List the bicycles purchased by the customers who have been referred by customer 'C1'.

d) List the manufacturer of red colored bicycles.

e) List the models of the bicycles given for service.

4. Create the following tables, enter at least 5 records in each table and answer the queries given below.

EMPLOYEE (Person_Name, Street, City)
WORKS (Person_Name, Company_Name, Salary)
COMPANY (Company_Name, City)
MANAGES (Person_Name, Manager_Name)

a) Identify primary and foreign keys.

b) Alter table employee, add a column ―email of type varchar(20).

c) Find the name of all managers who work for both Samba Bank and NCB Bank.

d) Find the names, street address and cities of residence and salary of all employees who work for 'Punjab National Bank' and earn more than ₹ 50,000.

e) Find the names of all employees who live in the same city as the company for which they work.

f) Find the highest salary, lowest salary and average salary paid by each company.

g) Find the sum of salary and number of employees in each company.

h) Find the name of the company that pays highest salary.

5. Create the following tables, enter at least 5 records in each table and answer the queries given below.

SUPPLIERS (SNo, Sname, Status, SCity)
PARTS (PNo, Pname, Colour, Weight, City)
PROJECT (JNo, Jname, Jcity)
SHIPMENT (Sno, Pno, Jno, Qunatity)

a) Identify primary and foreign keys.

b) Get supplier numbers for suppliers in 'Delhi' with status>20.

c) Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.

d) Get suppliers names for suppliers who do not supply part P2.

e) For each shipment get full shipment details, including total shipment weights.

f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.

g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.

h) Get the names of cities that store more than five red parts.

i) Get full details of parts supplied by a supplier in London.

j) Get part numbers for part supplied by a supplier in 'Mumbai' to a project in 'Mumbai'.

k) Get the total number of projects supplied by a supplier (say, S1).

l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

6. High level language extensions with cursors
7. Data types and create a database and write the program to carry out the following operation.

a. Create tables department and employee with required constraints.

8. Working with null values, matching the pattern from the table.
9. Aggregate functions: grouping the result of a query.
10. Set operators, Nested Queries, Joins and Sequences.
11. Views, indexes, database security and privileges: Grant and Revoke commands, Commit and Rollback commands.
12. PL/SQL Architecture, Assignments and Expressions, Writing PL/SQL Code, Referencing Non-SQL parameters.
13. Triggers and Cursor Management in PL/SQL.
14. Procedures and Functions

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	1	2	3	3	**1.6**
CO2	2	2	2	2	1	2	3	3	**2.1**
CO3	3	2	2	2	2	3	3	3	**2.5**
CO4	3	2	2	2	2	3	3	3	**2.5**
CO5	2	2	2	2	2	3	3	3	**2.4**
AVG.	**2.2**	**1.8**	**1.8**	**1.8**	**1.6**	**2.6**	**3.0**	**3.0**	**2.2**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	1	3	2	**2.0**
CO2	3	3	3	**3.0**
CO3	2	3	3	**2.7**
CO4	2	2	3	**2.3**
CO5	3	3	3	**3.0**
AVG.	**2.2**	**2.8**	**2.8**	**2.6**

SUBJECT: COMPUTER SCIENCE

SEC-1	BCS-S302	Web Technologies	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: NIL

Course Objectives:

- To develop a web application using HTML, DHTMAL and Java script technologies.
- To gain the skills and project-based experience needed for entry into web application and development careers.

Course Outcomes:

CO1	Analyze a web page and identify its elements and attributes.
CO2	Create web pages using HTML and Cascading Style Sheets.
CO3	Build dynamic web pages with validation using Java Script objects and by applying different event handling mechanisms.

Course Contents

UNIT	Contents	Lectures Required
1.	**Basics of Internet and Web:** The basics of Internet, World Wide Web, Web page, Home page, Web site, Static, Dynamic and Active web page, Overview of Protocols – Simple Mail Transfer Protocol, Gopher, Telnet, Emails, TFTP, Simple Network Management Protocol, Hyper Text Transfer Protocol, Client server computing concepts. **Web Client and Web Sever:** Web Browser, Browsers e.g. Netscape navigator, Internet Explorer, Mozilla Firefox, ClientSide Scripting Languages- VB Script and Java Script, Active X control and Plug-ins; Web Server Architecture, Image maps, CGI, API web database connectivity-DBC, ODBC.	5
2.	**Introduction to HTML:** Introduction to HTML, What is HTML and Where did it come from?, HTML Syntax, Semantic Markup, Structure of HTML Documents, Quick Tour of HTML Elements, HTML5 Semantic Structure Elements, Introduction to CSS, What is CSS, CSS Syntax, Location of Styles, Selectors, The Cascade: How Styles Interact, The Box Model, CSS Text Styling. **HTML Tables and Forms:** HTML Tables and Forms, Introducing Tables, Styling Tables, Introducing Forms, Form Control Elements, Table and Form Accessibility, Microformats, **Advanced CSS:** Layout, Normal Flow, Positioning Elements, Floating Elements, Constructing Multicolumn Layouts, Approaches to CSS Layout, Responsive Design, CSS Frameworks.	9
3.	**DHTML:** Dynamic HTML, Document Object Model, Features of DHTML, CSSP (Cascading Style Sheet Positioning) and JSSS (JavaScript assisted Style Sheet), Layers of Netscape, The ID Attribute, DHTML Events.	5
4.	**Java Script:** JS Introduction, Where To, Output, Statements, Syntax, Comments, Variables, Operators, Arithmetic, Assignment, Data Types, Functions, Objects, Events, Strings, String Methods, Numbers, Number Methods, Arrays, Array Methods, Array Sort, Array Iteration, Dates, Date	5

	Formats, Date Get Methods, Date Set Methods, Math, Random, Booleans, Comparisons, Conditions, Switch, Loop For, Loop While, Break, Type Conversion, Bitwise, RegExp, Errors, Scope, Hoisting, Strict Mode, JSON, Forms, Forms API JS Functions, Function Definitions, Function Parameters, Function Invocation, Function Call, Function Apply, Function Closures.	
	Total Lectures	24

Suggested Text Book(s):

1.	Burdman, Collaborative Web Development, Addison Wesley.
2.	Sharma & Sharma, Developing E-Commerce Sites, Addison Wesley
3.	Ivan Bayross, Web Technologies, BPB Publications.

Suggested Reference Book(s):

1.	Nicholas C Zakas, Professional JavaScript for Web Developers", Wrox/Wiley India.
2.	Zak Ruvalcaba Anne Boehm, Murach's HTML5 and CSS3, Murachs/Shroff Publishers & Distributors Pvt Ltd.

Other Useful Resource(s)

1.	https://www.egyankosh.ac.in/handle/123456789/618
2.	https://freevideolectures.com/course/3140/internet-technologies

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	3	2	3	2	2	**2.5**
CO2	3	3	2	3	3	3	3	3	**2.9**
CO3	3	3	2	2	3	3	3	3	**2.8**
AVG.	**3.0**	**3.0**	**2.0**	**2.7**	**2.7**	**3.0**	**2.7**	**2.7**	**2.7**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE

			L	T	P	C	Time for ESE
SEC-1	BCS-S352	Web Technologies Lab	-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of Notepad++.

Course Objectives:

- To Learn the basics involved in publishing content on the World Wide Web.

Course Outcomes:

CO1	Developing webpages using various HTML tags
CO2	Creation of different types of lists using HTML and DHTML.
CO3	Understanding JavaScript.
CO4	Using JavaScript for client-side validations.

Lab Exercises

1. Create a simple HTML page to demonstrate the use of different tags.
2. Design index page of a book on web designing.
3. Display Letter Head of your college on a web page.
4. Create a Hyperlink to move around within a single page rather than to load another page.
5. Display letter using different Text formatting Tags.
6. Design Time Table of your department and highlights of most important periods.
7. Use Tables to provide layout to your web page.
8. Embed Audio and Video into your web page.
9. Divide a web page vertically and horizontally and display logo of your college in left pane and logo of university in right pane.
10. Create a student Bio- Data.
11. Design front page of hospital with different style sheets.
12. Design a web page and display two different pages at a time.
13. Write a program to create a login form. On submitting the form, the user should get navigated to a profile page using JavaScript.
14. Write a code to create a Registration Form. On submitting the form, the user should be asked to login with the new credentials using JavaScript.
15. Write an HTML code to create Gurukula Kangri website/Department website/ Tutorial website for specific subject. Also use Java Script for validation.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	1	2	3	3	**1.6**
CO2	2	2	2	2	1	2	3	3	**2.1**
CO3	3	2	2	2	2	3	3	3	**2.5**
CO4	3	2	2	2	2	3	3	3	**2.5**
CO5	2	2	2	2	2	3	3	3	**2.4**
AVG.	**2.2**	**1.8**	**1.8**	**1.8**	**1.6**	**2.6**	**3.0**	**3.0**	**2.2**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	1	3	2	**2.0**
CO2	3	3	3	**3.0**
CO3	2	3	3	**2.7**
CO4	2	2	3	**2.3**
CO5	3	3	3	**3.0**
AVG.	**2.2**	**2.8**	**2.8**	**2.6**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C401	DESIGN AND ANALYSIS OF ALGORITHMS	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Knowledge of algebra and data structure.

Course Objectives:

- Analyze the asymptotic performance of algorithms.
- Write rigorous correctness proofs for algorithms.
- Demonstrate a familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes:

CO1	For a given algorithms analyze worst-case running times of algorithms based on asymptotic analysis and justify the correctness of algorithms.
CO2	Describe the greedy paradigm and explain when an algorithmic design situation calls for it. For a given problem develop the greedy algorithms.
CO3	Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Synthesize divide-and-conquer algorithms. Derive and solve recurrence relation.
CO4	Describe the dynamic-programming and backtracking paradigms and explain when an algorithmic design situation calls for them. For given problems of dynamic-programming/ backtracking and develop the dynamic programming/ backtracing algorithms, and analyze them to determine its computational complexity.
CO5	Become familiar with the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** Algorithm definition, Algorithm Specification. **Analysis of Algorithms:** Orders of Magnitude (Asymptotic notations), Growth rates, some common bounds (constant, logarithmic, linear, polynomial, exponential), Average and worst-case analysis, Analysing control statements, Recurrence Relations- substitution, change of variables, master's method.	6
2.	**Sorting and searching algorithms:** Selection sort, bubble sort, insertion sort, sorting in linear time, count sort, Linear search.	5
3.	**Divide and conquer algorithms:** Introduction; Quick sort, worst and average case complexity; Merge sort; Matrices multiplication; Binary search.	5
4.	**Greedy algorithms:** General Characteristics of greedy algorithms; Problem solving using Greedy Algorithm- Activity selection problem, Minimum Spanning trees (Kruskal's algorithm, Prim's algorithm)	9

	Graphs: Shortest paths; Purpose of Huffman Coding, Prefix Codes, Huffman Tree, Huffman Coding Algorithm.	
5.	**Dynamic Programming:** Concepts of Dynamic Programming approach for algorithm design, Greedy Algorithm vs Dynamic Programming, Recursion vs Dynamic Programming. Elements of Dynamic Programming Approach. Concept of Matrix Chain Multiplication, its Algorithm, examples and complexity analysis; String Editing Algorithm (edit distance problem with insertion, deletion, replace operation) and its complexity analysis; 0-1 Knapsack problem and its complexity analysis; Travelling Salesman Problem and its analysis. Memoization Strategy Concept of Memoization: Dynamic Programming vs Memoization.	10
6.	**Backtracking:** Concept of Backtracking Approach; Recursion vs Backtracking; Backtracking Algorithms: Concept of Subset Sum, Algorithm for Subset-Sum, its example and Complexity Analysis. 0-1 Knapsack Problem, algorithm with backtracking approach and its analysis; N-Queen Problem and its Analysis.	9
7.	**Introduction to Complexity Theory:** The class P and NP; Polynomial reduction; NP- Complete Problems; NP-Hard Problems.	4
	Total Lectures	48
Suggested Text Book(s):		
1.	Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein, Introduction to algorithms", The MIT Press.	
2.	Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekiaran, Computer Algorithms, Silicon Press.	
3.	Kleinberg, Jon, and Eva Tardos, Algorithm Design", Addison-Wesley.	
Suggested Reference Book(s):		
1.	S. Base, Computer Algorithms: Introduction to Design and Analysis, Addison Wesley.	
2.	A.V. Levitin, Introduction to the Design and Analysis of Algorithms, Pearson Education.	
Other Useful Resource(s)		
1.	https://onlinecourses.nptel.ac.in/noc18_cs20/preview	
2.	https://nptel.ac.in/courses/106101060/	
3.	https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/video_galleries/lecture-videos/	

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	1	1	-	1	3	-	-	**1.1**
CO2	3	1	1	-	1	3	-	-	**1.1**
CO3	3	3	2	1	1	3	2	2	**2.1**
CO4	3	3	2	1	1	3	2	2	**2.1**
CO5	3	3	2	1	1	3	2	2	**2.1**
AVG.	**3.0**	**2.2**	**1.6**	**0.6**	**1.0**	**3.0**	**1.2**	**1.2**	**1.7**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	2	3	**2.7**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
CO5	3	3	3	**3.0**
AVG.	**3.0**	**2.8**	**3.0**	**2.9**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C451	Design and Analysis of Algorithms Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of C/C++ Programming

Course Objectives:

- Student will understand the running time using time library functions.
- To learn to prepare table for input size vs. running time. L
- To learn to measure best run and worst run of the experiments.
- To implement various types of design for an algorithm and compare the approaches.

Course Outcomes:

CO1	Identify the problem given and design the algorithm using various algorithm design techniques.
CO2	Implement various algorithms in a high-level language.
CO3	Analyze the performance of various algorithms.
CO4	Compare the performance of different algorithms for same problem.

Lab Exercises

1. Basic iterative algorithms GCD algorithm, Fibonacci Sequences
2. Sequential and Binary Search.
3. Basic iterative sorting algorithms: Bubble Sort, selection Sort, Insertion Sort.
4. Binary Search with Divide and conquer approach.
5. Merge Sort
6. Heap sort
7. Quick Sort, Randomized Quick Sort.
8. Selection Problem with divide and Conquer approach
9. Fractional Knapsack Problem
10. Job sequencing with deadline
11. Kruskal's algorithm
12. Prims algorithm
13. Dijkstra's Algorithm.
14. Algorithms using Backtracking approach.
15. N-Queen Problem
16. String Editing Algorithm

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	1	1	3	-	1	**1.8**
CO2	3	3	2	1	1	3	-	1	**1.8**
CO3	3	3	2	1	1	3	-	1	**1.8**
CO4	3	3	3	2	2	3	2	2	**2.5**
AVG.	**3.0**	**3.0**	**2.3**	**1.3**	**1.3**	**3.0**	**0.5**	**1.3**	**1.9**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	3	3	2	**2.7**
CO3	3	3	2	**2.7**
CO4	3	3	2	**2.7**
AVG.	**3.0**	**3.0**	**2.0**	**2.7**

SUBJECT: COMPUTER SCIENCE

SEC-2	BCS-S402	PHP Programming	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: Students must have basic knowledge of any text editor like notepad++. Students must also know the background of HTML, Front-End, Back-End & concept of Structure Query Language.

Course Objectives:

- To help the students getting started with web programming using HTML5, PHP and MySQL.
- To learn how to build own website.
- To create dynamic content and user interface.
- Integrating the front end and backend perspective of the application.

Course Outcomes:

CO1	Learn the environment of Server-Side Script.
CO2	Compare and contrast between Client-Side Script & Server-Side Script.
CO3	Learn the use of control structures and numerous native data types with their methods.
CO4	Make Database connectivity between Front End and Back End.
CO5	Develop Dynamic Website that can interact with different kinds of Database Languages.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to PHP:** Evolution of PHP & its comparison Interfaces to External systems, Hardware and Software requirements, PHP Scripting. Basic PHP Development, Working of PHP scripts, Basic PHP syntax, PHP data types. **Displaying type information:** Testing for a specific data type, Changing type with Set type, Operators, Variable manipulation, Dynamic variables and Variable scope.	5
2.	**Control Statements:** if() and elseif() condition Statement, The switch statement, Using the? Operator, Using the while() Loop, The do while statement, Using the for() Loop. **Functions:** Creation, Returning values, Library Functions, User-defined functions, Dynamic function, default arguments, Passing arguments to a function by value.	5
3.	**String Manipulation:** Formatting String for Presentation, Formatting String for Storage, Joining and Splitting String, Comparing String. **Array:** Anatomy of an Array, Creating index based and Associative array, Looping array using each() and foreach() loop. **Forms:** Working with Forms, Super global variables, Super global array, Importing user input, Accessing user input, Combine HTML and PHP code, Using hidden fields, Redirecting the user.	8
4.	**Working with File and Directories:** Understanding file & directory, Opening and closing a file, Coping, renaming and deleting a file, Working with directories, File Uploading & Downloading. Generating Images with PHP: Basics computer Graphics, Creating Image. **Database Connectivity with MySql**: Connection with MySql Database, Performing basic database operation (DML) (Insert, Delete, Update, Select).	6

	Total Lectures	24

Suggested Text Book(s):

1. Steven Holzner, PHP: The Complete Reference, Tata McGraw Hill.
2. Kevin Tetroi, Programming PHP, O' Reilly.
3. Robin Nixon, Learning PHP, MySQL, and JavaScript, Shroff/O'Reilly.

Suggested Reference Book(s):

1. Alan Forbes, The Joy of PHP Programming: A Beginner’s Guide to Programming Interactive Web Applications with PHP and MySQL, Plum Island.
2. Tom Butler & Kevin Yank, PHP & MySQL Novice to Ninja, SitePoint.

Other Useful Resource(s)

1. https://www.tutorialspoint.com/php/php_tutorial.pdf
2. https://www.w3schools.com/php/
3. https://education.fsu.edu/wp-content/uploads/2015/04/Learning-PHP-MySQLJavaScript-and-CSS-2nd-Edition-1.pdf

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	1	3	3	3	-	3	**2.4**
CO2	3	3	1	3	3	3	-	3	**2.4**
CO3	3	3	2	3	3	3	-	2	**2.4**
CO4	3	3	1	3	3	3	-	3	**2.4**
CO5	3	3	1	3	3	3	-	2	**2.3**
AVG.	**3.0**	**3.0**	**1.2**	**3.0**	**3.0**	**3.0**	**-**	**2.6**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
CO5	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE

SEC-2	BCS-S452	PHP Programming Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of programming language and DBMS.

Course Objectives:

- To help the students getting started with web programming using HTML5, PHP and MySQL.
- To learn how to build own website.
- To create dynamic content and user interface.
- Integrating the front end and backend perspective of the application.

Course Outcomes:

CO1	Solve simple to advanced online problems of Web Pages.
CO2	Develop logics of various programming problems using numerous data types and control structures.
CO3	Client Server concepts, Static & Dynamic environment of the websites etc.
CO4	Design and implement the concept of Database connectivity.
CO5	Front-End & Back-End concept of Database System.

Lab Exercises

1. Take values from the user and compute sum, subtraction, multiplication, division and exponent of value of the variables.
2. Write a program to find area of following shapes: circle, rectangle, triangle, square, trapezoid and parallelogram.
3. Compute and print roots of quadratic equation.
4. Write a program to determine whether a triangle is isosceles or not?
5. Print multiplication table of a number input by the user.
6. Calculate sum of natural numbers from one to n number.
7. Print Fibonacci series up to n numbers e.g. - 1 1 2 3 5 8 13 21…..n
8. Write a program to find the factorial of any number.
9. Determine prime numbers within a specific range.
10. Write a program to compute, the Average and Grade of student's marks.
11. Compute addition, subtraction and multiplication of a matrix.
12. Count total number of vowels in a word "Develop & Empower Individuals".
13. Determine whether a string is palindrome or not?
14. Display word after Sorting in alphabetical order.
15. Check whether a number is in a given range using functions.
16. Write a program accepts a string and calculates number of upper-case letters and lower-case letters available in that string.
17. Design a program to reverse a string word by word.
18. Write a program to create a login form. On submitting the form, the user should navigate to profile page.
19. Design front page of a college or department using graphics method.
20. Write a program to upload and download files.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	-	3	-	1	**1.0**
CO2	1	2	1	2	-	3	-	1	**1.3**
CO3	1	1	2	2	-	3	2	2	**1.6**
CO4	1	2	1	2	1	3	2	2	**1.8**
CO5	2	2	1	3	2	3	2	3	**2.3**
AVG.	**1.2**	**1.6**	**1.2**	**2.0**	**0.6**	**3.0**	**1.2**	**1.8**	**1.6**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	-	3	2.0
CO2	3	2	3	2.7
CO3	2	2	3	2.3
CO4	2	2	3	2.3
CO5	3	2	3	2.7
AVG.	**2.6**	**1.6**	**3.0**	**2.3**

SUBJECT: COMPUTER SCIENCE							
DSE	BCS-E501	Operating Systems	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Basics of computers

Course Objectives:

- To learn the mechanisms of OS to handle processes and threads and their communication.
- To learn the mechanisms involved in memory management in contemporary OS.
- To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection algorithms and agreement protocols.
- To know the components and management aspects of concurrency management.

Course Outcomes:

CO1	An appreciation of the role of an operating system.
CO2	Develop algorithms for process scheduling for a given specification of CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.
CO3	For a given specification of memory organization develop the techniques for optimally allocating memory to processes by increasing memory utilization and for improving the access time
CO4	Design and implement file management system.
CO5	For a given I/O devices and OS (specify) develop the I/O management functions in OS as part of a uniform device abstraction by performing operations for synchronization between CPU and I/O controllers.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine. Case study on UNIX and WINDOWS Operating System.	4
2.	**Process:** Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching. Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads. Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time; Scheduling algorithms: Pre-emptive and Non pre-emptive, FCFS, SJF, Priority, RR; Multiprocessor scheduling: Real Time scheduling: RM and EDF .	8

3.	**Inter-process Communication:** Critical Section, Race Conditions, Mutual Exclusion, Hardware Solution, Strict Alternation, Peterson‟s Solution, Lamport‟s Bakery Algorithm, The Producer/ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader‟s & Writer Problem, Dinning Philosopher Problem etc.	8
4.	**Deadlocks:** Definition, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.	6
5.	**Memory Management:** Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition– Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.	7
6.	**Virtual Memory:** Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set , Dirty page/Dirty bit – Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).	7
7.	**I/O Hardware:** I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers, Device drivers, Device independent I/O software, Secondary-Storage Structure: Disk structure, Disk scheduling algorithms. **File Management:** Concept of File, Access methods, File types, File operation, Directory structure, File System structure, Allocation methods (contiguous, linked, indexed), Free-space management (bit vector, linked list, grouping), directory implementation (linear list, hash table), efficiency and performance. Disk Management: Disk structure, Disk scheduling - FCFS, SSTF, SCAN, CSCAN, Disk reliability, Disk formatting, Boot-block, Bad blocks.	8
	Total Lectures	48

Suggested Text Book(s):

1.	Silberschatz, Peter Galvin, Greg Gagne,
2.	William Stallings, Operating Systems: Internals and Design Principles, Pearson.

Suggested Reference Book(s):

1.	Charles Crowley, Operating System: A Design-oriented Approach, Irwin Publishing.
2.	Gary J. Nutt, Operating Systems: A Modern Perspective, Addison-Wesley.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106108101//
2.	https://nptel.ac.in/courses/106106144/

3.	https://www.class-central.com/course/udacity-introduction-to-operating-systems-3419

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	1	3	-	1	**1.1**
CO2	1	2	1	2	1	3	-	1	**1.4**
CO3	1	1	2	2	1	3	2	2	**1.8**
CO4	2	2	1	2	2	3	2	2	**2.0**
CO5	2	2	1	3	2	3	3	3	**2.4**
AVG.	**1.4**	**1.6**	**1.2**	**2.0**	**1.4**	**3.0**	**1.4**	**1.8**	**1.7**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	-	3	**2.0**
CO2	3	2	3	**2.7**
CO3	2	2	3	**2.3**
CO4	2	2	3	**2.3**
CO5	3	2	3	**2.7**
AVG.	**2.6**	**1.6**	**3.0**	**2.3**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E551	Operating Systems Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Concepts of operating systems and programming language (C/C++).

Course Objectives:

- Understand concepts such as Scheduling, Memory Management, Multi-threading, required in the design of Operating Systems.
- Analyze the various scheduling algorithms, fitting strategies, deadlock detection.

Course Outcomes:

CO1	Understand key mechanisms in design of operating systems modules
CO2	Understand process management, concurrent processes and threads, memory management, virtual memory concepts, deadlocks
CO3	Compare performance of processor scheduling algorithms
CO4	Produce algorithmic solutions to process synchronization problem
CO5	Learn file and disk space management

Lab Exercises

1. Process Handling
2. FCFS Scheduling Algorithm
3. SJF Scheduling Algorithm
4. Priority Scheduling Algorithm
5. Round-Robin Scheduling Algorithm
6. Inter-Process Communication
7. To simulate the following contiguous memory allocation Techniques:

   a) Worst fit b) Best fit c) First fit.
8. Shared Memory Concept
9. Peterson‟s Critical Section Problem Solution
10. Banker‟s Algorithm
11. Page Replacement algorithm: FIFO
12. Page Replacement algorithm: LRU
13. Page Replacement algorithm: Optimal Replacement
14. Disk Scheduling: C-SCAN
15. Disk Scheduling: C-LOOK

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	1	3	-	1	1.1
CO2	1	2	1	2	1	3	2	1	1.6
CO3	1	1	2	2	1	3	2	2	1.8
CO4	2	2	2	2	2	3	2	3	2.3
CO5	2	2	2	3	2	3	3	3	2.5
AVG.	1.4	1.6	1.6	2.0	1.4	3.0	1.8	2.0	1.9

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	-	3	2.0
CO2	3	2	3	2.7
CO3	2	2	3	2.3
CO4	2	2	3	2.3
CO5	3	2	3	2.7
AVG.	2.6	1.6	3.0	2.3

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E502	Artificial Intelligence	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Knowledge of linear algebra, statistics and programming.

Course Objectives:

- To acquaint the student with basic concepts of Artificial Intelligence.
- To aware the student about Theory and practical techniques of artificial intelligence.
- This course would provide emphasis to the principles and applications of Artificial Intelligence.

Course Outcomes:

CO1	Understand what Artificial Intelligence mean and the foundations of it.
CO2	Understand those elements constituting problems and learn to solve it by various uninformed and informed (heuristics based) searching techniques
CO3	Understand the formal method for representing the knowledge and the process of inference to derive representations of the knowledge to deduce what to do.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behaviour and environment.	5
2.	**Problem Solving and Searching Techniques:** Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis, Introduction to Game Playing, Min-Max and Alpha-Beta pruning algorithms.	15
3.	**Knowledge Representation:** Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs. Programming in Logic (PROLOG);	15
4.	**Dealing with Uncertainty and Inconsistencies:** Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.	7
5.	**Understanding Natural Languages:** Parsing Techniques, Context-Free and Transformational Grammars, Recursive and Augmented Transition Nets.	6
	Total Lectures	48

Suggested Text Book(s):

1.	DAN.W. Patterson, Introduction to A.I and Expert Systems – PHI.
2.	Russell & Norvig, Artificial Intelligence-A Modern Approach, Prentice Hall.

3.	W.F. Clocksin and Mellish, Programming in PROLOG, Narosa Publishing House.

Suggested Reference Book(s):

1.	Ivan Bratko, Prolog Programming for Artificial Intelligence, Addison-Wesley, Pearson Education
2.	Rich & Knight, Artificial Intelligence – Tata McGraw Hill.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106105077
2.	https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/video_galleries/lecture-videos/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	3	3	-	2	3	2	2	**2.0**
CO2	2	3	2	1	1	3	2	1	**1.9**
CO3	1	3	2	2	2	3	3	3	**2.4**
AVG.	**1.3**	**3.0**	**2.3**	**1.0**	**1.7**	**3.0**	**2.3**	**2.0**	**2.1**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	1	1	3	**1.7**
CO2	2	1	3	**2.0**
CO3	2	2	2	**2.0**
AVG.	**1.7**	**1.3**	**2.7**	**1.9**

SUBJECT: COMPUTER SCIENCE							
DSE	BCS-E552	Artificial Intelligence Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Concepts of Prolog

Course Objectives:

- To introduce PROLOG language, Data Objects and basic concepts (Atoms, Variables, Structures, Predicate, Facts, Clauses, Queries and Rules) used in prolog.

Course Outcomes:

CO1	Explain Unification, Recursion and Listing using Prolog.
CO2	Apply Uninformed Search strategy using Prolog.
CO3	Apply Heuristic Search strategy using Prolog.

Lab Exercises

1. Write a prolog program to calculate the sum of two numbers.
2. Write a prolog program to find the maximum of two numbers.
3. Write a prolog program to calculate the factorial of a given number.
4. Write a prolog program to calculate the nth Fibonacci number.
5. Write a prolog program, insert_nth (item, n, into_list, result) that asserts that result is the list into_list with item inserted as the n‘th element into every list at all levels.
6. Write a Prolog program to remove the Nth item from a list.
7. Write a Prolog program, remove-nth(Before, After) that asserts the After list is the Before list with the removal of every n‘th item from every list at all levels.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome(List).
10. Write a Prolog program to implement max(X,Y,Max) so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement sumlist(List,Sum) so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates evenlength(List) and oddlength(List) so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement reverse(List,ReversedList) that reverses lists.
15. Write a Prolog program to implement maxlist(List,Max) so that Max is the greatest number in the list of numbers List using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.
17. Write a prolog program that implements Semantic Networks/Frame Structures.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	3	2	3	2	2	**2.5**
CO2	3	3	2	3	1	3	2	3	**2.5**
CO3	3	3	2	2	2	3	3	3	**2.6**
AVG.	**3.0**	**3.0**	**2.0**	**2.7**	**1.7**	**3.0**	**2.3**	**2.7**	**2.5**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E503	Cryptography	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Knowledge of Data Structure

Course Objectives:

- To impart an essential study of computer security issues
- To develop basic knowledge on cryptography
- To impart an essential study of various security mechanisms

Course Outcomes:

CO1	Acquire fundamental knowledge on the concepts of finite fields and number theory.
CO2	Understand, compare and apply various private-key encryption and decryption techniques.
CO3	Study and analyze various public-key encryption and decryption techniques.
CO4	Learn the various techniques for message authentication.

Course Contents

UNIT	Contents	Lectures Required
1.	**Elementary number theory:** Prime numbers, Fermat's and Euler's theorems, Testing for primality, Chinese remainder theorem, discrete logarithms.	8
2.	**Finite fields:** Review of groups, rings and fields; Modular Arithmetic, Euclidean Algorithms, Finite fields of the form GF(p), Polynomial Arithmetic, Finite fields of the form GF(2).	10
3.	**Data Encryption Techniques:** Algorithms for block and stream ciphers, private key encryption –DES, AES, RC4.	10
4.	**Algorithms for public key encryption:** RSA, DH Key exchange, KERBEROS, elliptic curve cryptosystems.	10
5.	Message authentication and hash functions, Digital Signatures and authentication protocols, Public key infrastructure, Cryptanalysis of block and stream ciphers.	10
	Total Lectures	48

Suggested Text Book(s):

1.	W. Stallings, Cryptography and Network Security Principles and Practices, Prentice-Hall of India.
2.	C. Pfleeger and S.L. Pfleeger, Security in Computing, Prentice-Hall of India.
3.	Goldreich, O. Foundations of Cryptography: Basic Tools, New York, NY: Cambridge University

Suggested Reference Book(s):

1.	Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press.
2.	Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, Willy.
3.	Christof Paar, Understanding Cryptography: A Textbook for Students and Practitioners, Springer.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106105031
2.	https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/resources/lecture-21-cryptography-hash-functions/
3.	https://ocw.mit.edu/courses/6-046j-design-and-analysis-of-algorithms-spring-2015/resources/lecture-22-cryptography-encryption/
4.	Virtual Labs: http://cse29-iiith.virtual-labs.ac.in/index.php?section=Experiments Students are advised to practice virtual lab experiments at above link as and when the topics are covered in the class.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	1	1	2	3	2	2	**2.1**
CO2	3	3	2	1	1	3	2	2	**2.1**
CO3	3	3	2	2	1	3	2	3	**2.4**
CO4	3	3	3	2	2	3	2	3	**2.6**
AVG.	**3.0**	**3.0**	**2.0**	**1.5**	**1.5**	**3.0**	**2.0**	**2.5**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
CO4	3	3	2	**2.7**
AVG.	**3.0**	**3.0**	**2.8**	**3.0**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E553	Cryptography Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Fundamentals of computer networks and programming language.

Course Objectives:

- To provide deeper understanding into cryptography, its application to network security, threats/vulnerabilities to networks and countermeasures.
- To explain various approaches to Encryption techniques, strengths of Traffic Confidentiality, Message Authentication Codes.

Course Outcomes:

CO1	Apply the cryptographic algorithms for data communication.
CO2	Compare the performance of various security algorithms.
CO3	Apply the Digital signature for secure data transmission.
CO4	Utilize the different open source tools for network security and analysis.

Lab Exercises

1. Implement the encryption and decryption of 8-bit data using 'Simplified DES Algorithm' in 'C'.
2. Implement 'Linear Congruential Algorithm' to generate 5 pseudo-random numbers in 'C'.
3. Implement Rabin-Miller Primality Testing Algorithm in 'C'.
4. Implement the Euclid Algorithm to generate the GCD of an array of 10 integers in 'C'.
5. Implement RSA algorithm for encryption and decryption in 'C'.
6. Configure a mail agent to support Digital Certificates, send a mail and verify the correctness of this system using the configured parameters.
7. Configure SSH (Secure Shell) and send/receive a file on this connection to verify the correctness of this system using the configured parameters.
8. Configure a firewall to block the following for 5 minutes and verify the correctness of this system using the configured parameters:

   (i) Two neighbourhood IP addresses on your LAN
   (ii) All ICMP requests
   (iii) All TCP SYN Packets

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	2	1	1	2	3	2	2	**2.0**
CO2	3	3	2	1	2	3	2	2	**2.3**
CO3	3	3	2	2	2	3	3	3	**2.6**
CO4	3	3	3	2	2	3	3	3	**2.8**
AVG.	**3.0**	**2.8**	**2.0**	**1.5**	**2.0**	**3.0**	**2.5**	**2.5**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	2	3	**2.7**
CO2	3	2	3	**2.7**
CO3	3	2	3	**2.7**
CO4	3	2	3	**2.7**
AVG.	**3.0**	**2.0**	**3.0**	**2.7**

SUBJECT: COMPUTER SCIENCE

SEC-3	BCS-S504	Unix/ Linux Programming	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: Basic of Operating system

Course Objectives:

- To teach principles of operating system including File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking Commands, Basic Linux commands, Scripts and filters.
- To familiarize fundamentals of the Bourne again shell (bash), shell programming, pipes, input and output redirection Control structures, arithmetic in shell interrupt processing, functions, debugging shell scripts.

Course Outcomes:

CO1	Ability to use various Unix/ Linux commands that are used to manipulate system operations at admin level and a prerequisite to pursue job as a Network administrator.
CO2	Ability to write Shell Programming using Unix/ Linux commands.
CO3	Ability to design and write application to manipulate internal kernel level Unix/ Linux File System.
CO4	Ability to write Shell Programming using Unix/ Linux commands.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** What is LINUX/UNIX Operating systems, Difference between LINUX/UNIX and other operating systems Features and Architecture, Various Distributions available in the market Installation, Booting and shutdown process, System processes (an overview), External and internal commands, Creation of partitions in OS, Processes and its creation phases – Fork, Exec, wait.	6
2.	**User Management and the File System:** Types of Users, creating users, Granting rights; User management commands, File quota and various file systems available; File System Management and Layout, File permissions; Login process, Managing Disk Quotas; Links (hard links, symbolic links).	6
3.	**Shell introduction and Shell Scripting:** What is shell and various type of shell, Various editors present in Linux; Different modes of operation in vi editor; What is shell script, Writing and executing the shell script; Shell variable (user defined and system variables; System calls, Using system calls; Pipes and Filters; Decision making in Shell Scripts (If else, switch), Loops in shell; Functions; Utility programs (cut, paste, join, tr, unique utilities); Pattern matching utility (grep).	12
	Total Lectures	24

Suggested Text Book(s):

1.	Sumitabha, Das, Unix Concepts and Applications, Tata McGraw-Hill Education.

2.	Michael Jang RHCSA/ RHCE Red Hat Linux Certification: Exams (Ex200 & Ex300) (Certification Press).

Suggested Reference Book(s):

1.	Nemeth Synder & Hein, Linux Administration Handbook, Pearson Education.
2.	W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Unix Network Programming, The Sockets Networking API.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/117106113
2.	http://www.nitttrc.edu.in/nptel/courses/video/117106113/L11.html

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAG E
CO1	1	2	1	3	2	3	2	2	**2.0**
CO2	1	3	2	3	2	3	2	2	**2.3**
CO3	1	3	2	3	2	3	3	3	**2.5**
CO4	2	3	3	3	2	3	3	3	**2.8**
AVG.	**1.3**	**2.8**	**2.0**	**3.0**	**2.0**	**3.0**	**2.5**	**2.5**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**3.0**	**2.5**	**2.7**

SUBJECT: COMPUTER SCIENCE

SEC-3	BCS-S554	Unix/ Linux Programming Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of Unix/ Linux OS.

Course Objectives:

- To learn basics of shell programming.
- Shell programs to understand the operating environment of Linux are practiced.

Course Outcomes:

CO1	Apply Unix/ Linux operating system commands.
CO2	Understand different Unix/ Linux shell scripts and execute various shell programs.

Lab Exercises

1. Write a shell script to check if the number entered at the command line is prime or not.
2. Write a shell script to modify —call command to display calendars of the specified months.
3. Write a shell script to modify —call command to display calendars of the specified range of months.
4. Write a shell script to accept a login name. If not a valid login name display message – —Entered login name is invalid.
5. Write a shell script to display date in the mm/dd/yy format.
6. Write a shell script to display on the screen sorted output of —who command along with the total number of users .
7. Write a shell script to display the multiplication table any number,
8. Write a shell script to compare two files and if found equal asks the user to delete the duplicate file.
9. Write a shell script to find the sum of digits of a given number.
10. Write a shell script to merge the contents of three files, sort the contents and then display them page by page.
11. Write a shell script to find the LCD(least common divisor) of two numbers.
12. Write a shell script to perform the tasks of basic calculator.
13. Write a shell script to find the power of a given number.
14. Write a shell script to find the binomial coefficient C(n , x).
15. Write a shell script to find the permutation P(n,x).
16. Write a shell script to find the greatest number among the three numbers.
17. Write a shell script to find the factorial of a given number.

18. Write a shell script to check whether the number is Armstrong or not.

19. Write a shell script to check whether the file have all the permissions or not.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	3	2	3	2	3	3	3	**2.5**
CO2	1	3	2	3	2	3	3	3	**2.5**
AVG.	**1.0**	**3.0**	**2.0**	**3.0**	**2.0**	**3.0**	**3.0**	**3.0**	**2.5**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE

SEC-3	BCS-S505	Android Programming	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: Basic knowledge of object- oriented programming language (Java).

Course Objectives:

- To facilitate students to understand android SDK
- To help students to gain a basic understanding of Android application development
- To inculcate working knowledge of Android Studio development tool

Course Outcomes:

CO1	Comprehend Android platform and its usefulness in application development.
CO2	Acquire skill set to execute applications in Android based devices.
CO3	Design and develop deployable Android applications.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to Android**: The Android Platform, Android SDK, Android Studio installation, Android Installation, building First Android application, Understanding Anatomy of Android Application, Android Manifest file. **Android Application Design Essentials**: Anatomy of an Android applications, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions.	8
2.	**Introduction to Android**: The Android Platform, Android SDK, Android Studio installation, Android Installation, building First Android application, Understanding Anatomy of Android Application, Android Manifest file. **Android Application Design Essentials**: Anatomy of an Android applications, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions.	8
3.	**Introduction to Android**: The Android Platform, Android SDK, Android Studio installation, Android Installation, building First Android application, Understanding Anatomy of Android Application, Android Manifest file. **Android Application Design Essentials**: Anatomy of an Android applications, Android terminologies, Application Context, Activities, Services, Intents, Receiving and Broadcasting Intents, Android Manifest File and its common settings, Using Intent Filter, Permissions.	8
	Total Lectures	24

Suggested Text Book(s):

1.	Meier Reto and Lake Ian, *Professional Android,* Wrox.
2.	John Horton, Android Programming for Beginners, Packt Publishing

Suggested Reference Book(s):

1.	Deitel, P., Deitel, H., Deitle, A., and Morgano, M., Android for Programmers – An App-Driven Approach, Prentice Hall, Upper Saddle River, NY.

Other Useful Resource(s)

1.	http://www.developer.android.com
2.	http://developer.android.com/about/versions/index.html
3.	http://developer.android.com/training/basics/firstapp/index.html
4.	http://developer.android.com/guide/components/fundamentals.html
5.	http://developer.android.com/guide/components/intents-filters.html .
6.	http://developer.android.com/training/multiscreen/screensizes.html
7.	http://developer.android.com/guide/topics/ui/controls.html
8.	http://developer.android.com/guide/topics/ui/declaring-layout.html
9.	http://developer.android.com/training/basics/data-storage/databases.html
10.	http://www.developer.android.com

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	3	2	3	2	2	**2.5**
CO2	3	3	2	3	3	3	3	3	**2.9**
CO3	3	3	2	2	3	3	3	3	**2.8**
AVG.	**3.0**	**3.0**	**2.0**	**2.7**	**2.7**	**3.0**	**2.7**	**2.7**	**2.7**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE							
SEC-3	BCS-S555	Android Programming	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of Unix/ Linux OS.

Course Objectives:

- To Develop competence and confidence in android programming.
- To understand the entire Android Apps Development Cycle.
- To enable the students to independently create Android Applications.

Course Outcomes:

CO1	Demonstrate the Understanding of fundamental of Android Programming.
CO2	Build their ability to develop software with reasonable complexity on mobile platform.
CO3	Discover the life cycles of Activities, Applications, intents and fragments.
CO4	Design the Android apps by using Java Concepts.

Lab Exercises

1. Install Android Studio with Specific Latest SDK in your System.
2. Develop an android app which displays "Hello World" message.
3. Develop an android app which displays a form to get following information from user. 1) Username 2) Password 3) Email Address 4) Phone Number 5) Country

   Form should be followed by a Button with label "Submit". When user clicks the button, a message should be displayed to user describing the information entered.

   Utilize suitable UI controls (i.e. widgets). [When user enters country in Auto Complete TextView, list of states should be displayed in Spinner automatically.
4. Create sample application that demonstrates activity life cycle's all methods.
5. Using Android, Create a login Activity. It asks "username" and "password" from user. If username and password are valid, it displays Welcome message using new activity.
6. "Happy Birth Day" App using TextView and ImageView
7. Create "Hello Toast" App by implementing a click handler method for the button to display a message on the screen when the user clicks. Use Linear Layout for creating view.
8. Create the MP3 player like application with service
9. The Simple Calculator app has two edit texts and four buttons. When you enter two numbers and click a button, the app performs the calculation for that button and displays the result.
10. Develop one App. Which Contains Specific User Interface and design Interface.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	2	1	3	3	1	3	3	**2.0**
CO2	1	2	1	3	3	2	3	3	**2.0**
CO3	1	2	2	3	3	1	3	3	**2.2**
CO4	1	2	2	3	3	2	3	3	**2.3**
AVG.	**1.0**	**2.0**	**1.5**	**3.0**	**3.0**	**1.0**	**3.0**	**3.0**	**2.1**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	3	3	3	**3.0**
CO3	3	3	3	**3.0**
CO4	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**3.0**	**3.0**

SUBJECT: COMPUTER SCIENCE

			L	T	P	C	Time for ESE
DSE	BCS-E601	Information Security	4	-	-	4	3 Hrs.

Pre- requisite: Mathematical concepts: Random numbers, Number theory, finite fields

Course Objectives:

- To understand the basics of Information Security
- To know the legal, ethical and professional issues in Information Security
- To know the aspects of risk management
- To know the technological aspects of Information Security

Course Outcomes:

CO1	Explore the basic principles of the symmetric cryptography and techniques with their strengths and weaknesses from perspective of cryptanalysis
CO2	Implement and analyze various symmetric key cryptography algorithms and their application in different context.
CO3	Compare public key cryptography with private key cryptography and Implement various asymmetric key cryptography algorithms.
CO4	Explore the concept of hashing and implement various hashing algorithms for message integrity.
CO5	Explore and use the techniques and standards of digital signature, key management and authentication.

Course Contents

UNIT	Contents	Lectures Required
1.	**Overview of Security:** Protection versus security; aspects of security–data integrity, data availability, privacy; security problems, user authentication, Orange Book.	8
2.	**Security Threats**: Program threats, worms, viruses, Trojan horse, trap door, stack and buffer overflow; system threats- intruders; communication threats-tapping and piracy.	10
3.	**Cryptography:** Substitution, transposition ciphers, symmetric-key algorithms-Data Encryption Standard, advanced encryption standards, public key encryption - RSA; Diffie-Hellman key exchange, ECC cryptography, Message Authentication- MAC, hash functions.	10
4.	**Digital signatures:** Symmetric key signatures, public key signatures, message digests, public key infrastructures.	10
5.	**Security Mechanisms:** Intrusion detection, auditing and logging, tripwire, system-call monitoring.	10
	Total Lectures	48

Suggested Text Book(s):

1.	W. Stallings, Cryptography and Network Security Principles and Practices, Prentice-Hall of India.
2.	C. Pfleeger and SL. Pfleeger, Security in Computing, Prentice-Hall of India.
3.	D. Gollmann, Computer Security, John Wiley and Sons, NY.

Suggested Reference Book(s):

1.	J. Piwprzyk, T. Hardjono and J. Seberry, Fundamentals of Computer Security, Springer-Verlag Berlin.
2.	J.M. Kizza, Computer Network Security, Springer.
3.	M. Merkow and J. Breithaupt, Information Security: Principles and Practices, Pearson Education.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106106199
2.	https://nptel.ac.in/courses/106106141
3.	https://ocw.mit.edu/courses/6-857-network-and-computer-security-spring-2014/pages/lecture-notes-and-readings/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	1	1	2	2	3	1	2.0
CO2	3	3	1	2	-	1	3	2	1.9
CO3	3	3	1	2	-	1	3	2	1.9
CO4	3	3	1	2	1	1	3	2	2.0
CO5	3	3	1	1	1	2	3	2	2.0
AVG.	3.0	3.0	1.0	1.6	0.8	1.4	3.0	1.8	2.0

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	1	2.3
CO2	1	3	2	2.0
CO3	1	3	2	2.0
CO4	3	3	2	2.7
CO5	2	3	1	2.0
AVG.	2.0	3.0	1.6	2.1

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E651	Information Security Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Fundamentals of Networking Technologies.

Course Objectives:

- To introduce students to a broad range of network security-related topics including: confidentiality, integrity, authentication, and non-repudiation.

Course Outcomes:

CO1	Use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
CO2	Use of Password cracking tools
CO3	Performing encryption and decryption
CO4	Capturing and modifying the message
CO5	Use of gpg utility for signing and encrypting purposes

Lab Exercises

1. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
2. Use of Password cracking tools: John the Ripper, Ophcrack. Verify the strength of passwords using these tools.
3. Perform encryption and decryption of Caesar cipher. Write a script for performing these operations.
4. Perform encryption and decryption of a Rail fence cipher. Write a script for performing these operations.
5. Use nmap/zenmap to analyse a remote machine.
6. Use Burp proxy to capture and modify the message.
7. Demonstrate sending of a protected word document.
8. Demonstrate sending of a digitally signed document.
9. Demonstrate sending of a protected worksheet.
10. Demonstrate use of steganography tools.
11. Demonstrate use of gpg utility for signing and encrypting purposes.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	3	1	2	3	3	2	2	2.1
CO2	3	3	1	2	3	3	3	2	2.5
CO3	3	3	1	2	3	3	3	2	2.5
CO4	3	3	1	2	3	3	3	2	2.5
CO5	3	3	1	1	3	3	3	2	2.4
AVG.	2.6	3.0	1.0	1.8	3.0	3.0	2.8	2.0	2.4

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	1	**2.3**
CO2	3	3	2	**2.7**
CO3	3	3	2	**2.7**
CO4	3	3	2	**2.7**
CO5	3	3	2	**2.7**
AVG.	**3.0**	**3.0**	**1.8**	**2.6**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E602	Graph Theory	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Knowledge of data structures.

Course Objectives:

- To explain basic concepts in combinatorial graph theory.
- To define how graphs, serve as models for many standard problems.

Course Outcomes:

CO1	Solve problems using basic graph theory.
CO2	Identify induced subgraphs, matchings, covers in graphs.
CO3	Determine whether graphs are Hamiltonian and/or Eulerian.
CO4	Solve problems involving vertex and edge connectivity, planarity and crossing numbers.
CO5	Solve problems involving vertex and edge coloring.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction**: Applications of Graphs; Finite and Infinite Graphs; Incidence and Degree; Isolated and Pendant Vertex; Isomorphism; Sub Graph, Walks, Paths and Circuits; Connected and Disconnected Graphs; Components of A Graph; Euler Graphs; Hamiltonian Paths and Circuits; The Traveling Salesman Problem.	8
2.	**Trees, Circuits and Cut-sets**: Properties of Trees; Pendant Vertices in A Tree; Center of A Tree; Rooted and Binary Trees; Spanning Tree, Spanning Trees in A Weighted Graph, Algorithm for Shortest Spanning Tree, Fundamental Circuits, Cut-sets and Cut Vertices; Fundamental Cut-sets, Connectivity and Separability.	8
3.	**Planar Graphs**: Combinatorial Vs Geometric Graphs; Planar Graph; Kuratowski's Graphs; Detection of Planarity; Geometric Dual; Thickness and Crossings.	7
4.	**Matrix representation and coloring**: Path Matrix, Cut- Set Matrix, Circuit Matrix, Incidence Matrix, Adjacency Matrix and Their Properties. Chromatic Number, Chromatic Polynomial, Chromatic Partitioning, Matchings, Covering and Four-Color Problem.	7
5.	**Directed Graphs**: Digraphs and Binary Relations; Directed Path and Connectedness; Adjacency Matrix of Digraph.	6
6.	**Directed Tree**: Arborescence; Paired Comparison and Tournaments; Counting Labeled and Unlabeled Trees.	6
7.	**Algorithms:** Shortest path, minimal spanning tree, Connectedness and components, Fundamental circuits, Cut- vertices and separability, Isomorphism.	6
	Total Lectures	48

Suggested Text Book(s):

1.	N. Deo, Graph Theory with Applications to Engineering and Computer Science, PHI.
2.	Richard J. Trudeau, Introduction to Graph Theory, Dover Publications Inc.

Suggested Reference Book(s):

1.	JA Bondy and USR Murty, Graph theory with applications. Bulletin of the American Mathematical Society, The Macmillian Press Ltd.
2.	Doughlous B. West, Introduction to graph theory (Vol. 2). Upper Saddle River, NJ: Prentice hall.
3.	Gary Chartard and Ping Zhang, A First Course in Graph Theory, Courier Corporation.
4.	Geir Agnarsson and Raymond Greenlaw, Graph Theory: Modelling, Applications, and Algorithms, Pearson/Prentice Hall.

Other Useful Resource(s)

1.	http://nptel.ac.in/courses/111106050/13
2.	https://ocw.mit.edu/courses/18-217-graph-theory-and-additive-combinatorics-fall-2019/video_galleries/video-lectures/
3.	https://ocw.mit.edu/courses/18-315-combinatorial-theory-introduction-to-graph-theory-extremal-and-enumerative-combinatorics-spring-2005/pages/lecture-notes/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	1	3	2	2	2	2.3
CO2	3	2	1	1	3	2	2	1	1.9
CO3	3	2	1	1	3	2	2	1	1.9
CO4	3	2	1	2	3	2	3	1	2.1
CO5	3	2	1	2	3	2	3	1	2.1
AVG.	3.0	2.2	1.2	1.4	3.0	2.0	2.4	1.2	2.1

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	2	3	2.3
CO2	2	1	3	2.0
CO3	2	1	3	2.0
CO4	2	2	3	2.3
CO5	2	2	3	2.3
AVG.	2.0	1.6	3.0	2.1

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E652	Graph Theory Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of any programming language and data structures.

Course Objectives:

- To explain basic concepts in combinatorial graph theory.
- To define how graphs, serve as models for many standard problems.

Course Outcomes:

CO1	Solve problems using basic graph theory.
CO2	Identify induced subgraphs, matchings, covers in graphs.
CO3	Determine whether graphs are Hamiltonian and/or Eulerian.
CO4	Solve problems involving vertex and edge connectivity, planarity and crossing numbers.
CO5	Solve problems involving vertex and edge coloring.
CO6	Model real world problems using graph theory.

Lab Exercises

1. Program to implement Simple Path Graph.
2. Program to construct Graph with Simple cycles.
3. Program for computing average degree of nodes in a graph.
4. Program to find nodes of Odd/Even degree.
5. Program to find minimum distance pairs.
6. Program to create Complete graph.
7. Program to compute minimum weight matching in a graph.
8. Program to implement augment and original graph.
9. Program to compute Eulerian circuit.
10. Program to compute eigen values.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	2	1	3	3	2	2	**2.4**
CO2	3	3	2	1	3	3	2	2	**2.4**
CO3	3	2	2	1	3	3	2	1	**2.1**
CO4	3	2	2	2	3	3	3	2	**2.5**
CO5	3	2	2	2	3	3	3	2	**2.5**
CO6	3	3	2	2	3	3	3	3	**2.8**
AVG.	**3.0**	**2.5**	**2.0**	**1.5**	**3.0**	**3.0**	**2.5**	**2.0**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	3	**2.7**
CO2	2	3	3	**2.7**
CO3	2	3	3	**2.7**
CO4	2	3	3	**2.7**
CO5	2	3	3	**2.7**
CO6	2	3	3	**2.7**
AVG.	**2.0**	**3.0**	**3.0**	**2.7**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E603	Computer Graphics	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Knowledge of C++, calculus, linear algebra, integra, vectors, matrices, basis, solving systems of equations.

Course Objectives:

- To understand the basics of various inputs and output computer graphics hardware devices.
- To know 2D raster graphics techniques, 3D modelling, geometric transformations, 3D viewing and rendering.

Course Outcomes:

CO1	Understand the basics of computer graphics, different graphics systems and applications of computer graphics.
CO2	Discuss various algorithms for scan conversion and filling of basic objects and their comparative analysis.
CO3	Explore projections and visible surface detection techniques for display of 3D scene on 2D screen.
CO4	Render projected objects to naturalize the scene in 2D view and use of illumination models for this.
CO5	Understand the basics of computer graphics, different graphics systems and applications of computer graphics.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to Computer Graphics**: Video display devices, Raster- scan systems, Random- scan systems, Graphics monitors and workstations, Input devices, hardcopy devices, Computer graphics software. Point plotting techniques: Points and lines, line- drawing algorithm, Circle generating algorithms, Ellipse- generating algorithms.	8
2.	**Two-Dimensional Transformations**: Transformations of Points, Transformations of Straight Lines, Mid-Point Transformations, Transformations of Parallel Lines, Transformations of Intersecting Lines, Rotation, Reflection, Scaling, Projection, Combined Transformation, Transformation of the unit square.	8
3.	**Three-Dimensional Transformation**: Introduction to Three- Dimensional Transformation, 3-D Scaling, Shearing, Rotation, Reflection, Projection and Translation, Multiple Transformation, Rotation about an Axis parallel to a Coordinate Axis, rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane.	8
4.	**Two-Dimensional Viewing**: viewing pipeline, viewing coordinate reference frame, window- to- viewport coordinate transformation, Clipping operations, point clipping, Cohen- Sutherland line clipping, Sutherland- Hodgeman polygon clipping, Curve clipping, Text clipping, Exterior clipping.	8
5.	**Visible Surface Detection Methods**: Classification, back-face detection, depth-buffer, scan-line, depth sorting, BSP tree methods, area sub-division and	8

	octree methods.	
6.	**Computer Animation**: Design of animation sequences, General computer-animation functions, Raster animations, Computer- animation languages, Key-frame systems morphing simulating accelerations, Motion specifications.	8
	Total Lectures	48

Suggested Text Book(s):

1.	Hearn D., Baker P.M., Computer Graphics, Prentice-Hall of India.
2.	Rogers and Adams, Mathematical Elements of Computer Graphics, McGraw Hill Book Co.

Suggested Reference Book(s):

1.	Newman, W., Sproul, R.F., Principles of Interactive Computer Graphics, McGraw- Hill.
2.	John F. Hughes et. al., Computer Graphics: Principles and Practice, Addison-Wesley Professional.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106106090
2.	https://nptel.ac.in/courses/106102065
3.	https://ocw.mit.edu/courses/6-837-computer-graphics-fall-2012/pages/lecture-notes/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	1	1	1	2	1	1	**1.1**
CO2	2	2	2	2	1	2	2	1	**1.8**
CO3	3	2	2	2	2	3	2	2	**2.3**
CO4	3	2	2	2	2	3	2	2	**2.3**
CO5	2	2	2	2	2	3	2	3	**2.3**
AVG.	**2.2**	**1.8**	**1.8**	**1.8**	**1.6**	**2.6**	**1.8**	**1.8**	**1.9**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	1	-	1	**0.7**
CO2	3	2	3	**2.7**
CO3	2	2	3	**2.3**
CO4	2	2	3	**2.3**
CO5	3	2	3	**2.7**
AVG.	**2.2**	**1.6**	**2.6**	**1.9**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E653	Computer Graphics Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Basic knowledge of C/C++ programming.

Course Objectives:

- To make student able to implement the computer graphics algorithm and basic animation using ‘C’

Course Outcomes:

CO1	Understand the basic concepts of computer graphics.
CO2	Design scan conversion problems using C++ programming.
CO3	Apply clipping and filling techniques for modifying an object.
CO4	Understand the concepts of different type of geometric transformation of objects in 2D and 3D.
CO5	Understand the practical implementation of modeling, rendering, viewing of objects in 2D.

Lab Exercises

Write Programs in C/C++ Language:

1. To plot a point (pixel) on the screen.
2. To draw a straight line using DDA Algorithm.
3. To draw a straight line using Bresenham’s Algorithm.
4. Implementation of mid-point circle generating Algorithm.
5. Implementation of ellipse generating Algorithm.
6. To translate an object with translation parameters in X and Y directions.
7. To scale an object with scaling factors along X and Y directions.
8. To rotate an object with a certain angle about origin.
9. Perform the rotation of an object with certain angle about an arbitrary point.
10. To perform composite transformations of an object.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	2	3	1	2	2	2	2	**2.0**
CO2	3	3	2	2	3	2	2	2	**2.4**
CO3	3	3	1	2	3	2	2	3	**2.4**
CO4	3	3	1	2	3	2	2	3	**2.4**
CO5	3	3	1	2	3	2	3	3	**2.5**
AVG.	**2.8**	**2.8**	**1.6**	**1.8**	**2.8**	**2.0**	**2.2**	**2.6**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	3	3	3	**3.0**
CO3	2	2	2	**2.0**
CO4	2	2	2	**2.0**
CO5	2	3	3	**2.7**
AVG.	**2.4**	**2.6**	**2.4**	**2.6**

SUBJECT: COMPUTER SCIENCE

SEC-4	BCS-S604	Java Programming	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: Knowledge of Object-Oriented Concepts through any programming language like C++.

Course Objectives:

- To understand the basic concepts and fundamentals of platform independent object-oriented language.
- To demonstrate skills in writing programs using exception handling techniques and multithreading.
- To understand streams and efficient user interface design techniques.

Course Outcomes:

CO1	Familiarize with the concept of Object-Oriented concepts by implementing Java Programming.
CO2	Learn the concepts of classes & objects with the features of reusability and implementation of the same with various control structures to solve real world problems.
CO3	Understand and design built-in and user defined functions/methods, interfaces and packages etc.
CO4	Handle various types of data using arrays & strings and handling of exceptions occurred in programs.
CO5	Utilize multithreading and applet features of Java for efficient and effective programming.
CO6	Create and handle files in Java.

Course Contents

UNIT	Contents	Lectures Required
1.	**Java Programming Fundamentals:** Introduction to Java, Stage for Java, Origin, Challenges of Java, Java Features, Java Program Development, Object Oriented Programming. Elements of Java Program, Java API, Variables and Literals, Primitive Data Types, The String class, Variables, Constants, Operators, Scope of Variables & Blocks, Types of Comment in Java. **Control Statements:** Decision making statements (if, if-else, nested if, else if ladder, switch, conditional operator), Looping statements (while, do-while, for, nested loops), Jumping statements (Break and Continue). **Classes and Objects:** Basic concepts of OOPS, Classes and Objects, Modifiers, Passing arguments, Constructors, Overloaded Constructors, Overloaded Operators, Static Class Members, Garbage Collection. **Inheritance:** Basics of inheritance, Inheriting and Overriding Superclass methods, Calling Superclass Constructor, Polymorphism, Abstract Classes, Final Class.	8
2.	**Arrays and Strings:** Introduction to array, Processing Array Contents, Passing array as argument, Returning array from methods, Array of objects,	8

	2D arrays, Array with three or more dimensions. String class, string concatenation, Comparing strings, Substring, Difference between String and String Buffer class, String Tokenizer class. **Interface and Packages:** Basics of interface, Multiple Interfaces, Multiple Inheritance Using Interface, Multilevel Interface, Packages, Create and Access Packages, Static Import and Package Class, Access Specifiers. **Exception Handling:** Introduction, Try and Catch Blocks, Multiple Catch, Nested Try, Finally, Throw Statement, Built-In Exceptions.	
3.	**Multithreading:** Introduction, Threads in Java, Thread Creation, Lifecycle of Thread, Joining a Thread, Thread Scheduler, Thread Priority, Thread Synchronization. **Applets:** Introduction, Applet Class, Applet Life Cycle, Graphics in Applet, Event-Handling.	8
	Total Lectures	24

Suggested Text Book(s):

1.	E. Balagurusamy, Programming with Java A Primer, 5th Edition, TMH.
2.	Sagayaraja, Denis, Karthik, Gajalakshmi, Java Programming for Core and Advanced Learners, Universities Press.

Suggested Reference Book(s):

1.	H. Schildt , Java, The complete Reference, TMH.
2.	H. Schildt, D. Skrien, Java Fundamentals, A Comprehensive Introduction, TMH.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/106105191
2.	https://archive.nptel.ac.in/courses/106/105/106105191/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	3	3	2	1	2.4
CO2	3	3	3	2	3	3	3	3	2.9
CO3	3	3	3	1	2	2	3	2	2.4
CO4	3	3	1	1	3	2	2	2	2.1
CO5	2	3	2	2	2	2	2	3	2.3
CO6	2	3	1	1	1	2	2	2	1.8
AVG.	2.7	3.0	2.2	1.3	2.3	2.3	2.3	2.2	2.3

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	2	3	3	**2.7**
CO3	2	3	3	**2.7**
CO4	2	3	3	**2.7**
CO5	3	3	3	**3.0**
CO6	2	3	2	**2.3**
AVG.	**2.3**	**3.0**	**2.7**	**2.7**

SUBJECT: COMPUTER SCIENCE							
SEC-4	BCS-S654	Java Programming Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Any object- oriented programming language

Course Objectives:

- To gain knowledge about basic Java language syntax and semantics to write Java programs.

Course Outcomes:

CO1	Understand the use of oops concepts and Solve real world problems using OOP technique
CO2	Understand the use of Packages and Interface in java.
CO3	Develop and understand exception handling, multithreaded applications with synchronization.
CO4	Design GUI based applications and develop applets for web applications.
CO5	Understand the use of Collection Framework.

Lab Exercises

1. Write a program to perform following operations on two numbers input by the user:
   1) Addition 2) subtraction 3) multiplication 4) division
2. Write a Java program to compute area of: 1) Circle 2) rectangle 3) triangle 4) square
3. Write a program through Java that reads a number in inches, converts it to meters.
4. Write a program to convert minutes into a number of years and days.
5. Write a Java program that prints current time in GMT.
6. Write a program to sum values of an Single Dimensional array.
7. Design & execute a program in Java to sort a numeric array and a string array.
8. Calculate the average value of array elements through Java Program.
9. Write a Java program to test if an array contains a specific value.
10. Find the index of an array element by writing a program in Java.
11. Design a program to copy an array by iterating the array.
12. Write a Java program to insert an element (on a specific position) into Multidimensional array.
13. Write a program to perform following operations on strings:
    - Compare two strings.
    - Concatenate two strings.
    - Print a substring.
14. Write a Java method to count all words in a string.
15. Write a method in Java program to count all words in a string.

16. Write a Java program to handle following exceptions:
    - Divide by Zero Exception.
    - Array Index Out of bound Exception.
17. To represent the concept of Multithreading write a Java program.
18. To represent the concept of all types of inheritance supported by Java, design a program.
19. Write a program to implement Multiple Inheritance using interface.
20. Construct a program to design a package in Java.
21. To write and read a plain text file, write a Java program.
22. Write a Java program to append text to an existing file.
23. Design a program in Java to get a list of all file/directory names from the given.
24. Develop a Java program to check if a file or directory specified by pathname exists or not.
25. Write a Java program to check if a file or directory has read and write permission.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	3	3	3	2	3	**2.9**
CO2	3	2	3	2	2	3	1	2	**2.3**
CO3	3	3	2	3	2	3	3	2	**2.6**
CO4	3	3	1	2	1	3	3	3	**2.4**
CO5	3	2	3	1	1	3	2	2	**2.1**
AVG.	**3.0**	**2.6**	**2.4**	**2.2**	**1.8**	**3.0**	**2.2**	**2.4**	**2.5**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO→	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	3	3	2	**2.7**
CO3	2	3	3	**2.7**
CO4	3	3	3	**3.0**
CO5	3	2	3	**2.7**
AVG.	**2.8**	**2.8**	**2.6**	**2.7**

SUBJECT: COMPUTER SCIENCE

SEC-4	BCS-S605	Internet Technologies	L	T	P	C	Time for ESE
			2	-	-	2	3 Hrs.

Pre- requisite: NIL

Course Objectives:

- This Subject is useful for Making own Web page and how to host own web site on internet. Also, Students will learn what the protocols are involving in internet technology.

Course Outcomes:

CO1	Describe the evolution of the Internet.
CO2	Understand the protocols and standards used throughout the Internet.
CO3	Discuss a variety of Internet and WWW applications and related technologies.
CO4	Evaluate the opportunities and threats created by interconnecting computers via the Internet.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to Internet Technology:** Introduction to Internet, History of Internet, Internet Service Provider, Client/Server Architecture, Domain Name System, Web Server. **Basics of HTML :** Create static webpage using HTML tags; Apply CSS into webpage.	8
2.	**Active Server Pages 3.0:** Introduction to ASP; Benefits of ASP; Advantages of ASP over HTML; Using scripting language; Setting primary scripting language; Including other files; Using virtual keyword and File keyword, Including Files; Transferring data using GET and POST methods; Introduction to IIS.	8
3.	**Server side coding with VBScript and XML:** ASP Objects, Use different objects of ASP; Transfer values from one ASP web form to other web form using methods of objects of ASP; Use methods and properties of application and server objects of ASP; Manage session using session objects properties and methods; Apply Adrotator and Browser capability components in ASP web page.	8
	Total Lectures	24

Suggested Text Book(s):

1.	A. Russell Jones, Mastering Active Server Pages 3, BPB Publication
2.	Ivan Bayross, Practical ASP, BPB Publication.
3.	Web Enabled commercial application development using HTML, DHTML, JavaScript, Perl, CGI.

4.	Steven Holzner,HTML Black Book Dremtech press.

Suggested Reference Book(s):

1.	Web Technologies, Black Book, dreamtech Press.
2.	Knuckles,Web Applications : Concepts and Real World Design, Wiley-India
3.	P.J. Deitel & H.M. Deitel, Internet and World Wide Web How to program,Pearson.

Other Useful Resource(s)

1.	www.w3schools.com/asp/
2.	www.webwiz.co.uk
3.	www.w3schools.com/html/
4.	www.csstutorial.net/

Course Outcomes Contributed to Programme Outcomes

PO→ CO→	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	1	3	1	1	1	1	1	**1.3**
CO2	2	2	3	1	2	3	1	2	**2.0**
CO3	2	3	2	2	2	3	2	2	**2.3**
CO4	1	3	1	2	3	3	3	3	**2.4**
AVG.	**1.5**	**2.3**	**2.3**	**1.5**	**2.0**	**2.5**	**1.8**	**2.0**	**2.0**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	1	1	**1.7**
CO2	3	2	2	**2.3**
CO3	2	3	3	**2.7**
CO4	1	3	3	**2.3**
AVG.	**2.3**	**2.3**	**2.3**	**2.2**

SUBJECT: COMPUTER SCIENCE

SEC-4	BCS-S655	Internet Technologies Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of Web technologies and Notepad++

Course Objectives:

- Understand the web technologies to create adaptive web pages for web application.
- Use CSS to implement a variety of presentation effects to the web application.

Course Outcomes:

CO1	Install and setting of Web container/Web Server/Tomcat
CO2	Inter servlet communication
CO3	Use of JDBC
CO4	Understand the two tier and three tier applications using internet programming languages like Java
CO5	Use of JSP objects, DHTML and PHP to develop projects.

Lab Exercises

1. Create webpage using text formatting tags of HTML.
2. Create webpage using table tags and list tags of HTML.
3. Create webpage using HTML Hyper linking
4. Create webpage to include image using HTML tag.
5. Create employee registration webpage using HTML form objects.
6. Apply style sheet in Web page.
7. Create web page in which XML tags are used.
8. Create web page to display simple text message using VBScript in ASP.
9. Create web page to generate grade sheet of student using VBScript in ASP.
10. Create web page to demonstrate use of different ASP objects.
11. Create webpage to transfer data filled through various HTML form controls and collection of the same in ASP.
12. Create webpage to Send text with response object and embedded quotes in ASP.
13. Create webpage to Send text using AddHeader method of Response object in ASP.
14. Create webpage to Send text using Request method of Response object in ASP.
15. Create webpage to transfer data using Request. Cookie collection of in ASP.
16. Create webpage to transfer data using Request. Query String collection of in ASP.
17. Create webpage for Student Registration and validate data using Request. Form collection in ASP.

18. Create webpage to demonstrate use of Browser Capability and AdRotator components in ASP.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	3	3	3	1	3	**2.8**
CO2	3	2	2	1	2	2	2	1	**1.9**
CO3	3	3	1	1	2	2	2	2	**2.0**
CO4	3	3	1	2	1	3	2	3	**2.3**
CO5	3	3	3	3	3	3	2	3	**2.9**
AVG.	**3.0**	**2.8**	**2.0**	**2.0**	**2.2**	**2.6**	**1.8**	**2.4**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	2	3	**2.7**
CO2	2	2	2	**2.0**
CO3	3	3	3	**3.0**
CO4	2	3	3	**2.7**
CO5	3	3	3	**3.0**
AVG.	**2.6**	**2.6**	**2.8**	**2.6**

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C701	Software Engineering	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Discrete Mathematics, Algorithm.

Course Objectives:

- To understand the Software Engineering Practices and Process Models.

Course Outcomes:

CO1	Assessment in each module gives the overall Software engineering practice.
CO2	Ability to enhance the software project management skills.
CO3	Ability to design and develop a software product in accordance with Software Engineering principles.

Course Contents

UNIT	Contents	Lectures Required
1.	**Software Process**: Software Process, Characteristics, software development process models - Waterfall, Iterative, Prototype, Incremental, Spiral, win-win Spiral, Comparison. Project Management Process.	7
2.	**Software Requirement Analysis and specification**: Software Requirements, need for SRS, Problem analysis, Requirements specification, IEEE format of SRS, Requirements Engineering, Requirements Validation, Object-oriented Analysis Case Studies - Course Scheduling, Personal Investment Management System.	7
3.	**Software Architecture**: Role of Software Architecture, Architecture views, Component and Connector view. Architectural styles of C&C view. Evaluating Architectures.	7
4.	**Software Design**: Function Oriented Design: Principles, Module-level Concepts. Design notations and specifications, Structured design methodology, Verification, Metrics; Object-oriented design: OO Concepts, Design Concepts, Unified Modeling Language (UML); User Interface Design: Golden rules, User Interface Design, Interface Design Activities, Implementation tools	10
5.	**Testing Techniques & Strategies**: Fundamentals, Test case design, white box, black box, basis path, control structure testing, Strategic approach to software testing, Unit testing, Integration testing, Validation testing & System Testing.	7
6.	**Software Maintenance**: Definition, Maintenance activities, Software Reengineering, Reverse Engineering, Restructuring, Forward Engineering.	4
7.	**Effort & Schedule Estimation**: Software Project Estimation, Decomposition techniques, Empirical Estimation Models (COCOMO, Function Point Analysis, Delphi Approach), The Make/Buy decision. Automated Estimation tools.	6
	Total Lectures	48

Suggested Text Book(s):

1.	Agarwal, KK, et. al., Software Engineering, New Age International Publication
2.	Jalote Pankaj, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi
3.	Pressmann, RS, Software Engineering – A Practitioner's Approach, McGraw- Hill International Editions.

Suggested Reference Book(s):

1.	Sommerville, Ian, Software Engineering, Pearson Education Asia.
2.	Bruegge and Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, Pearson Education Asia.

Other Useful Resource(s)

1.	https://onlinecourses.nptel.ac.in/noc20_cs68/preview
2.	https://www.youtube.com/watch?v=Z6f9ckEElsU

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	3	3	3	1	2	1	2	**2.1**
CO2	2	3	2	3	2	1	3	3	**2.4**
CO3	2	3	3	3	2	3	3	3	**2.8**
AVG.	**2.0**	**3.0**	**2.7**	**3.0**	**1.7**	**2.0**	**2.3**	**2.7**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	3	3	2	**2.7**
CO3	3	3	3	**3.0**
AVG.	**3.0**	**3.0**	**2.3**	**2.8**

SUBJECT: COMPUTER SCIENCE							
DCS	BCS-C751	Software Engineering Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of programming language.

Course Objectives:

- Learn about software myths, generic view of the process and Understand about process models.
- Learn how to perform feasibility study of the projects under the requirement engineering process and system models.
- Understand about Function oriented design and Architectural styles..

Course Outcomes:

CO1	Prepare SRS document, design document, test cases and software configuration management and risk management related document.
CO2	Develop function oriented and object-oriented software design using tools like rational rose.
CO3	Develop a working protocol.
CO4	Perform unit testing and integration testing. Apply various white box and black box testing techniques.

Lab Exercises

1. Identify project scope and objective of given problem:
   a) College automation system.
   b) Banking Management System.
2. Develop software requirements specification for (1 a.) and (1 b.) problem.
3. Develop UML Use case model for a problem.
4. Develop Class diagrams
5. Represent project Scheduling of above-mentioned projects
6. Use any model for estimating the effort, schedule and cost of software project
7. Develop DFD model (level--, level-1 DFD and Data dictionary) of the project
8. Develop sequence diagram
9. Develop Structured design for the DFD model developed
10. Develop the waterfall model, prototype model and spiral model of the product
11. Explain with reason which model is best suited for the product
12. Develop a working protocol of any of two problem.
13. Use LOC, FP and Cyclomatic Complexity Metric of above-mentioned problem
14. Find Maintainability Index and Reusability Index of above-mentioned problem
15. Using any Case Tool find number of statements, depth and complexity of the prototype

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	3	3	3	1	3	3	1	**2.4**
CO2	2	3	2	2	2	2	2	3	**2.3**
CO3	2	2	2	2	2	2	2	2	**2.0**
CO4	2	2	2	2	2	3	3	3	**2.4**
AVG.	**2.0**	**2.5**	**2.3**	**2.3**	**1.8**	**2.5**	**2.5**	**2.3**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	3	**3.0**
CO2	2	3	3	**2.7**
CO3	1	3	1	**1.7**
CO4	1	3	2	**2.0**
AVG.	**1.8**	**3.0**	**2.3**	**2.4**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-C702	Python Programming	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Basic understanding of computer programming.

Course Objectives:

- To understand why Python is a useful scripting language for developers.
- To learn how to design and program Python applications.
- To learn how to use lists, tuples, and dictionaries in Python programs.
- To learn how to identify Python object types.
- To define the structure and components of a Python program.
- To learn how to write loops and decision statements in Python.

Course Outcomes:

CO1	Able to apply the principles of python programming.
CO2	Create applications using python programming.
CO3	Implementing database using SQLite.
CO4	Access database using python programming.
CO5	Develop web applications using python programming.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction to Python:** History, Features, setting up path, working with Python, Basic Syntax, Variable and Data Types, Operator. **Conditional Statements & Looping:** If, If- else, Nested if-else, For, While, Nested loops, Break, Continue, Pass. **String Manipulation:** Accessing Strings, Basic Operations, String slices, Function and Methods, Formatting strings.	8
2.	**Lists, Tuple and Dictionaries:** Lists – Introduction, accessing list, Operations, working with lists, Function and Methods, Tuple – Introduction, accessing tuples, Operations, Working, Functions and Methods, Dictionaries - Introduction, accessing values in dictionaries, working with dictionaries, Properties, Functions. **Functions:** Defining a function, calling a function, Types of functions, Function Arguments, Anonymous functions, Function documentation, Keyword and optional parameters, *args and **kwargs, passing collection to a function, variable number of arguments, scope, functions – "First Class Citizens", Passing functions to function, mapping functions in a dictionary, Global and local variables.	10
3.	**Modules:** Importing module, Math module, Random module, Packages, Composition, dir function **Input-Output:** Printing on screen, reading data from keyboard, Opening and closing file, Reading and writing files, Working with Directories, Metadata. **Object and Classes:** Classes in Python, Principles of Object Orientation, Creating Classes, Instance Methods, File Organization, Special Methods, Class	10

	Variables, Inheritance, Polymorphism, Type Identification, Custom Exception Classes.	
4.	**Error Handling:** Handling IO Exceptions, Errors, Run Time Errors, The Exception Model, Exception Hierarchy, Handling Multiple Exceptions. **Regular expressions:** Match function, Search function, Matching VS Searching, Modifiers Patterns.	10
5.	**CGI:** Introduction, Architecture, CGI environment variable, GET and POST methods, Cookies, File upload. **Database:** Introduction, Connections, Executing queries, Transactions, Handling error.	10
	Total Lectures	48

Suggested Text Book(s):

1.	Gowrishankar S, Veena A, Introduction to Python Programming, CRC Press
2.	Mark Lutz, Learning Python, O'Reilly Media

Suggested Reference Book(s):

1.	Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning.
2.	Chun, Wesley. Core python programming. Vol. 1. Prentice Hall Professional.

Other Useful Resource(s)

1.	https://onlinecourses.nptel.ac.in/noc18_cs35/preview
2.	https://nptel.ac.in/courses/106106145/
3.	https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/index.htm
4.	https://docs.python.org/3/tutorial/index.html

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	1	3	1	1	**2.0**
CO2	3	3	1	2	3	2	2	3	**2.4**
CO3	3	3	1	2	2	2	2	2	**2.1**
CO4	3	3	2	2	1	3	2	2	**2.3**
CO5	2	3	1	3	3	3	2	3	**2.5**
AVG.	**2.8**	**3.0**	**1.6**	**2.0**	**2.0**	**2.6**	**1.8**	**2.2**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	2	3	3	**2.7**
CO3	2	3	2	**2.3**
CO4	2	3	2	**2.3**
CO5	2	3	3	**2.7**
AVG.	**2.2**	**3.0**	**2.4**	**2.6**

SUBJECT: COMPUTER SCIENCE

DCS	BCS-C752	Python Programming Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Understanding of programming language like C/C++.

Course Objectives:

- To interpret the use of procedural statements like assignments, conditional statements, loops and function calls.
- To infer the supported data structures like lists, dictionaries and tuples in Python.
- To illustrate the application of matrices and regular expressions in building the Python programs.
- To discover the use of external modules in creating excel files and navigating the file systems.
- To describe the need for Object-oriented programming concepts in Python.

Course Outcomes:

CO1	Describe the Python language syntax including control statements, loops and functions to write programs for a wide variety problem in mathematics, science, and games.
CO2	Examine the core data structures like lists, dictionaries, tuples and sets in Python to store, process and sort the data.
CO3	Interpret the concepts of Object-oriented programming as used in Python using encapsulation, polymorphism and inheritance.
CO4	Discover the capabilities of Python regular expression for data verification and utilize matrices for building performance efficient Python programs.
CO5	Identify the external modules for creating and writing data to excel files and inspect the file operations to navigate the file systems.

Lab Exercises

1. Write a program to demonstrate basic data type in python.
2. Write a Program for checking whether the given number is an even number ornot.
3. Using a for loop, write a program that prints out the decimal equivalents of
4. 1/2, 1/3, 1/4, . . ., 1/10
5. Write a program to find the union of two lists.
6. Write a program to find the sum of all the primes below two million.
7. By considering the terms in the Fibonacci sequence whose values do not exceed four million, write a program to find the sum of the even-valued terms.
8. Write a program to count the numbers of characters in the string and storethem in a dictionary data structure
9. Write a program to use split and join methods in
10. Write a Python program to find the intersection of two lists.
11. Write a Python program to remove the “i” th occurrence of the given word in a list where words repeat
12. Write a Python program to count the occurrences of each word in a given string sentence.
13. Write a Python program to check if a substring is present in a given string.

14. Write a Python program to map two lists into a dictionary.

15. Write a Python program to count the frequency of words appearing in a string using a dictionary.

16. Write a Python program to create a dictionary with key as first character and value as words starting with that character.

17. Write a Python program to find the length of a list using recursion.

18. Write a Python program to plot the Line chart in MS Excel Sheet using XlsxWriter module to display the annual net income of the companies mentioned below.

19. Write a Python program to read a file and capitalize the first letter of every word in the file.

20. Write a Python program to read the contents of a file in reverse order.

21. Write a Python program to create a class in which one method accepts a string from the user and prints it in reverse.

22. Using Regular Expressions, develop a Python program to
a) Identify a word with a sequence of one upper case letter followed by lower case letters.
b) Find all the patterns of “1(-+)1” in a given string.
c) Match a word containing ‘z’ followed by one or more o’s.
Prompt the user for input.

23. Study and Implementation of Database, Structured Query Language and database connectivity.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	3	2	1	1	3	3	3	**2.3**
CO2	2	3	2	2	3	3	3	3	**2.6**
CO3	2	3	2	2	2	3	3	3	**2.5**
CO4	2	3	2	2	1	3	3	3	**2.4**
CO5	2	3	2	3	3	3	3	3	**2.8**
AVG.	**2.0**	**3.0**	**2.0**	**2.0**	**2.0**	**3.0**	**3.0**	**3.0**	**2.5**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	2	3	3	**2.7**
CO3	2	3	2	**2.3**
CO4	2	3	2	**2.3**
CO5	2	3	3	**2.7**
AVG.	**2.2**	**3.0**	**2.4**	**2.6**

SUBJECT: COMPUTER SCIENCE							
DSC	BCS-O703	Online Course	L	T	P	C	Time for ESE
			-	-	-	4	-

This course has to be completed by the students by choosing courses from NPTEL/ SWAYAM/ MOOCs/ etc. They should undergo the online course completely, submit assignments, projects, etc. and appear for the final exam conducted by the online instructor/ NTA/ competent authority. The awarded certificate must be submitted for the award of credits in this course.

This is an online course which student can select from the offered online courses and must be of computer discipline but should not be the core courses offered in this course of study.

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C761	Industrial Training/ Research Project/ Dissertation	L	T	P	C	Time for ESE
			-	-	-	6	3 Hrs.

Pre- requisite: Fundamental knowledge of programming languages and computer science.

Course Objectives:

- To provide to students the feel of the actual working environment and to gain practical knowledge and skills, which in turn will motivate, develop and build their confidence.
- Industrial training is also expected to provide the students the basis to identify their key operational area of interest.

Course Outcomes:

CO1	Participate in the projects in industries during his or her industrial training.
CO2	Describe use of advanced tools and techniques encountered during industrial training and visit.
CO3	Interact with industrial personnel and follow engineering practices and discipline prescribed in industry.
CO4	Develop awareness about general workplace behavior and build interpersonal and team skills.
CO5	Prepare professional work reports and presentations.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	-	2	3	1	2	3	3	**2.3**
CO2	2	2	2	2	1	2	3	3	**2.1**
CO3	2	2	2	2	2	3	3	3	**2.4**
CO4	2	2	2	2	2	3	3	3	**2.4**
CO5	2	2	2	2	2	3	3	3	**2.4**
AVG.	**2.0**	**1.6**	**2.0**	**2.2**	**1.6**	**2.6**	**3.0**	**3.0**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	3	3	3	**3.0**
CO3	2	3	3	**2.7**
CO4	2	2	3	**2.3**
CO5	3	3	3	**3.0**
AVG.	**2.6**	**2.8**	**2.8**	**2.8**

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C801	Computer Networks	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Basic knowledge of computer network thermology and programming language.

Course Objectives:

- To develop an understanding of modern network architectures from a design and performance perspective.
- To introduce the student to the major concepts involved in wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs).

Course Outcomes:

CO1	Explain the functions of the different layer of the OSI Protocol.
CO2	Draw the functional block diagram of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) describe the function of each block.
CO3	For a given requirement (small scale) of wide-area networks (WANs), local area networks (LANs) and Wireless LANs (WLANs) design it based on the market available component
CO4	Configure DNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls using open source available software and tools.

Course Contents

UNIT	Contents	Lectures Required
1.	**Data communication Components**: Representation of data and its flow Networks, Various Connection Topologies, Protocols and Standards, OSI model, Transmission Media. **LAN**: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN. **Techniques for Bandwidth utilization**: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.	10
2.	**Data Link Layer and Medium Access Sub Layer**: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD, CDMA/CA.	10
3.	**Network Layer**: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols.	10
4.	**Transport Layer**: Process-to-Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.	10
5.	**Application Layer:** Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.	8
	Total Lectures	48

Suggested Text Book(s):

1.	Behrouz A. Forouzan, Data Communication and Networking, McGraw- Hill
2.	William Stallings, Data and Computer Communication, Pearson

Suggested Reference Book(s):

1.	James F. Kurose, Keith W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, Pearson Education.
2.	Andrew S. Tanenbaum, Computer Networks, PHI.

Other Useful Resource(s)

1.	https://www.coursera.org/specializations/networking-basics
2.	https://nptel.ac.in/courses/106105080/
3.	https://swayam.gov.in/course/4066-computer-networks

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	1	3	1	1	1	1	1	**1.5**
CO2	2	2	3	1	1	2	1	1	**1.6**
CO3	2	3	2	3	2	3	3	3	**2.6**
CO4	1	3	1	2	3	3	3	3	**2.4**
AVG.	**2.0**	**2.3**	**2.3**	**1.8**	**1.8**	**2.3**	**2.0**	**2.0**	**2.0**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	1	1	**1.7**
CO2	3	2	2	**2.3**
CO3	2	2	3	**2.3**
CO4	1	3	3	**2.3**
AVG.	**2.3**	**2.0**	**2.3**	**2.1**

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C851	Computer Networks Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Knowledge of computer networks and programming language C/C++.

Course Objectives:

- To get practical knowledge of working principles of various communication protocols.
- Implementing various network algorithms such as error control, error detection, routing, and security related algorithms.

Course Outcomes:

CO1	Implement go back n sliding window and selective repeat window protocols
CO2	Stimulate Cyclic Redundancy Check error detection algorithm for noisy channel.
CO3	Implement stop and wait, protocols for noisy channel.
CO4	Implement distance vector and Dijkstra algorithms.

Lab Exercises

1. Representation of a computer network using matrix representation of a graph
2. Finding shortest path between any two nodes in a computer network using Dikjstra‟s shortest path algorithm
3. Finding shortest path between any two nodes in a computer network using Prim‟s shortest path algorithm
4. Study of network troubleshooting using Ping and Traceroute commands
5. Study of various networking and inter – networking devices
6. Implementation of CRC generator and checker algorithm in C / C++ / Java
7. Implementation of Hamming code algorithm in C / C++ / Java
8. Study of client – server programming using sockets in a UNIX / Linux and Windows environment
9. Implementing client – server program using TCP / UDP sockets
10. Implementation of Stop – and – Wait protocol in C / C++ / Java in a client – server environment using sockets
11. Implementation of Sliding Window protocol in C / C++ / Java in a client – server environment using sockets
12. Implementation of encryption algorithm converting plain text to cipher text using C / C++ / Java
13. Design and implement Traffic Shaping Algorithms:
    a. Leaky Bucket
    b. Token Bucket
14. Implementation of chat system

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	2	2	2	2	2	2	3	**2.0**
CO2	3	3	2	2	3	3	1	3	**2.5**
CO3	1	2	2	2	2	2	2	3	**2.0**
CO4	3	3	2	2	2	2	2	3	**2.4**
AVG.	**2.0**	**2.5**	**2.0**	**2.0**	**2.3**	**2.3**	**1.8**	**3.0**	**2.2**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	3	2	**2.3**
CO2	2	3	2	**2.3**
CO3	2	3	2	**2.3**
CO4	2	3	2	**2.3**
AVG.	**2.0**	**3.0**	**2.0**	**2.3**

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C802	Computer System Architecture	L	T	P	C	Time for ESE
			5	1	-	6	3 Hrs.

Pre- requisite: Basic Understanding of Computer System

Course Objectives:

- To understand the basic hardware and software issues of computer organization.
- To provide an overview on the design principles of digital computing systems.
- To understand the representation of data at machine level.

Course Outcomes:

CO1	Understand the basics of digital electronics and elaborate basic computer organization, control unit and central processing unit.
CO2	To understand the fixed-point and floating-point numbers are represented in a computer.
CO3	Wide understanding of memory organization and management in a modern digital computer, including virtual and physical memory, address translation, multilevel, unified, and multi-way set-associative caches, the translation-look-aside buffer (TLB), and the page table.
CO4	To understand the working strategies of parallel processing and multi-core computers.
CO5	Discuss about pipelining in a processor functions and describe how hazards are resolved in various ways.

Course Contents

UNIT	Contents	Lectures Required
1.	**Digital Electronics**: Boolean algebra and logic Gates, Simplification of Boolean Functions, Adders, subtractors, Binary parallel adder, Decimal adder, Magnitude comparator, Decoders, Multiplexers. Flip- flops (RS, D, JK, Master-slave & T flip- flops), Flip- flop Excitation table, analysis, Design of counters, Registers, Shift register, Ripple Counter, Synchronous Counters, Timing sequences.	16
2.	**Central Processing Unit**: Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt, Bus Interconnection design of basic computer, register organization; Stack organization; Instruction Format and Addressing Modes. **Control Unit**: Control memory, Address Sequencing, Micro program, Design of Control Unit.	12
3.	**Arithmetic Algorithms**: Integer multiplication; Integer division, Floating point representations and Arithmetic algorithms. **I/O Organization**: Peripheral Devices, Input-Output Interface, Asynchronous Data Transfer, Modes of Data Transfer, Priority Interrupt, Direct Memory Access, Input Output Processor.	12
4.	**Memory Organization**: Memory Hierarchy, RAM, ROM, Associative Memory, Cache Memory Organization and Virtual Memory Organization.	10

5.	**Instruction level Parallelism:** Overviews, design issues, vector processing. **Parallel Processing:** Multiple Processor Organization, Symmetric Multiprocessors, cache coherence, multithreading, clusters, non- uniform memory access, and vector computation.	10
	Total Lectures	60

Suggested Text Book(s):

1.	M. Morris Mano, Computer System Architecture, Pearson.
2.	William Stallings, Computer Organization & Architecture - Designing for Performance Eighth Edition, Pearson.

Suggested Reference Book(s):

1.	John L. Hennessy and David A. Patterson, Computer Architecture: A Quantitative Approach, Fourth Edition, Morgan Kaufmann Publishers.
2.	Hamacher Cart, Vranesic Zvono, Zaky Safwat, Computer Organization, McGraw Hill.

Other Useful Resource(s)

1.	https://nptel.ac.in/syllabus/106103068/
2.	https://www.geeksforgeeks.org/computer-organization-and-architecture- tutorials/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	3	1	1	1	**2.0**
CO2	3	2	1	2	3	1	2	1	**1.9**
CO3	3	3	3	2	3	3	3	3	**2.9**
CO4	3	3	3	2	2	3	3	3	**2.8**
CO5	2	1	2	1	1	2	3	2	**1.8**
AVG.	**2.8**	**2.4**	**2.4**	**1.6**	**2.4**	**2.0**	**2.4**	**2.0**	**2.3**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	1	**2.3**
CO2	3	1	2	**2.0**
CO3	3	1	3	**1.0**
CO4	2	3	3	**2.7**
CO5	2	1	3	**2.0**
AVG.	**2.6**	**1.8**	**2.4**	**1.8**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E803	Machine Learning	L	T	P	C	Time for ESE
			4	-	-	4	3 Hrs.

Pre- requisite: Computer programming (python); Calculus; Linear Algebra.

Course Objectives: This course provides an advanced level of understanding to machine learning and statistical pattern recognition. It offers some of the most cost-effective approaches to automated knowledge acquisition in emerging data-rich disciplines and focuses on the theoretical understanding of these methods, as well as their computational implications.

Course Outcomes:

CO1	Demonstrate in-depth knowledge of methods and theories in the field of machine learning. To introduce the basic principles, techniques, and applications of Machine Learning, Classification Tasks, Decision tree learning
CO2	Understand and use Bayesian perspective on machine learning, Artificial neural networks, back propagation algorithm
CO3	Assess learning algorithms modelled after biological evolution, including Genetic Algorithm
CO4	Demonstrate knowledge of the disciplinary foundation and of proven experience in the design and analysis of learning algorithms and systems.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** Well-Posed learning problems, Basic concepts, Designing a learning system, Issues in machine learning. Types of machine learning: Learning associations, Supervised learning, Unsupervised learning, Reinforcement learning.	10
2.	**Decision Tree Learning:** Decision tree representation, appropriate problems for decision tree learning, Univariate Trees (Classification and Regression), Multivariate Trees, Basic Decision Tree Learning algorithms, Hypothesis space search in decision tree learning, Inductive bias in decision tree learning, Issues in decision tree learning.	10
3.	**Bayesian Learning:** Bayes theorem and concept learning, Bayes optimal classifier, Gibbs algorithms, Naive Bayes Classifier, Bayesian belief networks, The EM algorithm. **Artificial Neural Network:** Neural network representation, Neural Networks as a paradigm for parallel processing, Linear discrimination, pairwise separation, Gradient Descent, Logistic discrimination, Perceptron, Training a perceptron, Multilayer perceptron, Back propagation Algorithm. Recurrent Networks, dynamically modifying network structure.	10
4.	**Genetic Algorithms:** Basic concepts, Hypothesis space search, Genetic programming, Models of evolution and learning, Parallelizing Genetic Algorithms.	8

5.	**Data Mining Techniques for Analysis:** Classification: Decision tree induction, Bayes classification, Rule-based classification, Support Vector Machines, Classification Using Frequent Patterns, k-Nearest-Neighbor, Fuzzy-set approach Classifier, Clustering: K-Means, k-Medoids, Agglomerative versus Divisive Hierarchical Clustering Distance Measures in Algorithmic Methods, Mean-shift Clustering.	10
	Total Lectures	48

Suggested Text Book(s):

1. Mitchell T.M., Machine Learning, McGraw Hill.
2. Bishop C., Pattern Recognition and Machine Learning, Springer-Verlag.

Suggested Reference Book(s):

1. Stephen Marsland, Machine Learning: An Algorithmic Perspective, CRC Press.
2. David Barber, Bayesian Reasoning and Machine Learning, Cambridge University Press.

Other Useful Resource(s)

1. https://nptel.ac.in/courses/106106139
2. https://nptel.ac.in/courses/106105152
3. https://ocw.mit.edu/courses/6-867-machine-learning-fall-2006/pages/lecture-notes/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	3	2	1	1	**2.1**
CO2	2	2	2	3	2	3	3	2	**2.4**
CO3	3	2	2	3	3	3	2	2	**2.5**
CO4	3	3	2	3	3	2	3	3	**2.8**
AVG.	**2.8**	**2.5**	**2.3**	**2.5**	**2.8**	**2.5**	**2.3**	**2.0**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	2	1	3	**2.0**
CO3	3	3	2	**2.7**
CO4	3	3	3	**3.0**
AVG.	**2.8**	**2.5**	**2.5**	**2.4**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E853	Machine Learning Lab	L	T	P	C	Time for ESE
			-	-	4	2	3 Hrs.

Pre- requisite: Comfortable with variables, linear equations, graphs of functions, histograms, and statistical means; good knowledge of programming (Python).

Course Objectives:

- To understand the basic theory underlying machine learning.
- To be able to apply machine learning algorithms to solve problems of moderate complexity.
- To apply the algorithms to a real-world problem, optimize the models learned and report on the expected accuracy that can be achieved by applying the models.

Course Outcomes:

CO1	Understand the mathematical and statistical prospective of machine learning algorithms through python programming.
CO2	Design and evaluate the unsupervised models through python in built functions.
CO3	Evaluate the machine learning models pre-processed through various feature engineering algorithms by python programming.
CO4	Design and apply various reinforcement algorithms to solve real time complex problems.
CO5	Design and develop the code for recommender system using Natural Language processing
CO6	Understand the basic concepts of deep neural network model and design the same.

Lab Exercises

1. Apply Naive Bayes Classifier on a given dataset and evaluate the performance of classifier model.

2. Apply Simple Linear Regression on a given dataset and evaluate the performance of prediction model obtained.

3. Apply Multiple Linear Regression on a given dataset and evaluate the performance of prediction model obtained.

4. Apply Logistic Regression on a given dataset and evaluate the performance of prediction model obtained.

5. Apply Support Vector Machine classifier (SVM) on a given dataset and evaluate the performance of classifier model obtained.

6. Apply Decision Tree classifier (ID3) on a given dataset and evaluate the performance of classifier model obtained.

7. Build an Artificial Neural Network by implementing the Backpropagation algorithm to classify a given dataset and evaluate the performance of classifier model obtained.

8. Apply Random forest algorithm on a given dataset and compare the classification accuracy with that of Decision Tree classifier (ID3).

9. Apply k-nearest neighbor classifier on a given dataset and evaluate the performance of classifier model obtained.

10. Apply K-means clustering algorithm on a given dataset and evaluate the clusters obtained.

11. Apply Hierarchical clustering algorithm using different linkages on a given dataset and evaluate the clusters obtained.

12. Apply DBSCAN clustering algorithm on a given dataset and evaluate the clusters obtained.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	3	2	1	1	**2.1**
CO2	2	2	2	3	2	3	3	2	**2.4**
CO3	3	2	2	3	3	3	2	2	**2.5**
CO4	3	3	2	3	3	2	3	3	**2.8**
CO5	3	3	2	3	2	3	3	3	**2.8**
CO6	3	2	3	1	1	1	2	3	**2.0**
AVG.	**2.8**	**2.5**	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**2.3**	**2.4**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	**2.7**
CO2	2	1	3	**2.0**
CO3	3	3	2	**2.7**
CO4	3	3	3	**3.0**
CO5	3	3	3	**3.0**
CO6	1	1	3	**1.7**
AVG.	**2.5**	**2.3**	**2.7**	**2.4**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E804	Soft Computing	L	T	P	C	Time for ESE
			5	1	-	6	3 Hrs.

Pre- requisite: A strong mathematical background; proficiency with algorithms and programming skills in C/C++/Java/Python.

Course Objectives:

- To develop the skills to gain a basic understanding of neural network theory, fuzzy logic theory and Genetic Algorithm.
- To introduce students to artificial neural networks, fuzzy theory and Genetic algorithm from an engineering perspective.

Course Outcomes:

CO1	Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
CO2	Apply artificial neural networks and fuzzy logic theory for various problems.
CO3	Determine the use of Genetic algorithm to obtain optimized solutions to problems.

Course Contents

UNIT	Contents	Lectures Required
1.	**Introduction:** What is Soft Computing? Difference between Hard and Soft computing, Requirement of Soft computing, Major Areas of Soft Computing, Applications of Soft Computing.	4
2.	**Neural Networks:** Introduction, what is Neural Network, Learning rules and various activation functions, Supervised Learning Networks, Single layer Perceptron, Back Propagation networks, Architecture of Backpropagation (BP) Networks, Backpropagation Learning, Variation of Standard Back propagation Neural Network, Introduction to Associative Memory, Adaptive Resonance theory and Self Organizing Map, Recent Applications. Unsupervised Learning Networks**.**	16
3.	**Fuzzy Systems:** Fuzzy Set theory, Fuzzy vs. Crisp set, Fuzzy Relation, Fuzzification, Minmax Composition, Defuzzification Method, Fuzzy Logic, Fuzzy Rule based systems, Predicate logic, Fuzzy Decision Making, Fuzzy Control Systems, Fuzzy Classification.	16
4.	**Genetic Algorithm:** History of Genetic Algorithms (GA), Working Principle, Various Encoding methods, Fitness function, GA Operators- Reproduction, Crossover, Mutation, Convergence of GA, Bit wise operation in GA, Multi-level Optimization.	16
5.	**Introduction to Hybrid Systems.**	8
	Total Lectures	60

Suggested Text Book(s):

1. S.N. Sivanandam, S.N. Deepa, Principles of Soft Computing, Wiley India.
2. Samir Roy, Udit Chakraborty, Soft Computing, Pearson India.

Suggested Reference Book(s):

1. Padam Gulwani Anshuman Sharma, Fundamentals of Soft Computing and Intelligent System, Wiley.
2. Saroj Kaushik and Sunita Tewari, Soft Computing, Tata McGraw Hill.

Other Useful Resource(s)

1. https://nptel.ac.in/courses/106105173
2. http://vlabs.iitkgp.ernet.in/scte/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	3	3	3	1	3	2	1	2	**2.3**
CO2	3	3	1	3	3	2	3	3	**2.6**
CO3	2	3	2	3	2	2	3	3	**2.5**
AVG.	**2.7**	**3.0**	**2.0**	**2.3**	**2.7**	**2.0**	**2.3**	**2.7**	**2.5**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	2	1	3	**2.0**
CO2	2	3	3	**2.7**
CO3	1	2	3	**2.0**
AVG.	**1.7**	**2.0**	**3.0**	**2.2**

SUBJECT: COMPUTER SCIENCE

DSE	BCS-E805	Theory of Computation	L	T	P	C	Time for ESE
			5	1	-	6	3 Hrs.

Pre- requisite: A strong background in discrete mathematics, data structures, and algorithms.

Course Objectives:

- To give an overview of the theoretical foundations of computer science from the perspective of formal languages
- To illustrate finite state machines to solve problems in computing
- To explain the hierarchy of problems arising in the computer sciences.
- To familiarize Regular grammars, context frees grammar.

Course Outcomes:

CO1	Write a formal notation for strings, languages and machines.
CO2	Design finite automata to accept a set of strings of a language.
CO3	For a given language determine whether the given language is regular or not.
CO4	Design context free grammars to generate strings of context free language.
CO5	Determine equivalence of languages accepted by Push Down Automata and languages generated by context free grammars
CO6	Write the hierarchy of formal languages, grammars and machines.
CO7	Distinguish between computability and non-computability and Decidability and undecidability.

Course Contents

UNIT	Contents	Lectures Required
1.	Introduction to Languages; Recursive Definitions; Regular Expressions; Finite Automata; Transition Graphs; Kleene's Theorem.	12
2.	Non- Deterministic Finite Automata, Finite Automata with Output - Moore and Mealy machines, Equivalence of Moore and Mealy machines; Regular Languages; Non-regular Languages; Decidability.	12
3.	Context-Free Grammars. Trees; Regular Grammars; Chomsky's Normal Form; Pushdown Automata; Context- Free Languages; Non- Context- Free Languages; Parsing; Decidability for CFG - Emptiness, Finiteness and Membership questions about CFG.	12
4.	Turing Machines (TM); Post Machines (PM), simulating a PM on TM.	12
5.	Recursively Enumerable Languages; Encoding of Turing Machines; Phrase Structure Grammar; Context Sensitive Grammar; Defining the Computer and Computable Functions; Church’s Thesis; Halting Problem for Turing Machines.	12
	Total Lectures	60

Suggested Text Book(s):

1.	Cohen, Daniel I.A., Introduction to Computer Theory, John Wiley & Sons

2.	K. L. P. Mishra and N. Chandrasekaran, Theory of Computer Science: Automata Languages and Computation, PHI

Suggested Reference Book(s):

1.	Lewis & Papadimitriou, Elements of the theory of computation, PHI.
2.	Hoperoft, Aho, Ullman, Introduction to Automata theory, Language & Computation, Pearson Education.

Other Useful Resource(s)

1.	https://nptel.ac.in/courses/111103016/
2.	https://ocw.mit.edu/courses/mathematics/18-404j-theory-of-computation-fall-2006/
3.	http://cse.iitkgp.ac.in/~abhij/course/theory/FLAT/Spring13/

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	1	3	3	1	1	2	2	2	**1.9**
CO2	3	3	3	2	2	3	2	3	**2.6**
CO3	2	3	1	1	3	3	2	2	**2.1**
CO4	3	3	2	2	2	3	3	3	**2.6**
CO5	3	2	2	1	3	2	3	2	**2.3**
CO6	1	1	3	1	1	1	1	2	**1.4**
CO7	2	1	1	2	3	2	1	1	**1.6**
AVG.	**2.1**	**2.3**	**2.1**	**1.4**	**2.1**	**2.3**	**2.0**	**2.1**	**2.1**

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	1	1	**1.7**
CO2	3	3	3	**3.0**
CO3	3	3	2	**2.7**
CO4	3	2	3	**2.7**
CO5	3	1	3	**2.3**
CO6	3	1	2	**2.0**
CO7	2	1	1	**1.3**
AVG.	**2.9**	**1.7**	**2.1**	**2.4**

SUBJECT: COMPUTER SCIENCE

DSC	BCS-C861	Industrial Training/ Research Project/ Dissertation	L	T	P	C	Time for ESE
			-	-	-	6	3 Hrs.

Pre- requisite: Fundamental knowledge of programming languages and computer science.

Course Objectives:

- To provide to students the feel of the actual working environment and to gain practical knowledge and skills, which in turn will motivate, develop and build their confidence.
- Industrial training is also expected to provide the students the basis to identify their key operational area of interest.

Course Outcomes:

CO1	Participate in the projects in industries during his or her industrial training.
CO2	Describe use of advanced tools and techniques encountered during industrial training and visit.
CO3	Interact with industrial personnel and follow engineering practices and discipline prescribed in industry.
CO4	Develop awareness about general workplace behavior and build interpersonal and team skills.
CO5	Prepare professional work reports and presentations.

Course Outcomes Contributed to Programme Outcomes

PO→ CO↓	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	AVERAGE
CO1	2	-	2	3	1	2	3	3	2.3
CO2	2	2	2	2	1	2	3	3	2.1
CO3	2	2	2	2	2	3	3	3	2.4
CO4	2	2	2	2	2	3	3	3	2.4
CO5	2	2	2	2	2	3	3	3	2.4
AVG.	2.0	2.0	2.0	2.2	1.6	2.6	3.0	3.0	2.3

Course Outcomes Contributed to Programme Specific Outcomes

PSO→ CO↓	PSO1	PSO2	PSO3	AVERAGE
CO1	3	3	2	2.7
CO2	3	3	3	3.0
CO3	2	3	3	2.7
CO4	2	2	3	2.3
CO5	3	3	3	3.0
AVG.	2.6	2.8	2.8	2.8