DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MINOR SCHEME

Minor in Computer Science													
Sl. No.	Course and Course code		Course Title	Teaching Dept.	Teaching Hours/Week				Examination				Credits
					Theory Lecture	Tutorial	Practical/ Drawing	PBL	Duration in hr	CIE Marks	SEE Marks	Total Marks	
					L	T	P	J					
1		CS1006-1	Fundamentals of C Programming	CS	2	0	2	0	04	50	50	100	**3**
		CS2005-1	Introduction to Object oriented programming										
2		CS2006-1	Fundamentals of Data Structures	CS	2	0	2	0	04	50	50	100	**3**
3		CS3006-1	Introduction to the Design and Analysis of Algorithms	CS	2	0	2	0	04	50	50	100	**3**
4		CS2009-1	Fundamentals of Database Management Systems	CS	2	0	2	0	04	50	50	100	**3**
5		CS2104-1	Overview of Operating Systems	CS	3	0	0	0	03	50	50	100	**3**
6		CS2007-1	Introduction to Machine Learning	CS	2	0	2	0	04	50	50	100	**3**
		CS3007-1	Fundamentals of Computer Network and communication										
TOTAL					**13**	**0**	**10**	**0**	**23**	**300**	**300**	**600**	**18**

FUNDAMENTALS OF C PROGRAMMING

Course Code:	**CS1006-1**		
Teaching Hours/Week (L: T: P)	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisites	**Nil**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Make students learn the basics of C programming language including the basic data types, Operators and Evaluating expressions in C.
2.	Apply the concepts of decision making and looping in problem solving to demonstrate its usage using simple programs.
3.	Apply the concepts of Arrays, User-defined functions and code reusability in problem solving along with parameter passing and returning with the help of user defined functions.
4.	Demonstrate the usage of Strings and Structures
5.	Demonstrate the usage of Pointers, and File handling that are essential for understanding the concepts with simple examples.

UNIT-I

Introduction To C Programming Language	**10 Hours**
Basic C DataTypes, operators, Operator precedence, Arithmetic expressions and type conversion.	
Decision Making and Branching:	
Decision making with if statement, Nesting of if…else statements, ternary operator, the switch statement, the go to statement, break and continue statements.,	
Decision Making and Looping:	
The while statement, the do…while statement, the for statement, Jumps in Loops.	

UNIT-II

Arrays	**10 Hours**
Arrays (1-D, 2-D) Initialization and Declaration.	
User-Defined Functions	
Argument Passing – call by value, call by reference, Category of Functions. Managing Command line arguments Examples: Linear Search, Binary Search, Bubble sort, Selection Sort, Trace and Transpose, Matrix Multiplication.	
Strings	
Declaring and Initializing strings, String manipulation functions.	

UNIT-III

Structures	**05 Hours**
Structures and Unions: Usage and nesting	
Pointers and File Handling:	
Accessing of variables using Pointers, array of pointers	

Basic file operations: Open, Close, Read, Write, Append and concatenate

Suggested List of Experiments

PART A

1.	Write a C program to find the roots of a quadratic equation $ax^2+bx+c=0$
2.	Write a C program to find the sum of all the digits and occurrence of a digit in the number.
3.	Write a C program to find the GCD and LCM of given two numbers using Euclid's method.
4.	Write a C program to print the prime numbers in a given range.
5.	Write a C program to find if a given string is a palindrome or not using string manipulation functions.
6.	Write a C program to input N real numbers in 1-D array. Compute mean, variance and Standard Deviation. [Mean= sum/N, Variance = Σ (Xi-mean) 2 /N, STD Deviation= √variance.]
7.	Write a C program to read N integers into an array A and find the sum of elements using pointers.
8.	Write a C program to copy contents of one file to another file.
9.	Write a C program to perform a linear search for a given key integer in a single dimensional array of numbers and report success or failure in the form of a suitable message using functions.
10.	Write a program to demonstrate the use of pointers and files.
11.	Write a C program to enter the information like name, register number, marks in 6 subjects of N students into an array of structures, and find the average & display grade based on average for each student.

Average	**Grade**
80-100	Distinction
60-79	First Class
40-59	Second Class
<40	Fail

12.	Write a C program to read the values into a two dimensional array A, find the sum of all the elements of a row, sum of all the elements of a column, find the total sum of all the elements of the two dimensional array A and print the results.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→ / **↓ Course Outcomes**	1	2	3	4	5	6	7	8	9	10	11	12	**PSO↓** 1	2
CS1006-1.1	3	-	-	-	-	-	-	-	-	-	-	-	-	2
CS1006-1.2	2	3	-	-	-	-	-	-	-	-	-	-	-	3
CS1006-1.3	2	3	-	-	-	-	-	-	-	-	-	-	-	3
CS1006-1.4	2	2	3	-	-	-	-	-	-	-	-	-	-	3
CS1006-1.5	2	3	-	-	-	-	-	-	-	-	-	-	-	3

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	E. Balaguruswamy, "Programming in ANSI C", Tata McGraw Hill, 3rd Edition, 2004.
2.	Jacqueline A. Jones & Keith Harrow, "C Programming with Problem Solving", Pearson,
REFERENCE BOOKS:	
1	Kernighan & Ritchie, "The C Programming (ANSI C)", Prentice Hall; 2ndEdition, 1998.
2	Rajiv Khanna, "Computer Concepts and C Programming", New Age International Pvt Ltd Publishers, 1st Edition, 2006.
3	Yashwant Kanetkar, "Let Us C", 5th Edition, BPB Publications, New Delhi, 2004.
E Books / MOOCs/ NPTEL	
1	http://www.lysator.liu.se/c/bwk-tutor.html#introduction
2	http://www.acm.uiuc.edu/webmonkeys/book/c_guide/
3	C programming Tutorial by Mark Burgers http://markburgess.org/CTutorial/C-Tut-4.02.pdf
4	http://nptel.ac.in/courses/106105085/4
5	https://www.lynda.com/C-training-tutorials/1249-0.html

INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

Course Code:	**CS2005-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisite	**CS1001-1/CS1004-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Learn fundamental features of object-oriented language and JAVA programming constructs.
2.	Develop and run simple Java programs using OOPS concepts of java.
3.	Create multi-threaded programs and event driven Graphical User Interface (GUI) programming using swing package.

UNIT-I

INTRODUCTION:	**10 Hours**

I**ntroducing Classes**–Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, this keyword, Method overloading, using objects as parameters, Argument passing, returning objects, Access control, static, final, Using command line arguments, variable length arguments.

Inheritance – Inheritance Basics, using super, creates a Multilevel Hierarchy, when constructors are called? Method Overriding, using abstract classes, Using final with Inheritance.

(Textbook 1, Chapter – 6-9)

UNIT-II

Packages & Exception Handling:	**10 Hours**

Packages and Interfaces – Packages, Access protection, Importing Packages, Interfaces.
Exception Handling – Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, multiple catch Clauses, Nested try statements, throw, and throws, finally.
(Textbook 1, Chapter – 10-11, 19, 22)

UNIT-III

File Handling & String Handling	**5 Hours**

File Handling – Serial Access Files, File Methods.
Java basic string handling(Basic Methods)

Suggested List of Experiments

1.	Use java program to demonstrate the OOP concepts.
2.	Demonstrate the file handling using Java
3.	Implement the java programs that uses the concepts of exception handling.
4.	Develop Java program to store and retrieve data from database.
5.	Java programs to establish network connectivity
6.	Demonstrate the web application development using servlets and JSP
7.	Mini Project

Course Outcomes: At the end of the course student will be able to

1.	Develop classes and apply object-oriented features to solve real world problems.
2.	Develop robust Java programs using exception handling features, implement multiple inheritance using interfaces and organize the application classes using packages.
3.	Develop programs that can run concurrent tasks using multithreading and perform basic file operations.
4.	Develop GUI applications using Java swings and manage various events generated by user interactions with the UI using event handling mechanisms.
5.	Develop type independent classes using generics; Choose and apply the right data structure to manage collection of data using the collections framework.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO↓**		
↓ Course Outcomes													**1**	**2**	**3**
CS2005-1.1	3	1	3		1							2	2	3	
CS2005-1.2	3	1	3		2							2		3	
CS2005-1.3	3	1	3		3							2		3	
CS2005-1.4	3	1	3		3							2	2	3	
CS2005-1.5	3	1	3		3							2		3	

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	Herbert Scheldt, Java the Complete Reference, 7th Edition, Tata McGraw Hill, 2007.
2.	Jan Graba, An Introduction to Network Programming with Java, 2007, Springer Publications.

REFERENCE BOOKS:

1.	Mahesh Bhave and Sunil Patekar, "Programming with Java", First Edition, Pearson Education, 2008, ISBN:9788131720806
2.	Rajkumar Buyya,SThamarasiselvi, xingchen chu, Object oriented Programming with Java, Tata McGraw Hill education private limited.
3.	Richard A Johnson, Introduction to Java Programming and OOAD, CENGAGE Learning.
4.	E Balagurusamy, Programming with Java A primer, Tata McGraw Hill companies.
E Books / MOOCs/ NPTEL	
1.	Online course material by Oracle :http://docs.oracle.com/javase/tutorial/index.html
2.	https://www.udemy.com/courses/search/?q=java&price=pricefree&view=grid
3.	Oracle: www.oracle.com/events/global/en/java.../java-a-beginners-guide-1720064.pdf
4.	NPTEL:www.nptelvideos.com/java/java_video_lectures_tutorials.php

FUNDAMENTALS OF DATA STRUCTURES

Course Code:	**CS2006-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisite	**CS1001-1/CS1004-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Outline the concepts of data structure, it's operations, Memory allocation functions and design the programs using arrays and structures, pointers, pointer to structure.
2.	Implement linear data structure stack and usage of stacks in various applications.
3.	Implement linear data structure Ordinary Queue, Circular Queue and priority queues
4.	Implement the operations of singly linked lists and circular linked lists, doubly linked list and circular doubly lists.
5.	Identify and differentiate different types of binary trees and binary search trees data structures and also implement them and illustrate threaded binary trees, expression trees, graph representation and techniques of hashing.

UNIT-I

10 Hours

Introduction: Data Structure Definition, Classification (Primitive and non-primitive), data structure

operations.

Arrays and Structures: Array of Structures and Pointer to Structure, Programming Example.

Linear Data structures-Stack: Introduction and Definition, **Representation of stack:** Representation of stacks, Primitive operations on stacks

Applications of Stack: Conversion of Expressions

Algorithms and C programs with tracing Examples: For evaluating postfix expression, infix to postfix conversion.

Recursion: Definition, Implementation, Examples on Recursion with tracing: Factorial function, Fibonacci sequence.

UNIT-II

10 Hours

Linear Data structures-Queue: Introduction and Definition, Representation of Queue, circular queue, priority queue.

Dynamic Memory Allocation functions with programming examples

Linear Data structures-Linked List: Singly Linked List Using dynamic variables ,Inserting and deleting nodes, Other list Operations on singly Linked List, Header Nodes,

Circular Linked List, Doubly Linked List and Circular doubly Link list : Representation and Operations

UNIT-III

5 Hours

Non linear Data structures – Tree data structures:

Introduction: Tree definition, Terminology, **Binary Trees:** Definition, Types, Properties, **Representation of Binary Tree**: Array representation, Linked representation, Binary Tree traversals-Preorder, Inorder and postorder. **Binary Search Tree**: Definition, Construction- Searching, Insertion operations, deletion process, Traversal examples

Suggested List of Experiments

1.	Programs on arrays and structures using Pointers
2	Stack and Ordinary Queue implementation using array and structure.
3	Application of stack data structure- (i) Evaluation of post fix and Tower of Hanoi problem using recursion. (ii) Conversion infix to postfix
4	Circular queues.
5	Operation on Singly Linked list implementation using dynamic variables.
6	Dynamic implementation of stack and Queue data structure.
7	Circular linked list
8	Doubly linked list implementation.
9	Binary Search Tree Construction and Tree traversal operations.

Course Outcomes: At the end of the course student will be able to

1.	**Acquire** the fundamental knowledge of various types of data structures, dynamic memory allocation and designthe programs using arrays, structures and pointers
2.	**Apply** the fundamental knowledge of data structures to **design** stack and use them for solving problems.
3.	**Apply** the fundamental knowledge of data structures to **design** queues and use them for solving problems.
4.	**Design and develop** singly linked lists, circular linked lists and doubly linked list.
5.	**Acquire** theknowledge of trees and **employ** binary trees and binary search tree data

	structure, advanced trees, representation of graphs and hashing techniques.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→ / ↓ Course Outcomes	1	2	3	4	5	6	7	8	9	10	11	12	PSO↓ 1	2	3
CS2006-1.1	3	1	3									1	3		
CS2006-1.2	3	1	3									1	3		
CS2006-1.3	3	1	3									1	3		
CS2006-1.4	3	1	3									1	3		
CS2006-1.5	3	1	3									1	3		

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	Aaron M. Tenenbaum, YedidyahLangsam& Moshe J. Augenstein, "Data Structures using C", Pearson Education/PHI, 2009.
2.	Ellis Horowitz and Sartaj Sahni, "Fundamentals of Data Structures in C", 2nd edition, Universities Press, 2014.

REFERENCE BOOKS:

1.	Seymour Lipschutz, "Data Structures, Schaum's Outlines", Revised 1st edition, McGraw Hill, 2014.

E Books / MOOCs/ NPTEL

1.	Data Structures Using C, ISRD Group, Tata McGraw Hill, 2006.
2.	Data Structures Using C, Reema Thareja, 2nd edition, Oxford University Press, 2014
3.	Introduction to Data Structures by edx , URL: https://www.edx.org/course/
4.	Data structures by Berkley, URL: https://people.eecs.berkeley
5.	Advance Data Structures by MIT OCW , URL: https://www.mooclab.club/
6.	Data Structure by Harvard Extension School, URL: http://www.extension.harvard.

INTRODUCTION TO THE DESIGN AND ANALYSIS OF ALGORITHMS

Course Code:	**CS3006-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisite	**CS2006-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Understand the notion of algorithms, Algorithm design and analysis process, asymptotic notations and Analyze the non-recursive and recursive algorithms and to represent efficiency of these algorithms in terms of the standard asymptotic notations.
2.	Devise the Brute Force and Divide and Conquer techniques to design the algorithms and apply these methods in designing algorithms to solve a given problem.
3.	Apply the Decrease and Conquer, Transform and Conquer algorithm design techniques to solve a given problem.
4.	Get idea of Time versus Space Trade-offs and Apply and Analyse dynamic programming methods in designing algorithms to solve a given problem.
5.	Describe and illustrate the idea of Greedy method, Backtracking and Branch and Bound algorithm design techniques to solve a given problem and to describe P, NP and NP Complete problems.

UNIT-I

10 Hours

INTRODUCTION: What is an Algorithm? Fundamentals of Algorithmic Problem Solving

FUNDAMENTALS OF THE ALGORITHMS EFFICIENCY: Analysis Framework, Asymptotic Notations and Basic efficiency classes, Mathematical Analysis of Non-recursive and Recursive Algorithms,

BRUTE FORCE: Background, Selection Sort and Bubble sort, Brute-Force String Matching algorithms

DIVIDE AND CONQUER: General Method, Merge sort, Binary Search algorithms with Complexity analysis
(Text Book-1: Chapter 4: 4.1 to 4.3)

UNIT-II

10 Hours

DECREASE & CONQUER: General method, Insertion Sort algorithm, **Graph algorithms:** Depth First Search, Breadth First Search.
TRANSFORM AND CONQUER: General method, **Balanced Search Trees:** AVL trees, Heaps and Heap sort algorithms with complexity analysis
TIME AND SPACE TRADEOFFS: Input Enhancement in String Matching: Horspool's algorithm and analysis
DYNAMIC PROGRAMMING: General method, The Floyd-Warshall Algorithm,

UNIT-III

5 Hours

GREEDY TECHNIQUE:
General method of Greedy technique, **Minimum Spanning Trees:** Prim's Algorithm, Huffman Trees
BACKTRACKING: General method, State space trees and algorithms for N-Queens problem

BRANCH AND BOUND: General method,Travelling Salesman problem,

Suggested List of Experiments

1	Various Sorting/Searching algorithms
2	Graph traversals –DFS and BFS, Connectivity and Reachability of graphs
3	Descending Priority Queue using Heap
4	Horspool string matching algorithm
5	Binomial coefficient, Warshall's algorithm, Floyd's algorithm,
6	Prim's,
7	N-Queens problem.

Course Outcomes: At the end of the course student will be able to

1.	Analyze the performance of the algorithms, state the efficiency using asymptotic notations and analyze algorithms mathematically for the complexity of the algorithm.
2.	Apply Brute force method, divide and conquer approaches in solving the problems and analyze the same.
3.	Apply the appropriate algorithmic design technique like decrease and conquer approaches, transform and conquer approaches and compare the efficiency of algorithms to solve the given problem.
4.	Apply and analyze dynamic programming approaches to solve some problems. And improve an algorithm time efficiency by sacrificing space.
5.	Apply and analyze greedy method, backtracking, branch and bound methods to solve problems and to describe P, NP and NP Complete problems.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→	1	2	3	4	5	6	7	8	9	10	11	12	PSO↓		
↓ Course Outcomes													1	2	3
CS3006-1.1	2	3	2	2								1		2	
CS3006-1.2	2	2	2	2								1		3	
CS3006-1.3	2	3	2	2								1		3	
CS3006-1.4	2	2	2	2								1		3	
CS3006-1.5	2	3	2	2								1		2	

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	Anany Levitin, "Introduction to the Design & Analysis of Algorithms", 2nd Edition, Pearson Education, 2011.
2.	Thomas H. Cormen, Charles E.Leiserson, Ronald L. Rivest, Clifford Stein, "Introduction to Algorithms", 3rd Edition, PHI, 2014.

REFERENCE BOOKS:

1.	Horowitz E., Sahni S., Rajasekaran S, "Computer Algorithms", Galgotia Publications, 2001.
2.	R.C.T. Lee, S.S. Tseng, R.C. Chang &Y.T.Tsai, "Introduction to the Design and Analysis of Algorithms

	A Strategic Approach", Tata McGraw Hill, 2005.
E Books / MOOCs/ NPTEL	
1.	http://www.facweb.iitkgp.ernet.in/~sourav/daa.html
2.	http://nptel.ac.in/courses/106101060/https://www.coursera.org/specializations/algorithms
3.	http://nptel.ac.in/courses/106101060/https://www.coursera.org/specializations/algorithms
4.	http://elearning.vtu.ac.in/econtent/courses/video/CSE/06CS43.html
5.	http://elearning.vtu.ac.in/econtent/courses/video/FEP/ADA.html
6.	http://openclassroom.stanford.edu/MainFolder/CoursePage.php?course=IntroToAlgorithms

FUNDAMENTALS OF DATABASE MANAGEMENT SYSTEMS

Course Code:	**CS2009-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits**	**03**
Total Teaching Hours	**25+0+26**	**CIE + SEE Marks**	**50+50**
Prerequisite	**CS1002-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Provide a strong foundation in database concepts, design and application.
2.	Understand the concepts of relational model and relational algebra in database design.
3.	Learn structured **query** language (**SQL**) to an intermediate/advanced level and evaluate the result set.
4.	Understand the use of normalization techniques for building effective database design. Learn the concepts of NOSQL Systems to manage bigdata
5.	Demonstrate the use of File organization andIndexing,Concurrency Control and transactions in databases.

UNIT-I

Databases and Database users, Database System Concepts	**10 Hours**

Databases and Database Users: Introduction, An Example, Characteristics of the database approach. **Database System Concepts and Architecture:** Three-Schema Architecture and data Independence. **Data Modeling Using the Entity–Relationship (ER) Model:** Entity Types, Entity Sets, Attributes and Keys, Relationship types, Relationship Sets, Roles and Structural Constraints, Weak Entity Types, Refining the ER Design for the COMPANY Database, ER Diagrams. **The Relational Data Model and Relational Database Constraints:** Relational Model Concepts, Relational Model Constraints and Relational Database Schemas, Update Operations, transactions, and dealing with constraint violations. **The Relational Algebra and Relational Calculus:** Unary Relational Operations: SELECT and PROJECT, set operations Binary Relational Operations: JOIN

Relational Database Design by ER-to-Relational Mapping: Relational Database Design Using ER-to-Relational Mapping.

UNIT-II

Basic SQL: **10 Hours**

Basic SQL:SQL Data Definition and Data Types, Specifying Constraints in SQL, Basic retrieval queries in SQL, Insert, Delete and Update Statements in SQL,

More SQL: Complex Queries, Views, and Schema Modification: More complex SQL retrieval queries, Specifying constraints as assertions and Actions as Triggers, Views in SQL

UNIT-III

Functional Dependencies and Normalization: **5 Hours**

Basics of Functional Dependencies and Normalization for Relational Databases: Functional Dependencies, Normal Forms Based on Primary Keys, general definitions of Second and Third Normal Forms.

Relational Database Design Algorithms and Further Dependencies: Inference Rules, Equivalence,Properties of Relational Decomposition

Introduction to transaction Processing and locks

Course Outcomes: At the end of the course student will be able to

1.	Illustrate the concepts of database objects for the given problem.
2.	Identify and enforce integrity constraints on a database using RDBMS.
3.	Apply structured query language for (SQL) for database manipulation.
4.	Model normalized database structures by creating simple database systems. Understan the concepts of NOSQL Systems to manage bigdata
5.	Illustrate the concepts of File organization and Indexing,Concurrency Control and transactions in databases.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→	1	2	3	4	5	6	7	8	9	10	11	12	**PSO↓**		
↓ Course Outcomes													1	2	3
CS2009-1.1	2									1		1			
CS2009-1.2	2	2								1		1			
CS2009-1.3	2	3								1		1		3	
CS2009-1.4	2	2	3							1		1		2	
CS2009-1.5	2									1		1		2	

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	Database Systems Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson.
2.	Database Management Systems, Raghu Ramakrishnan,JohannesGehrke, Indian Edition, Mc Graw Hill Education.

REFERENCE BOOKS:

1.	Database Systems: Models, Languages, Design and Application Programming, RamezElmasri and Shamkant B. Navathe, 6th Edition, 2017, Pearson.
2.	Database System Concepts, SilberschatzKorth and Sudharshan, 6thEdition, McGraw Hill, 2013.

E Books / MOOCs/ NPTEL

1.	https://www.udemy.com/course/introduction-to-basic-database-concepts/, Introduction to Basic Database Concepts (Udemy).
2.	https://www.udemy.com/course/database-management-systems-mysql/, Database Management Systems – MySQL (Udemy).
3.	https://swayam.gov.in/nd1_noc19_cs46/preview, Database Management System (Swayam).
Suggested List of Experiments	
1	Design and implementation of SQL queries involving various constructs of SQL as discussed in the Unit-III of the syllabus. **Note:** 1. Create the tables by properly specifying the primary keys and the foreign keys. 2. Enter at least four tuples for each relation
	Insurance Database Consider the Insurance database given below. PERSON (driver – id #: String, name: string, address: string) CAR (regno: string, model: string, year: int) ACCIDENT (report-number: int, accd-date: date, location: string) OWNS (driver-id #: string, regno: string) PARTICIPATED (driver-id: string, Regno: string, report-number: int, damage amount: int) 1. Find the total number of people who owned cars that were involved in accidents in 1989. 2. Find the number of accidents in which the cars belonging to "John Smith" were involved. 3. Update the damage amount for the car with reg number "KA-12" in the accident with report number "1" to $3000.
	Order Database Consider the following relations for an order processing database application in a company: CUSTOMER (cust #: int, cname: string, city: string) ORDER (order #: int, odate: date, cust #: int, ord-Amt: int) ORDER – ITEM (order #: int, item #: int, qty: int) ITEM (item #: int, unit price: int) SHIPMENT (order #: int, warehouse#: int, ship-date: date) WAREHOUSE (warehouse #: int, city: string) 1. Produce a listing: CUSTNAME, #oforders, AVG_ORDER_AMT, where the middle column is the total numbers of orders by the customer and the last column is the average order amount for that customer.

	2. For each item that has more than two orders , list the item, number of orders that are shipped from atleast two warehouses and total quantity of items shipped 3. List the customers who have ordered for every item that the company produces
	Consider the following database of student enrollment in courses & books adopted for each course: STUDENT (regno: string, name: string, major: string, bdate: date) COURSE (course #: int, cname: string, dept: string) ENROLL (regno: string, course#: int, sem: int marks: int) BOOK _ ADOPTION (course#: int, sem: int, book-ISBN: int) TEXT (book-ISBN: int, book-title: string, publisher: string, author: string) 1. Produce a list of text books (include Course #, Book-ISBN,Book-title) in the alphabetical order for courses offered by th 'CS' department that use more than two books. 2. List any department that has all its adopted books published by a specific publisher **3.** List the bookISBNs and book titles of the department that has maximum number of students
	The following tables are maintained by a book dealer: AUTHOR (author-id: int, name: string, city: string, country: string) PUBLISHER (publisher-id: int, name: string, city: string, country: string) CATALOG (book-id: int, title: string, author-id: int, publisher-id: int, category-id: int, year: int, price: int) CATEGORY (category-id: int, description: string) ORDER-DETAILS (order-no: int, book-id: int, quantity: int) 1. Find the author of the book which has maximum sales. 2. Increase the price of the books published by a specific publisher by 10% 3. Find the number of orders for the book that has minimum sales.
	Consider the following database for a banking enterprise: BRANCH (branch-name: string, branch-city: string, assets: real) ACCOUNT (accno: int, branch-name: string, balance: real) DEPOSITOR (customer-name: string, accno: int) CUSTOMER (customer-name: string, customer-street: string, customer-city: string) LOAN (loan-number: int, branch-name: string, amount: real) BORROWER (customer-name: string, loan-number: int) 1. Find all the customers who have atleast 2 accounts at all the branches located in a specific city. 2. Find all the customers who have accounts in atleast 1 branch located in all the cities

	3. Find all the customers who have accounts in atleast 2 branches located in a specific city.
2	Implementation of a mini project that involves a user interface design, database design and design of SQL queries to suit the need of the designed application.

OVERVIEW OF OPERATING SYSTEMS

Course Code:	**CS2104-1**		
Teaching Hours/Week (L: T: P)	**3:0:0**	**Credits**	**03**
Total Teaching Hours	**40+0+0**	**CIE + SEE Marks**	**50+50**
Prerequisite	**CS1004-1/ CS2006-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Explain the concepts, principles and services of operating system.
2.	Identify fundamental operating system concepts such as processes, inter-process communication, threads, CPU scheduling and demonstrate them.
3.	Assess the need of concurrency and synchronization and apply them to write concurrent programs and analyze the cause for the occurrence of deadlocks and determine solutions to overcome the deadlocks.
4.	Study the concepts of main memory and virtual memory allocation methods and demonstrate them.
5.	Analyze the need for file system concepts, directory implementation and disk scheduling algorithms and demonstrate them.

UNIT-I

Operating System structure: **15 Hours**

Operating System structure: Operating System Services, User and Operating System interface, System calls, System Services, Linkers and Loaders, **Process Management:** Process concept; Process scheduling; Operations on processes; Inter-process communication. **Threads:** Multithreading Models. **CPU Scheduling:** Basic concepts, Scheduling criteria, Scheduling algorithms, Thread scheduling

UNIT-II

Process Synchronization: **15 Hours**

Process Synchronization: The Critical section problem, Peterson's solution, Synchronization hardware, Semaphores, Monitors, Classical problems of synchronization. **Deadlocks:** System model; Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, avoidance, detection and recovery

from deadlock. **Main Memory:** Paging, Structure of page table, Swapping.

UNIT-III

Virtual Memory: **10 Hours**

Virtual Memory: Demand paging, Copy-on-write, Page replacement, Allocation of frames. Implementing File System: File system Concepts, File System Structure, Operations and implementation, Directory implementation, Allocation methods, Free space management. disk scheduling algorithms

Course Outcomes: At the end of the course student will be able to

1.	**Recognize** the structural components of operating system
2.	**Demonstrate** the creation and termination of the processes, threads and CPU scheduling algorithms.
3.	**llustrate** critical section problem and **demonstrate** the Peterson's solution. **Investigate** the Deadlock condition and **determine** the solution to avoid.
4.	**Summarize** Main memory and Virtual Memory allocation methods and **prepare** a page replacement schedule to the given set of page requirement request.
5.	Classify file systems based on operations and implementations and **illustrate** the disk scheduling algorithms.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→ ↓ **Course Outcomes**	1	2	3	4	5	6	7	8	9	1	1	1	**PSO↓** 1	2	3
CS2104-1.1	3	2													2
CS2104-1.2	3	2											1	2	3
CS2104-1.3	3	2											1	2	3
CS2104-1.4	3	2											1	2	3
CS2104-1.5	3	2											1	2	3

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	Operating System Concepts, Abraham Silberschatz, Greg Gagne, Peter B. Galvin, 10th Edition, John Wiley & Sons, 2018, ISBN: 9781119320913.

REFERENCE BOOKS:

1.	D.M Dhamdhere: Operating systems - A concept based Approach, 2nd Edition, Tata McGraw- Hill, 2002.
2.	P.C.P. Bhatt: Operating Systems, 2nd Edition, PHI, 2006.
3.	Harvey M Deital: Operating systems, 3rd Edition, Addison Wesley, 1990.

E Books / MOOCs/ NPTEL

1.	http://www.uobabylon.edu.iq/download/M.S%2020132014/Operating_System_Concepts,_8th_ition%5BA4%5D.pdf
2.	http://iips.icci.edu.iq/images/exam/Abraham-Silberschatz-Operating-System-Concepts--9th2012.12.pdf
3.	http://www.uobabylon.edu.iq/download/M.S%2020132014/Operating_System_Concepts,_8th_Edition BA4%5D.pdf

4.	https://freevideolectures.com/university/iit-bombay/
5.	https://www.cse.iitb.ac.in/~mythili/os/

INTRODUCTION TO MACHINE LEARNING

Course Code:	**CS2007-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisite	**CS1005-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Understand the need and basics of machine learning and learn the Decision Tree model.
2.	Learn ANN and Genetic Algorithm along with its applications.
3.	Explore the various learning algorithms using Supervised Learning.
4.	Understand the important aspects of Analytical Learning and difference between Analytical and Inductive Learning Algorithms.
5.	Explore the difference between Analytical and Inductive Learning Algorithms and analyse the techniques related to reinforcement learning.

UNIT-I

Introduction: Machine learning: what and why?- Types of machine learning, Supervised learning and Unsupervised learning.

Decision tree: Representation, Appropriate problems for decision tree learning and basic decision tree learning algorithm.

Artificial Neural Networks: Introduction, Neural Network Representations, Appropriate problems for

neural network learning, Perceptrons and basics of back propagation algorithm.
(Text Book-1: Chapter 1- 1.1-1.3
Text Book-2: Chapter 3and 4 -- 4.1-4.5.2)

10 Hours

UNIT-II

Bayesian Learning: Bayes theorem, Bayes theorem and concept Learning, Minimum Description Length, Bayes Optimal Classifier, Naive Bayes Classifier.
Instance Based Learning: k-nearest neighbour learning and Locally Weighted Regression.
*A*nalytical Learning: Inductive and Analytical learning problems, PROLOG-EBG.

(Text Book-2: Chapter 6 -- 6.2 - 6.3, 6.6-6.7, 6.9 , Chapter 8 and Chapter 11)

10 Hours

UNIT-III

Combining Inductive and Analytical Learning: Motivation, Inductive –Analytical Approaches to Learning and Using Prior Knowledge to Initialize the Hypothesis.
Reinforcement Learning:
Introduction, Learning Task, Q Learning.
(Text Book-2: 12 and Chapter 13 --13.1-13.2, 13.3.1-13.3.3)

5 Hours

Suggested List of Experiments

	PART-A
1.	Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .CSV file.
2.	For a given set of training data examples stored in a .CSV file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples.
3.	Develop a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.
4.	Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.
5.	Develop a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.
6.	Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering. You can add Java/Python ML library classes/API in the program.
7.	Implement and demonstrate the working of k-Nearest Neighbour algorithm and apply it to classify the iris data set.
8.	Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.
9.	Build a model to classify email as spam or ham. First, download examples of spam and ham from Apache SpamAssassin's public datasets and then train a model to classify email.
	PART-B

Mini Project on Machine Learning:

The main goal is to prepare students to apply machine learning algorithms to real-world tasks, or to leave students well-qualified to start machine learning or AI research. The mini project is intended to start in these directions.

Students shall carry out either one of the following three kinds of projects:

1. Application project. Pick an application that is of interest and explore how best to apply learning algorithms to solve it.
2. Algorithmic project. Pick a problem or family of problems, and develop a new learning algorithm, or a novel variant of an existing algorithm, to solve it.
3. Theoretical project. Prove some interesting/non-trivial properties of a new or an existing learning algorithm. (This is often quite difficult, and so very few, if any, projects will be purely theoretical.)

Course Outcomes: At the end of the course student will be able to

1.	Explain the fundamental concept and importance of machine learning, identify, analyze and categorize applications using decision tree algorithm.
2.	Demonstrate the application of ANN and Genetic algorithm for real world problems.
3.	Design and implement algorithms for supervised learning system.
4.	Design and implement algorithms for Analytical and Inductive Learning.
5.	Develop machine learning algorithm and reinforcement techniques for real world problems.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→ / **↓ Course Outcomes**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**PSO↓** **1**	**2**	**3**
CS2007-1.1	2	2	3										3	3	
CS2007-1.2	2	2	3										3	3	
CS2007-1.3	2	2	3										3	3	
CS2007-1.4	2	3											1	3	
CS2007-1.5	2	2	3										2	3	

1: Low 2: Medium 3: High

TEXTBOOKS:

1.	T. M. Mitchell, "Machine Learning", McGraw Hill, 2017.

REFERENCE BOOKS:

1.	EthemAlpaydin, "Introduction to Machine Learning", Second Edition, The MIT Press, 2004.
2.	C. M. Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.
3.	R. O. Duda, P. E. Hart and D. G. Stork Pattern Classification, Wiley Publications, 2001
4.	T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning, 2e, 2008.
5.	P. Flach, "Machine Learning: The art and science of algorithms that make sense of data", Cambridge University Press, 2012.
6	K. P. Murphy, "Machine Learning: A probabilistic perspective", MIT Press, 2012.
7.	M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of Machine Learning", MIT Press, 2012.
8.	S. Russel and P. Norvig, "Artificial Intelligence: A Modern Approach", Third Edition, Prentice Hall, 2009.

E Books / MOOCs/ NPTEL

1.	https://in.mathworks.com/
2.	https://www.kdnuggets.com/

3.	https://blog.cambridgespark.
4.	https://www.udemy.com/topic/
5.	https://www.mooc-list.com/
6.	https://peltarion.com

FUNDAMENTALS OF COMPUTER NETWORK AND COMMUNICATION

Course Code:	**CS3007-1**		
Teaching Hours/Week (L: T: P):	**2:0:2**	**Credits:**	**03**
Total Teaching Hours:	**25+0+26**	**CIE + SEE Marks:**	**50+50**
Prerequisites	**CS2006-1**		

Teaching Department: Computer Science & Engineering

Course Objectives:

1.	Outline the principles of computer networks and its application
2.	Illustrate the concept of types of network
3.	Identify the issues in network layer and solution for it
4.	Analyze the process of congestion control algorithms.
5.	Illustrate IP Packets and fragmentation process.

UNIT-I

Network Layer (Part-I):	**10 Hours**

Network layer design issues: Store and Forward packet Switching, Services Provided to the Transport Layer, Implementation of Connectionless Service, Implementation of Connection-Oriented Service, Comparison of Virtual Circuit and Datagram Subnets; **Routing algorithms**: The Optimality Principal, Shortest Path Routing, Flooding. Distance Vector Routing, Link state Routing, Hierarchical Routing, Broadcast Routing,
Multicast Routing, Anycast Routing, Routing for Mobile Hosts, Routing in Ad hoc Networks.
Congestion Control Algorithms: Approaches to Congestion Control, Traffic-AwareRouting,AdmissionControl,TrafficThrottling,LoadShedding;

UNIT-II

Network layer (Part – II):	**10 Hours**

Quality of Service: Application Requirements, Traffic Shaping, Packet Scheduling, Admission

Control, Integrated Services, Differentiated Services

Internetworking: How networks differ, How Networks Can Be Connected, Tunneling, Internetwork Routing, Fragmentation;

The Network Layer in the Internet: The IP Version 4 Protocol, IP Addresses, IP Version 6, Internet Control Protocols, OSPF, BGP, Internet Multicasting, Mobile IP.

The Transport Layer:

The Transport Service: Services Provided to the Upper Layers, Transport Service Primitives,

Elements of Transport Protocols: Addressing, Connection Establishment, Connection Release, Error Control and Flow Control, Multiplexing, Crash Recovery;

UNIT-III

The Internet Transport Protocols (TCP)/UDP	**05 Hours**

The Internet Transport Protocols (UDP): Introduction to UDP

The Internet Transport Protocols (TCP): Introduction to TCP, The TCP Service Model, The TCP Protocol, The TCP Segment Header, TCP Connection Establishment, TCP Connection Release, TCP Connection Management Modeling, TCP Sliding Window, TCP Timer Management, TCP Congestion Control.

The Application Layer:

Principles of Network Applications:

Network Application Architectures, Processes Communicating, Transport Services Available to Applications, Transport Services Provided by the Internet;

Suggested List of Experiments

	PART-A
1	Write a program for simple RSA algorithm to encrypt and decrypt the data.
2	Write a program for error detecting code using CRC-CCITT (16-bits).
3	Write a program for Hamming Code generation for error detection and correction.
4	Write a program for frame sorting technique used in buffers.
5	Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.
6	Write a program for distance vector algorithm to find suitable path for transmission.
7	Write a program for congestion control using Leaky bucket algorithm.
	PART-B
1	Simulate a three nodes point-to-point network with duplex links between them. Set the queue size vary the bandwidth and find the number of packets dropped.
2	Simulate a four node point-to-point network, and connect the links as follows: n0-n2, n1-n2 and n2-n3. Apply TCP agent between n0-n3 and UDP n1-n3. Apply relevant applications over TCP and UDP agents changing the parameter and determine the number of packets by TCP/UDP.
3	Simulate the different types of Internet traffic such as FTP a TELNET over a network and analyze the throughput.
4	Simulate the transmission of ping messaged over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion
5	Simulate an Ethernet LAN using N-nodes (6-10), change error rate and data rate and compare the throughput.
6	Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and determine collision across

	different nodes.
7	Simulate an Ethernet LAN using N nodes and set multiple traffic nodes and plot congestion window for different source/destination.

Course Outcomes: At the end of the course student will be able to

1.	Express the basic concept of computer network.
2.	Design the network layer and the related issues.
3.	Explain the congestion control, and prevention methods.
4.	Explain different type of networks and protocols.
5.	Explain network applications and describe application layer protocols.

Course Outcomes Mapping with Program Outcomes & PSO

Program Outcomes→	1	2	3	4	5	6	7	8	9	10	11	12	PSO↓		
↓ Course Outcomes													1	2	3
CS3007-1.1	3	2			1					1			2		
CS3007-1.2	3	2	1							2		1	2		
CS3007-1.3	3	1			1		1			1		1	3		
CS3007-1.5	3	2	1		1		1			2		1	3	1	
CS3007-1.5	3	1			1	1				1		2	2	1	

1: Low 2: Medium 3: High

TEXTBOOKS:

1	Computer Networks,AndrewS.TanenbaumDavidJ.Wetherall,5th Edition, Pearson, ISBN 10: 1292024224,2014
2	Computer Networking. A Top-down Approach, James F. Kurose, Keith W. Ross, Pearson, ISBN: 1292153598,2017

REFERENCE BOOKS:

1	Computer Networking. ATop-Down Approach, Kurose & Ross, 5th Edition, Mc Graw Hill, ISBN: 9780073376226,2013
2	Data and Computer Communication, 8th Edition, WilliamStallings, Prentice Hall, 0132433109,2007
3	An Introduction to Computer Networks, Peter L Dordal, OpenBook, http://intronetworks.cs.luc.edu/, 2020

E Books / MOOCs/ NPTEL

1	Computer Networks web course by Prof. Ajit Pal, IIT Khargpur, https://nptel.ac.in/courses/106/105/106105080/
2	Computer Networks, Fall 2019 by Peter Dordal, Loyola University Chicago Dept of Computer Science, http://pld.cs.luc.edu/courses/443/now/
3	Computer Networks Lecture Notes, MIT Open Course Ware, Massachusetts Institute of Technology, https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-829-computer-networks-fall-2002/lecture-notes/