UNIVERSITY SCHOOL

OF

INFORMATION AND COMMUNICATION TECHNOLOGY

Department Of Information Technology

COURSE STRUCTURE

B. TECH

Information Technology

4 YEAR PROGRAMME

2023-2027

GAUTAM BUDDHA UNIVERSITY
Gautam Budh Nagar, Greater Noida,
Uttar Pradesh 201308

B.Tech IT (2023-27)

SEMESTER I

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CS101	Fundamentals of Computer Programming	3	1	0	4	CC1 / FC
2	MA101	Engineering Mathematics-I	3	1	0	4	GE1
3	EE102	Basic Electrical Engineering	3	1	0	4	GE2
4	ME101	Engineering Mechanics	3	1	0	4	GE3
5	ES101	Environmental Studies	3	1	0	4	OE1 / AECC
6	EN151	Language Lab	0	0	2	1	OE L1 / AECC
7	CS181	Computer Programming Lab	0	0	2	1	CC-L1 / SEC
8	EE104	Basic Electrical Engineering Lab	0	0	2	1	GE-L1
9	ME102	Workshop Practice	1	0	2	2	GE-L2 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**16**	**5**	**8**	**25**	

SEMESTER II

S. No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA102	Engineering Mathematics-II	3	1	0	4	GE4
2	PH102	Engineering Physics	3	1	0	4	GE5
3	IT102	Object-Oriented Programming	3	1	0	4	CC2/SEC
4	EC101	Basic Electronics Engineering	3	1	0	4	GE6
5	CS105	Introduction of Artificial Intelligence	2	0	0	2	CC3/FC
6	EN101	English Proficiency	2	0	0	2	OE2/AECC
7	CE103	Engineering Graphics Lab	1	0	2	2	GE-L3
8	PH104	Engineering Physics Lab	0	0	2	1	GE-L4
9	IT181	Object-Oriented Programming Lab	0	0	2	1	CC-L2/SEC

10	EC181	Basic Electronics Engineering	0	0	2	1	GE-L5
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**17**	**4**	**8**	**25**	

SEMESTER III

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT201	Animation and Computer Graphics	3	0	0	3	CC4
2	IT203	Operating System	3	0	0	3	CC5
3	IT205	Software Engineering	3	0	0	3	CC6 / SEC
4	IT207	System Design & Analysis Techniques	3	0	0	3	CC7
5	IT209	Web Technologies	3	0	0	3	CC8
6	MA201	Engineering Mathematics-III	3	1	0	4	GE7
7	IT281	Animation & Computer Graphics Lab	0	0	3	2	CC-L3
8	IT283	Operating System Lab	0	0	3	2	CC-L4 / SEC
9	IT285	Web Technologies Lab I	0	0	3	2	CC-L5 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER IV

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT202	Data Structure	3	0	0	3	CC9
2	IT204	Database Management System	3	0	0	3	CC10
3	IT206	Web Development(PHP)	3	0	0	3	CC11 / SEC
4	IT208	Discrete Structure	3	0	0	3	CC12
5	IT210	Fundamentals of Digital Electronics Circuits	3	0	0	3	CC8
6	IT212	Digital Communication & Coding	3	1	0	4	GE7
7	IT282	Data Structures	0	0	3	2	CC-L3
8	IT284	Database Management System Lab	0	0	3	2	CC-L4 / SEC

9	IT286	Web Development PHP lab	0	0	3	2	CC-L5 / SEC
10	IT288	Digital Electronics Circuits Lab	0	0	3	2	CC-L9
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER V

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT301	Theory of Automata	3	0	0	3	CC11
2	IT303	Computer Networks	3	0	0	3	CC12
3	IT305	Compiler Design	3	1	0	4	CC13
4	IT307	Soft Computing Techniques	3	0	0	3	CC14 / SEC
5	IT 315	Elective 1	3	0	0	3	E1 / DSE
6	IT 319	Elective 2	3	0	0	3	E2 / DSE
7	IT381	Computer Networks Lab	0	0	3	2	CC-L9
8	IT383	Compiler Design Lab	0	0	3	2	CC-L10
9	IT385	Soft Computing Techniques Lab	0	0	3	2	CC-L11 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

SEMESTER VI

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT302	Cloud Computing	3	0	0	3	
2	IT304	Algorithm Design & Analysis	3	0	0	3	
3	IT306	Computer Organization	3	1	0	3	
4	IT308	Information and Network Security	3	0	0	4	
5	IT 315	Elective 1	3	0	0	3	
6	IT 319	Elective 2	3	0	0	3	
7	IT382	Cloud Computing Lab	0	0	3	2	

8	IT384	Algorithm Design & Analysis Lab	0	0	3	2	
9	IT386	Information & Network Security Lab	0	0	3	2	
10	GP	General Proficiency	Non Credit				
Total Hours and Credits			**18**	**1**	**9**	**25**	

SEMESTER VII

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA401	Modeling and simulation	3	0	0	3	
2	IT401	Sensor Networks and IoT	3	0	0	3	
3	IT403	Soft Computing Techniques	3	0	0	3	
4	IT405	Big Data Analytics	3	0	0	3	
5		Elective 5	3	0	0	3	
6	IT481	Sensor Networks and IoT Lab	0	0	3	2	
7	IT491	Industrial Training	0	0	6	3	
8	IT493	Minor Project	0	0	10	5	
9	GP	General Proficiency	Non Credit				
Total Hours and Credits			**15**	**0**	**19**	**25**	

SEMESTER VIII

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT490	Seminar	0	0	3	2	
2	IT492	Major Project	0	0	16	8	
3	IT494	Intenship	0	0	30	15	
4	IT490	Seminar	0	0	3	2	
5	GP	General Proficiency	Non Credit				
Total Hours and Credits			**00**	**0**	**49**	**25**	

SEMESTER - I

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CS101	Fundamentals of Computer Programming	3	1	0	4	CC1 / FC
2	MA101	Engineering Mathematics-I	3	1	0	4	GE1
3	EE102	Basic Electrical Engineering	3	1	0	4	GE2
4	ME101	Engineering Mechanics	3	1	0	4	GE3
5	ES101	Environmental Studies	3	1	0	4	OE1 / AECC
6	EN151	Language Lab	0	0	2	1	OE L1 / AECC
7	CS181	Computer Programming Lab	0	0	2	1	CC-L1 / SEC
8	EE104	Basic Electrical Engineering Lab	0	0	2	1	GE-L1
9	ME102	Workshop Practice	1	0	2	2	GE-L2 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**16**	**5**	**8**	**25**	

FUNDAMENTALS OF COMPUTER PROGRAMMING			
Course Code:	**CS101**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**03 + 02**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 To develops fundamental understanding of computers, its components and programming environment.
2 To create programming logics and learn C language programming concepts.
3 To design and develop algorithms and programs with different data declarations, initialization and related operations.
4 To develop the ability to define and manage functions, array, structures, pointers etc. based on program objective.
5 To understand and develop C programs to handle computer files, their usage and perform various operations on files.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand the Computer fundamentals.
2 Understand the use of various programming concepts and techniques.
3 Understand the C programming fundamentals and applications.
4 Understand C by using arrays, functions, structures and union.
5 Develop the Programs in C using its advance features

UNIT I INTRODUCTION TO COMPUTER AND PROGRAMMING CONCEPTS

Definition, characteristic, generation of computers, basic components of a computer system, memory, input, output and storage units, high level language and low level language, Soft-ware: system software, application software, hardware, firmware, Operating System, compiler, interpreter and assembler, linker, loader, debugger, IDE. Introduction to algorithm and flow chart; representation of algorithm using flow chart symbol, pseudo code, basic algorithm de-sign, characteristics of good algorithm, development of algorithm.

UNIT II INTRODUCTION TO C PROGRAMMING LANGUAGE

Introduction to C programming language , Declaring variables, preprocessor statements, arithmetic operators, programming style, keyboard input , relational operators, introduction, feature of C language, concepts, uses, basic program structure, simple data types, variables, constants, operators, comments, control flow statement :if, while, for, do-while, switch.

UNIT III DATA TYPES AND STRUCTURES

Bitwise operators, Pre-defined and User defined data types, arrays, declaration and operations on arrays, searching and sorting on arrays, types of sorting, 2D arrays, passing 2D arrays to functions, structure, member accessing, structure and union, array of structures, functions, declaration and use of functions, parameter passing, and recursion.

UNIT IV FUNDAMENTALS OF POINTERS

Introduction to pointers, pointer notations in C, Declaration and usages of pointers, operations that can be performed on computers, use of pointers in programming exercises, parameter passing in pointers, call by value, call by references, array and characters using pointers, dynamic memory allocation

UNIT V FILE HANDLING IN C AND ENUM

Introduction to file handling, file operations in C, defining and opening in file, reading a file, closing a file, input output operations on file, counting: characters, tabs , spaces, file opening modes, error handling in input/output operations, sEnumerated data types, use of Enum, declaration of Enum.

Text Books:

[1] C Programming, Herbert Shield
[2] C Programming Language 2nd Edition by Brian, W Kernighan Pearson Education.

Reference Books:

[3] Programming in ANSI C by E. Balagurusamy, Tata Mgraw Hill
[4] C Puzzle Book: Puzzles For The C. Programming Language by Alan R Feuer Prentice Hall-Gale
[5] Expert C Programming: Deep C Secrets (s) by Peter Van Der Linden Dorling Kindersley India.
[6] Introduction To UNIX System by Morgan Rachel Tata Mcgraw Hill Education.
[7] C: A Reference Manual (5th Edition) by Samuel P. Harbison&Samuel P. Harbison.
[8] Programming Using the C Language by Hutchison,R.C, Mcgraw Hill Book Company, New York.
[9] Fundamentals of computers and programming with C, A.K. SHARMA

COMPUTER PROGRAMMING LAB			
Course Code:	**CS181**	**Course Credits:**	**1**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**02**	**Mid Sem. Exam Hours:**	**--**
Total No. of Lectures (L + T):	**10**	**End Sem. Exam Hours:**	**2**
COURSE OBJECTIVES			
1 To develops fundamental understanding C programming environment.			
2 To create programming logics and learn C language programming concepts.			
3 To design and develop algorithms and programs with different data declarations, initialization and related operations.			
4 To develop the ability to define and manage functions, array, structures, pointers etc. based on program objective.			
5 To understand and develop C programs to handle computer files, their usage and perform various operations on files.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Understand the C programming fundamentals.			
2 Understand the use of various programming concepts and techniques.			
3 Understand the C data types and operators with their applications.			
4 Understand C by using arrays, functions, structures and union.			
5 Develop the Programs in C using its advance features.			

LIST OF EXPERIMENTS:

1. Write a program to find the sum of the digits of a number.
2. Write a program to calculate factorial of a number using recursion.
3. Write a program to find the reverse of a given number.
4. Write a program to check whether the year is leap or not.
5. Write a program to take marks of a student of 5 subjects as an input and print the grade.
   marks<40 = FAIL
   marks>=40 and <=59 =GOOD
   marks>=59 and <80 =EXCELLENT
   marks>=80 = OUTSTANDING
6. Perform program number 5 using switch case statement.
7. Write a program to compute the roots of a quadratic equations.
8. Write a program to compute the length of a string using While Loop.
9. Write a program to print the following pattern: -
   a)
```
*
**
***
****
```

```
          *****
    b)       *
            * *
          * * *
         * *  *  *
    c)     0
          1  2
        3  4  5
       6  7  8  9
```

10. Write a program to compute and display the product of two matrices.
11. Write a program to illustrate the difference between call by value and call by reference.
12. Write a program to check whether a given string is palindrome or not.
13. Create a structure called STUDENT having name, reg no., class as its field.
    Compute the size of structure STUDENT.
14. Write a program to compute the length of a string using pointers.
15. Write a program to create a file , input data and display its content.

SEMESTER

S. No.	Course Code	Course Name	L	T	P	Credits	Types
1	MA102	Engineering Mathematics-II	3	1	0	4	GE4
2	PH102	Engineering Physics	3	1	0	4	GE5
3	IT102	Object-Oriented Programming	3	1	0	4	CC2/SEC
4	EC101	Basic Electronics Engineering	3	1	0	4	GE6
5	CS105	Introduction of Artificial Intelligence	2	0	0	2	CC3/FC
6	EN101	English Proficiency	2	0	0	2	OE2/AECC
7	CE103	Engineering Graphics Lab	1	0	2	2	GE-L3
8	PH104	Engineering Physics Lab	0	0	2	1	GE-L4
9	IT181	Object-Oriented Programming Lab	0	0	2	1	CC-L2/SEC
10	EC181	Basic Electronics Engineering	0	0	2	1	GE-L5
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**17**	**4**	**8**	**25**	

ENGINEERING MATHEMATICS – II			
Course Code:	**MA102**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**04**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**60**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			

COURSE OUTCOMES
At the end of the course the students should be able to:

ENGINEERING PHYSICS			
Course Code:	**PH102**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**04**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**60**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
COURSE OUTCOMES			
At the end of the course the students should be able to:			

OBJECT ORIENTED PROGRAMMING			
Course Code:	**IT102**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**04**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**60**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1.To understand Object oriented concepts like data abstraction, encapsulation, etc.

2. To solve the real world scenarios using top down approach.

3. To understand various Java programming constructs.

4.The objectives of the course are to have students identify and practice the object-oriented programming concepts and techniques

5. Practice the use of C++ classes and class libraries, arrays, vectors

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Creating simple programs using classes and objects in C++.

2. Implement Object Oriented Programming Concepts in C++

3. Develop applications using stream I/O and file I/O.

4. Implement simple graphical user interfaces.

5. Implement Object Oriented Programs using templates and exceptional handling concepts.

Unit I: Object-Oriented Programming

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history,Java features, Java streaming, Java and Internet, Java contribution to Internet: Java applets, security,portability; Java environment, Java library, Java program structure, Java program, Java Virtual Machine(JVM) architecture, Just In Time compiler (JIT), data type, variables and arrays, operators, controlstatements, object-oriented paradigms; abstraction, encapsulation, inheritance, polymorphism, Java classand OOP implementation

Unit II: Data Type, Operators and Control Statement

Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable,symbolic constant, type casting, arithmetic operator, relational operator, logical operator, assignmentoperator, increment and decrement operator, conditional operator, bitwise operator, ?: operator,arithmetic expressions, expressions, type conversions in expressions, mathematical functions, more datatypes: arrays, strings, vectors, wrappers classes, program control statements: decision making andbranching: if, if….else, else….if, else if ladder, switch, decision making and looping: while, do….while,for.

Unit III: Classes, Objects and Methods

Java class libraries, class fundamentals, object, methods, adding variables, add methods, creating objects,accessing class members, constructors, methods overloading, static members, nesting of methods,inheritance: extending a class, overriding methods, final variables and methods, final classes, finalizermethods, abstract methods and classes, visibility control, exception handling fundamental.

Unit IV: Interfaces and Packages

Interfaces, extending interfaces, implementing interfaces, interfaces references, accessing interfacevariable, creating queue interface, variable in interfaces, packages, finding a packages and classpath,package and member access, Java API package, system package, naming conventions, creating package,accessing a package, adding a class to a package, hiding classes,

Unit V: Multithreading and Applet Programming

Multithreading programming: creating threads, thread class and runnable interface extending the threadclass, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, threadpriority, synchronization, thread communication using notify(), wait(), and notify all(), appletprogramming : applet basic, applets architecture, a complete applet skeleton, building applets code,applets life cycle, creating a executable applet, designing a web page, applets tag, passing parameters toapplets, applets and HTML.

Text Books:

[1] E. Balagurusawamy,Programming with JAVA, Tata McGraw Hill.
[2] Herbert Schildt, JAVA Beginner‟s guide, Tata McGraw Hill.

Reference Books:

[3] Deitel &Deitel,Java How to Program, Prentice-Hall.
[4] The Complete Reference JAVA 2, Herbert Schildt, 5th and 7thEdition, Tata McGraw Hill.
[5] Ken Arnold, James Gosling, Addison, The Java Programming Language, Wesley.

INTRODUCTION TO ARTIFICIAL INTELLIGENCE			
Course Code:	**CS105**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**02**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**30**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To introduce students to artificial intelligence			
1. To introduce the concepts of artificial intelligence to students			
1. To familiarize students with flow of artificial intelligence projects			
1. To familiarize students with different domains and application areas of artificial intelligence			
1. To enable students in implementation of artificial intelligence projects			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
At the end of the course the students should be able to:			
1. Understand the artificial intelligence evolution			
1. Understand the significant concepts of artificial intelligence			
1. Understand the flow of artificial intelligence projects			
1. Aware of application areas of artificial intelligence			
1. Implement artificial intelligence projects			

UNIT 1 Introduction

Introduction to AI, Components of AI, Goals of AI, Types of AI, History of AI, Turing Test in AI, Advantages and Disadvantages of AI, Intelligence, Intelligent System, Role of IS, Comparison of various IS, Weak AI and Strong AI, Mind Body Problem in AI, Chinese Room Experiment in AI, Parallel and Distributed AI.

UNIT II Agents in AI

Intelligent Agents, Types of AI Agents, Simple Reflex Agent,Model-based reflex agent, Goal-based agents, Utility-based agent, Learning agent, Structure of an AI Agent, Agent Environment in AI, Examples of Agents, Knowledge Engineering, Knowledge Based System, Knowledge Engineering Techniques, Knowledge Engineering Principles, Knowledge Engineering Methodology.

UNIT III Searching Techniques and AI problems

Searching in AI, Search Algorithm Terminologies, Properties of Search Algorithms, Breadth-first search, Depth-first search, Best First Search, Tic-Tac Toe Problem, Water Jug problem, Chess Problem, Tower of Hanoi problem, Travelling Salesman problem, Monkey and Banana Problem, Magic Square.

UNIT IV Knowledge Representation

Knowledge Representation Definition, Declarative Knowledge, Procedural knowledge, Meta Knowledge, Heuristic Knowledge, Structural Knowledge, Inheritable Knowledge, Inferential Knowledge, Relational Knowledge, Explicit Knowledge, Tacit Knowledge, Uncertain Knowledge, Knowledge Storage, Relation between Knowledge and Intelligence, AI knowledge cycle.

UNIT V AI Techniques and applications

Introduction to Machine Learning, Introduction to Deep Learning, Introduction to Expert system, Introduction to Natural Language Processing, AI in future, AI in social Media, AI in Entertainment and education, AI in drones, AI in Automated Computer support, AI in personalized shopping experience, AI in Finance, AI in smart Cars, AI in travel and navigation, AI in smart home devices, AI in security and surveillance, Ai in education, AI in healthcare, AI in E commerce.

Text Books:

1. Artificial Intelligence, Elaine Reich: Tata Mcgraw Hill publishing house, 2008.
2. Artificial Intelligence, Ela Kumar, IK Publishing.
3. Artificial Intelligence, Peterson, TataMcGraw Hill, 2008.
4. Artificial Intelligence, Russel and Norvig, Pearson Printice Hall Publication, 2006.
5. Artificial Intelligence, Winston, PHI publication, 2006.
6. Artificial Intelligence- A modern approach (3rd Edition) By Stuart Russell & Peter Norvig.
7. Artificial Intelligence: The Basics By Kevin Warwick

OBJECT ORIENTED PROGRAMMING LAB			
Course Code:	**IT181**	**Course Credits:**	**1**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**02**	**Mid Sem. Exam Hours:**	**–**
Total No. of Lectures (L + T):	**15**	**End Sem. Exam Hours:**	**2**
COURSE OBJECTIVES			
1.To understand Object oriented concepts like data abstraction, encapsulation, etc.			
2. To solve the real world scenarios using top down approach.			
3. To understand various Java programming constructs.			
4.The objectives of the course are to have students identify and practice the object-oriented programming concepts and techniques			
5. Practice the use of C++ classes and class libraries, arrays, vectors			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Creating simple programs using classes and objects in C++.			
2. Implement Object Oriented Programming Concepts in C++			
3. Develop applications using stream I/O and file I/O.			
4. Implement simple graphical user interfaces.			
5. Implement Object Oriented Programs using templates and exceptional handling concepts.			

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments:

1. Write a separate Java Code to implement each of the following:
   Class, Command Line Argument, how to enter value through keyboard

2. Write a separate Java Code to implement each of the following data types:
   Variable, Constant, Arrays, Strings, Vectors, Wrappers Classes, Type Casting

3. Write a separate Java Code to implement each of the following operators:
   Arithmetic operator, Relational operator, Logical operator, Assignment operator, Increment& Decrement operator, Conditional operator, Bitwise operator, ?: operator

4. Write a separate Java Code to implement each of the following control statements:

Decision statement, Loops statement and Branch statements

5. Write a separate Java Code to implement each of the following sorting:
Bubble Sort, Selection Sort, Insertion Sort, Merge Sort

6. Write a separate Java Code to implement each of the following:
Class, Object, Constructors, Method, Method Overloading and Method Overriding

7. Write a separate Java Code to implement each of the following:
Final variable, final class, final method, abstract class, abstract method and concrete method

8. Write a separate Java Code to implement each of the following OOPs concepts:
Abstraction, Polymorphism, Encapsulation, Inheritance

9. Write a separate Java Code to implement each of the following:
Exception handling with Try, Catch, Throw, Throws, Finally
Multiple catch statement with the following exceptions:
ArithmeticException, ArrayOutOfBoundsException and ArrayStoreException

10. Write a separate Java Code to implement each of the following:Visibility Controls: Private, Public and Protected

11. Write a separate Java Code to implement each of the following:Interface, extending and implementing interface.

12. Write a separate Java Code to implement each of the following:
Multithreading: Create thread with thread class and runnable interface, thread priorities, synchronization

13. Write a separate Java Code to implement each of the following:
Packages: Create package A with following methods and import this package A into another Java program to show the result of methods of package A.
(i) First method: Factorial number with the help of recursion;
(ii) Second method: Fibonacci Series
(iii) Third Method: Generate first 10 prime numbers and show the sum of first 10prime numbers.

14. Write Java Code to generate the following output on applet with the help of two dimensional array and show the result with the help of HTML file.

```
7 14 21 28 35 42 49 56 63 70 Sum = 385
5 10 15 20 25 30 35 40 45 50 Sum = 275
3 6 9 12 15 18 21 24 27 30 Sum = 165
```

15. Write a Java Code to design the following web page with the help of applet and HTML.

SEMESTER III

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT201	Animation and Computer Graphics	3	0	0	3	CC4
2	IT203	Operating System	3	0	0	3	CC5
3	IT205	Software Engineering	3	0	0	3	CC6 / SEC
4	IT207	System Design & Analysis Techniques	3	0	0	3	CC7
5	IT209	Web Technologies	3	0	0	3	CC8
6	MA201	Engineering Mathematics-III	3	1	0	4	GE7
7	IT281	Animation & Computer Graphics Lab	0	0	3	2	CC-L3
8	IT283	Operating System Lab	0	0	3	2	CC-L4 / SEC
9	IT285	Web Technologies Lab I	0	0	3	2	CC-L5 / SEC
10	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

ANIMATION & COMPUTER GRAPHICS THEORY			
Course Code:	**IT 201**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
This course will enable students to:			
1. Understand the need of developing graphics application			
2. Learn algorithmic development of graphics primitives like: line, circle, polygon etc.			
3. Learn the representation and transformation of graphical images and pictures.			
4. To be familiar with image fundamentals and animations			
5. Applying efficient graphics technique to solve engineering problems			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Explain fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering.			

2. Explain the ideas in some fundamental algorithms for computer graphics and to some extent be able to compare and evaluate them.
3. Be able to Compare various graphics algorithm used in 2D and 3D
4. Be able to understand and identify the performance characteristics of graphics algorithms and animation.
5. Employ algorithms to model engineering problems, when appropriate.

UNIT I INTRODUCTION

Application areas of Computer Graphics, overview of graphics systems, video-display devices, raster-scan systems, random scan systems, graphics monitors and work stations and input devices, Output primitives : Points and lines, line drawing algorithms, mid-point circle and ellipse algorithms. Filled area primitives: Scan line polygon fill algorithm, boundary-fill and flood-fill algorithms.

UNIT II 2-D GEOMETRICAL TRANSFORMS

Translation, scaling, rotation, reflection and shear transformations, matrix representations and homogeneous coordinates, composite transforms, transformations between coordinate systems, 2-D viewing : The viewing pipeline, viewing coordinate reference frame, window to view-port coordinate transformation, viewing functions, Cohen-Sutherland and Cyrus-beck line clipping algorithms, Sutherland –Hodgeman polygon clipping algorithm.

UNIT III REPRESENTATION AND TRANSFORMATION

3-D object representation Polygon surfaces, quadric surfaces, spline representation, Hermite curve, Bezier curve and B-Spline curves, Bezier and B-Spline surfaces, basic illumination models, polygon rendering methods, 3-D Geometric transformations: Translation, rotation, scaling, reflection and shear transformations, composite transformations, 3-D viewing: Viewing pipeline, viewing coordinates, view volume and general projection transforms and clipping.

UNIT IV VISIBLE SURFACE DETECTION METHODS

Classification, back -face detection, depth-buffer, scan-line, depth sorting, BSP-tree methods, area sub-division and octree methods. Tools of Multimedia: Paint and Draw Applications, Graphic effects and techniques, Image File Format, Anti-aliasing, Morphing, Multimedia Authoring tools, professional development tools.

UNIT V COMPUTER ANIMATION

Introduction and Principles of Animations, Power of Motion, Animation Techniques, Animation File Format, Making animation for Rolling Ball, making animation for a Bouncing Ball, Animation for the web, GIF, Plugins and Players, Animation tools for World Wide Web. Design of animation sequence, general computer animation functions, raster animation, computer animation languages, key frame systems, motion specifications.

REFERENCE BOOKS:

1. Computer Graphics: principals and practice Foley, vanDam, Feiner Hughes Addision Wesley

2. Mathematical Elements of Graphics Roges Tata McGrow Hill

3. Computer Graphics Donald Hearn and M.Pauline Baker Prentice Hall India

4. Procedural Elements-Computer Graphics, David Rogers, TMH

5. Principals of Computer graphics, Shalini Govil-pal, springer

OPERATING SYSTEM			
Course Code:	**IT203**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Understanding how Operating Systems are Important for Computer Systems.
2 To make aware of different types of Operating Systems and their services.
3 To learn different process scheduling algorithms and synchronization techniques to achieve better performance of a computer system
4 To know virtual memory concepts and secondary memory management
5 Understanding of Security & protection in Operating System

COURSE OUTCOMES
At the end of the course the students should be able to:
1Understands the different services provided by Operating System at different level
2They learn real life applications of Operating System in every field.
3 Understands the use of different process scheduling algorithm and synchronization techniques to avoid deadlock.
They will learn different memory management techniques like paging, segmentation and demand paging etc.
5 Perform implementation of protection mechanisms in operating system

UNIT I INTRODUCTION TO OPERATING SYSTEM

Importance of operating systems, basic concepts and terminology about operating system, memory management, processor management, device management, information management functions.

UNIT II PROCESS MANAGEMENT

Elementary concept of process, job scheduler, process scheduling, operation on process, threads, overview, scheduling criteria, scheduling algorithms, algorithm evaluation process synchronization, synchronization hardware, semaphores, classical problem of synchronization, monitors and atomic transaction deadlocks: system model, deadlock characterization, deadlocks prevention, deadlocks avoidance, deadlocks detection, recovery from deadlock.

UNIT III MEMORY & STORAGE MANAGEMENT

Basic Memory Management: Definition, Logical and Physical address map, Memory allocation: Contiguous Memory allocation, partition, Fragmentation, Compaction, Paging, Segmentation.

Virtual Memory: Basics of virtual memory, Hardware and control structures-Locality of reference, Page fault, Demand paging, page replacement policies: First In First Out (FIFO), second chance (SC), Not recently used (NRU) and Least recently used (LRU).

UNIT IV UNIX/LINUX OPERATING SYSTEM: Development Of Unix/Linux, Role & Function Of Kernel, System Calls, Elementary Linux command & Shell Programming, Directory Structure, System Administration, Case study: Linux, Windows Operating System

UNIT V SECURITY & PROTECTION: Security Environment, Design Principles of Security, User authentication, Protection Mechanism: Protection Domain, Access Control List

Text Books:

[1]. Galvin, Wiley, Operating Systems Concepts, 8th edition, 2009.
[2]. James L Peterson, Operating Systems Concept, John Wiley & Sons Inc, the 6Rev edition, 2007.

Reference Books:

[3]. Deitel H. M., An Introduction to Operating Systems, Addison-Wesley, 1990.
[4]. Stallings William, Operating Systems, PHI, New Delhi, 1997.
[5]. S. Tanenbaum Modern Operating Systems, Pearson Education, 3rd edition, 2007.
[6]. Nutt, Operating System, Pearson Education, 2009.
[7]. S. Tanenbaum, Distributed Operating Systems, Prentice Hall, 2nd edition, 2007.

SOFTWARE ENGINEERING			
Course Code:	**IT 205**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
This course will enable students to:			
1 Knowledge of basic SW engineering methods and practices and application.			
2 A general understanding of software process models			
3 Understanding of software requirements and the SRS documents.			
4 Understanding of software design process.			
5 Understanding of software coding, testing and maintenance.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Basic knowledge and understanding of the analysis and design of complex systems.			
2 Ability to apply software engineering principles and techniques.			
3 Ability to design, develop, maintain and evaluate large-scale software systems.			
4 To produce efficient, reliable, robust and cost-effective software solutions.			
5 Ability to perform independent research and analysis.			

UNIT I SOFTWARE ENGINEERING

Introduction to software engineering: definitions, role of software engineering, planning a software project, defining the problem, developing a solution strategy, planning the development process, software engineering process paradigms, principles of software engineering, software engineering activities.

UNIT II REQUIREMENT ANALYSIS AND DESIGN

Software Requirement Specification (SRS): Introduction, need of SRS, significance, characteristics of SRS, Structure of SRS, IEEE standards for SRS design, functional and non-functional requirements, Requirement gathering and analysis, requirement engineering and management.

UNIT III SOFTWARE DESIGN PROCESS

Software Design: Introduction, design process activities: architectural design, Abstract specification, Interface design, component design, data structure design, algorithm design modular approach, top-down design, bottom up design, design methods: data-flow model: data flow diagram, entity-relation-attribute model: E-R diagram, structural model: structure charts, context diagrams, object models: use case modeling, use case diagrams, sequence diagrams, cohesion and coupling.

UNIT IV SOFTWARE LIFE CYCLE MODELS

Software Development Life Cycle (SDLC), SDLC models, waterfall model and its variations, prototype model, iterative enhancement model, spiral model, RAD model, comparison of these models, software development teams, software development environments, validation and traceability, maintenance, prototyping requirements, Software project management.

UNIT V SOFTWARE CODING, TESTING AND MAINTENANCE

Coding, Testing Methods: unit testing, integration testing, system testing, acceptance testing, testing techniques: white box testing, black box testing, thread testing, regression testing, alpha testing, beta testing, static testing, dynamic testing, Evolution of software products, economics of maintenance, category of software maintenance, Role of product development life cycle, deployment model, adaptive maintenance, corrective maintenance, perfective maintenance, enhancement request, proactive defect prevention, problem reporting, problem resolution, software maintenance from customers' perspective, maintenance standard: IEEE-1219, ISO-12207.

REFERENCE BOOKS:

1. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publishing House, New Delhi 1997.
2. Ian Sommerville, Software Engineering, Pearson Education, 2009.
3. Pressman Roger S., Software Engineering: Practitioner's Approach, McGraw-Hill Inc., 2004.
4. Software Engineering: Software Reliability, Testing and Quality Assurance, Nasib S. Gill, Khanna Book Publishing Co (P) Ltd., New Delhi, 2002.

SYSTEM DESIGN AND ANALYSIS TECHNIQUES			
Course Code:	**IT207**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1Explain what systems are and how they are developed.			
2 Identify and describe the phases of the systems development life cycle.			
3 Describe the role and responsibilities of the systems analyst in the development and management of systems.			
4 Use tools and techniques for process and data modeling.			
5 Develop and deliver a Requirements Definition Proposal for a new system in a well-structured business proposal			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Understand the basis for understanding the life cycle of a systems development project;			
2 Experience in developing information systems models			
3 Experience in developing systems project documentation			
4 Able to analyze business problems and develop a requirements° document, written in clear and concise business language.			
5 An understanding of the object-oriented methods models as covered by the Unified Modelling Language.			

UNIT-1

DATA AND INFORMATION

Types of information: operational, tactical, strategic and statutory, why do we need information systems, management structure, requirements of information at different levels of management, functional allocation of management, requirements of information for various functions, qualities of information, small case study.

UNIT-2

SYSTEMS ANALYSIS AND DESIGN LIFE CYCLE

Requirements determination, requirements specifications, feasibility analysis, final specifications, hardware and software study, system design, system implementation, system evaluation, system modification, role of systems analyst, attributes of a systems analyst, tools used in system analysis.

UNIT-3

INFORMATION GATHERING

Strategies, methods, case study, documenting study, system requirements specification. from narratives of requirements to classification of requirements as strategic, tactical, operational and statutory**.**

UNIT- 4
FEASIBILITY ANALYSIS

Deciding project goals. examining alternative solutions, cost benefit analysis, quantifications of costs and benefits, payback period, system proposal preparation for managements, parts and documentation of a proposal, tools for prototype creation.

UNIT-5
TOOLS FOR SYSTEMS ANALYSTS

Data flow diagrams, case study for use of DFD, good conventions, leveling of DFDs, leveling rules, logical and physical DFDs, software tools to create DFDs, decision tables for complex logical specifications, specification oriented design vs procedure oriented design

Text Books:

1. Elias M.Awad., System Analysis and Design.

2.Perry Edwards,System Analysis and Design.

WEB TECHNOLOGIES			
Course Code:	**IT209**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Explain the history of the internet and related internet concepts that are vital in understanding web development.
2 Discuss the insights of internet programming and implement complete application over the web
3 Demonstrate the important HTML tags for designing static pages and separate design from content using Cascading Style sheets.
4 Describe the concepts of client side scripting like JavaScript.
5 Understanding the concept of web hosting and seo

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Students are able to develop a dynamic webpage by the use of java script and HTML.
2 Understand the concept of JAVA SCRIPTS.
3 Able to build interactive web applications using CSS
4 Understand the basic concepts of websites.
5 To develop and deploy real time web applications in web servers

UNIT-1

Introduction: Basic principles involved in developing a web site, Planning process , Domains and Hosting, Responsive Web Designing , Types of Websites (Static and Dynamic Websites), Web Standards and W3C recommendations.

UNIT-2

Introduction to HTML: What is HTML, HTML Documents, Basic structure of an HTML document, creating an HTML document, Mark up Tags, Heading-Paragraphs, Line Breaks.
Elements of HTML: HTML Tags., Working with Text, Working with Lists, Tables and Frames, Working with Hyperlinks, Images and Multimedia, Working with Forms and controls.

UNIT-3

Concept of CSS: Creating Style Sheet, CSS Properties, CSS Styling (Background, Text Format, Controlling Fonts), Working with block elements and objects, Working with Lists and Tables, CSS Id and Class, Box Model (Introduction, Border properties, Padding Properties, Margin properties).

UNIT-4

Introduction to Client Side Scripting: Introduction to Java Script , Javascript Types , Variables in JS, Operators in JS , Conditions Statements , Java Script Loops, JS Popup Boxes , JS Events , JS Arrays, Working with Arrays, JS Objects, JS Functions, Using Java Script in Real time, Validation of Forms, Related Examples.

UNIT-5

Web Hosting: Web Hosting Basics, Types of Hosting Packages, Registering domains, Defining Name Servers, Using Control Panel, Creating Emails in Cpanel , Using FTP Client, Maintaining a Website.

Concepts of SEO: Basics of SEO, Importance of SEO, on-page Optimization Basics and off-page SEO.

Text Books:

1. Web Technolgies, Uttam Roy, OXFORD University press
2. Web programming with HTML, XHTML and CSS, 2e, Jon Duckett, Wiley India
3. Steven M. Schafer, “HTML, XHTML, and CSS Bible, 5ed”, Wiley India
4. Ian Pouncey, Richard York, “Beginning CSS: Cascading Style Sheets for Web Design”, Wiley India
5. SEO for Dummies, 6th Edition, by Peter Kent
6. Teach Yourself Javascript in 24 Hours, 5th Edition, by Ballard and Moncur

REFERENCE BOOKS:

1. Web programming Bai, Michael Ekedahl, CENGAGE Learning , India edition.
2. An Introduction to Web Design + Programming, Paul S.Wang, India Edition

ENGINEERING MATHEMATICS-III			
Course Code:	**MA201**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
COURSE OUTCOMES			
At the end of the course the students should be able to:			

ANIMATION & COMPUTER GRAPHICS LAB

Course Code:	**IT 281**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures(L + T): 10	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

This course will enable students to:

1.Learn algorithmic development of graphics primitives like: line, circle, polygon etc.

2. Learn the representation and transformation of graphical images and pictures.

3. Learn algorithmic development of graphics primitives like: line, circle, polygon etc

4. Learn the representation and transformation of graphical images and pictures.

5. To familiarize the students with various approaches, methods and techniques of Animation Technology.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Explain fundamental concepts within computer graphics such as geometrical transformations, illumination models, removal of hidden surfaces and rendering

2. Explain the ideas in some fundamental algorithms for computer graphics and to some extent be able to compare and evaluate them

3. Be able to Compare various graphics algorithm used in 2D and 3D.

4. To be familiar with image fundamentals and animations.

5. Employ algorithm to model engineering problems, when appropriate.

<u>List of Experiments:</u>

1. Procedure to create an animation to represent the growing moon.
2. Procedure to create an animation to indicate a ball bouncing on steps.
3. Procedure to simulate movement of a cloud.
4. Procedure to draw the fan blades and to give proper animation.
5. Procedure to display the background given (filename: tulip.jpg) through your name.
6. Procedure to display the background given (filename: garden.jpg) through your name using mask.
7. Procedure to create an animation with the following features.
WELCOME (Letters should appear one by one .The fill color of the text should change to a different colour after the display of the full word.)
8. Procedure to simulate a ball hitting another ball.
9. Procedure to design a visiting card containing at least one graphic and text information.
10. Procedure to take a photographic image. Give a title for the image. Put the border. Write yournames. Write the name of institution and place.

11. Procedure to prepare a cover page for the book in your subject area. Plan your own design.
12. Procedure to extract the flower only from given photographic image and organize it on a Background, Selecting your own background for organization.

Operating System Lab

Course Code:	**IT283**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):U	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures(L + T):10	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1 To learn about file management and different types of permission setup

2To understand how system processes work and how to manage them

3 To learn & implement different Operating system algorithm

4 Apply concepts of Deadlock and its prevention.

5 Apply concept of OS to develop Producer Consumer problem & real scenario problems

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Able to implement and analyze the performance of different algorithms of Operating Systems like CPU scheduling algorithm, page replacement algorithms, deadlock avoidance, detection algorithm and so on.

2 Able to design and develop a course project that can have a positive impact on the environment or society or mankind.

3 Demonstrate the various operations of file system.

4 Apply the various methods in memory allocation and page replacement algorithms.

5 Apply the process synchronous concept using message queue, shared memory, semaphore and Dekker’s algorithm for the given situation

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments:

1. Program for file handling.
2. Program for Dining Philosophers Problem.
3. Program for Producer – Consumer Problem concept.
4. Program for First Come First Serve Algorithm.
5. Program for Shortest Job First Scheduling Algorithm.

6. Program for Round Robin Scheduling Method.
7. Program for Priority Scheduling Algorithm.
8. Implement the concept of Fragmentation and Defragmentation.
9. Simulate Bankers Algorithm for Deadlock Avoidance
10. Simulate Bankers Algorithm for Deadlock Prevention

WEB TECHNOLOGIES LAB

Course Code:	**IT285**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3U**
No. of Labs + Tutorials (Hrs/Week):	**03+00**	**Mid Sem. Exam Hours:**	
Total No. of Labs (L) :	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1 Students will learn to analyze a web page and identify its elements and attributes

2 Create web pages using XHTML and Cascading Style Sheets

3 Students will learn the ordered list and unordered list

4 Student will learn the inline CSS, internal CSS and external CSS

5 Students will learn that how to create the login form using HTML.

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Learn the how to display attributes and their values using HTML

2 Learn the how to embed images in a web pages

3 Learn to create the Registration form, login form.

4 Learn the Inline CSS, Internal CSS, External CSS

5 Learn the how to create the websites.

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments: In this lab programs related to HTML etc. are suggested.

1. Write an HTML code to display your education details.

2. Write an HTML code to display attributes and their values for HTML elements.

3. Write an HTML code to create a web page having website links.

4. Write HTML code to embed images in a web page.

5. Write an HTML code to create a Registration Form. On submitting the form, the user should be asked to login with this new credentials.

6. Write an HTML code to create a login form. On submitting the form, the user should get navigated to a profile page.

7. Write an HTML code to illustrate the usage of the following: • Ordered List • Unordered List

8. Write code to create a frameset having header, navigation and content sections.

9. Write code to create your Institute website, Department Website.

10. Write code to demonstrate the usage of inline CSS.

11. Write code to demonstrate the usage of internal CSS.

12. Write code to demonstrate the usage of external CSS.

13. Make a form for keeping student record and validate it.

14. Write program to design an entry form of student details.

SEMESTER IV

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT202	Data Structure	3	0	0	3	CC9
2	IT204	Database Management System	3	0	0	3	CC10
3	IT206	Web Development(PHP)	3	0	0	3	CC11 / SEC
4	IT208	Discrete Structure	3	0	0	3	CC12
5	IT210	Fundamentals of Digital Electronics Circuits	3	0	0	3	CC8
6	IT212	Digital Communication & Coding	3	1	0	4	GE7
7	IT282	Data Structures	0	0	3	2	CC-L3
8	IT284	Database Management System Lab	0	0	3	2	CC-L4 / SEC
9	IT286	Web Development PHP lab	0	0	3	2	CC-L5 / SEC
10	IT288	Digital Electronics Circuits Lab	0	0	3	2	CC-L9
11	GP	General Proficiency	Non Credit				
		Total Hours and Credits	**18**	**1**	**9**	**25**	

<table>
<tr><th colspan="4">DATA STRUCTURES</th></tr>
<tr><td colspan="4"></td></tr>
<tr><td><b>Course Code:</b></td><td><b>IT202</b></td><td><b>Course Credits:</b></td><td><b>3</b></td></tr>
<tr><td><b>Course Category:</b></td><td><b>CC</b></td><td><b>Course (U / P)</b></td><td><b>U</b></td></tr>
<tr><td><b>Course Year (U / P):</b></td><td><b>2U</b></td><td><b>Course Semester (U / P):</b></td><td><b>3 U</b></td></tr>
<tr><td><b>No. of Lectures + Tutorials (Hrs/Week):3</b></td><td><b>03 + 00</b></td><td><b>Mid Sem. Exam Hours:</b></td><td><b>1</b></td></tr>
<tr><td><b>Total No. of Lectures (L + T):45</b></td><td><b>45 + 00</b></td><td><b>End Sem. Exam Hours:</b></td><td><b>3</b></td></tr>
<tr><td colspan="4"></td></tr>
<tr><th colspan="4">COURSE OBJECTIVES</th></tr>
<tr><td colspan="4">1.To emphasize the importance of appropriate data structure in developing and implementing efficient algorithms</td></tr>
<tr><td colspan="4">2.Understand basic data structures such as arrays, stacks, queues, hash tables and linked list</td></tr>
<tr><td colspan="4">3.To analyze the asymptotic performance of various algorithms</td></tr>
<tr><td colspan="4">4.Solve problems using graphs, trees and heaps</td></tr>
<tr><td colspan="4">5.Apply important algorithmic design paradigms and methods of analysis</td></tr>
<tr><td colspan="4"></td></tr>
<tr><th colspan="4">COURSE OUTCOMES</th></tr>
<tr><td colspan="4">At the end of the course the students should be able to:</td></tr>
<tr><td colspan="4">1.Define basic static and dynamic data structures and relevant standard algorithms for them.</td></tr>
<tr><td colspan="4">2.Select basic data structures and algorithms for autonomous realization of simple programs or program parts.</td></tr>
<tr><td colspan="4">3.Determine and demonstrate bugs in program, recognise needed basic operations with data structures</td></tr>
<tr><td colspan="4">4.Formulate new solutions for programming problems or improve existing code using learned algorithms and data structures</td></tr>
<tr><td colspan="4">5.Evaluate algorithms and data structures in terms of time and memory complexity of basic operations.</td></tr>
</table>

UNIT I INTRODUCTION TO DATA STRUCTURES: Abstract data types, sequences as value definitions, data types in C, pointers in C, data structures and C, arrays in C, array as ADT, one dimensional array, Implementing one dimensional array, array as parameters, two dimensional array, structures in C, implementing structures, Unions in C, implementation of unions, structure parameters, allocation of storage and scope of variables, recursive definition and processes: factorial function, fibonacci sequence, recursion in C, efficiency of recursion, hashing: hash function, open hashing, closed hashing: linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNITII STACK, QUEUE AND LINKED LIST:
Stack definition and examples, primitive operations, example -representing stacks in C, push and pop operation implementation, queue as ADT, C Implementation of queues, insert operation, priority queue, array implementation of priority queue, inserting and removing nodes from a list-linked implementation of stack, queue and priority queue, other list structures, circular lists: stack and queue as circular list - primitive operations on circular lists, header nodes, doubly

linked lists, addition of long positive integers on circular and doubly linked list.

UNIT III TREES:
Binary trees: operations on binary trees, applications of binary trees, binary tree representation, node representation of binary trees, implicit array representation of binary tree, binary tree traversal in C, threaded binary tree, representing list as binary tree, finding the Kth element, deleting an element, trees and their applications: C representation of trees, tree traversals, evaluating an expression tree,
constructing tree.

UNIT IV SORTING AND SEARCHING:
General background of sorting: efficiency considerations, notations, efficiency of sorting, exchange sorts: bubble sort; quick sort; selection sort; binary tree sort; heap sort, heap as a priority queue, sorting using a heap, heap sort procedure, insertion sorts: simple insertion, shell sort, address calculation sort, merge sort, radix sort, sequential search: indexed sequential search, binary search, interpolation search.

UNIT V GRAPHS:
Application of graph, C representation of graphs, transitive closure, Warshall's algorithm, shortest path algorithm, linked representation of graphs, Dijkstra's algorithm, graph traversal, traversal methods for graphs, spanning forests, undirected graph and their traversals, depth first traversal, application of depth first traversal, efficiency of depth first traversal, breadth first traversal, minimum spanning tree, Kruskal's algorithm, round robin algorithm.

Text Books:
[1]. Aaron M. Tenenbaum, Yeedidyah Langsam, Moshe J. Augenstein, 'Data structures using C', Pearson Education, 2004 / PHI.
References Books:
[2]. E. Balagurusamy, 'Programming in Ansi C', 2nd Edition, TMH, 2003.
[3]. Robert L. Kruse, Bruce P. Leung Clovis L.Tondo, 'Data Structures and Program Design in C',
Pearson Education, 2000 / PHI.

DATABASE MANAGEMENT SYSTEM

Course Code:	**IT204**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1.Describe the fundamental elements of relational database management systems

2.Explain the basic concepts of relational data model, entity-relationship model, relational database design, relational algebra and SQL.

3.Design ER-models to represent simple database application scenarios

4.Convert the ER-model to relational tables, populate relational database and formulate SQL queries on data.

5.Improve the database design by normalization.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Understand of database concepts and thorough knowledge of database software's.

2.Model an application's data requirements using ER diagrams

3. Write SQL commands to create tables and query data in a relational DBMS

4. Execute various advanced SQL queries related to transactions, concurrency

5.Explain the principle of transaction management design.

UNIT I DATABASE SYSTEM

Database system vs. file system, view of data, data abstraction, instances and schemas, data models, ER model, relational model, database languages, DDL, DML, database access for applications programs, database users and administrator, transaction management, database system structure, storage manager, query processor, history of database systems, database design and ER diagrams, beyond ER design entities, attributes and entity sets, relationships and relationship sets, additional features of ER model, concept design with the ER model, and conceptual design for large enterprises.

UNIT II RELATIONAL MODEL

Introduction to the relational model, integrity constraint over relations, enforcing integrity constraints, querying relational data, and logical database design, destroying /altering tables and views. relational algebra and calculus: relational algebra, selection and projection set operations, renaming, joins, division, relational calculus, tuple relational calculus, domain relational calculus, expressive power of algebra and calculus.

UNIT III BASIC SQL QUERY

Examples of basic SQL queries, nested queries, correlated nested queries set, comparison operators, aggregative operators, NULL values, comparison using null values, logical connectivity, AND, OR and NOTR, impact on SQL constructs, outer joins, disallowing NULL values, complex integrity constraints in SQL triggers and active databases.

UNIT IV SCHEMA REFINEMENT

Problems caused by redundancy, decompositions, problem related to decomposition, reasoning about FDS, FIRST, SECOND, THIRD normal form, BCNF, fourth normal form, lossless join decomposition, dependency preserving decomposition, schema refinement in database design, multi valued dependencies.

UNIT V OVERVIEW OF TRANSACTION MANAGEMENT

ACID properties, transactions and schedules, concurrent execution of transaction, lock based concurrency control, performance locking, and transaction support in SQL, crash recovery, concurrency control, Serializability and recoverability, lock management, lock conversions, dealing with deadlocks, specialized locking techniques, concurrency without locking, crash recovery: ARIES, log, other recovery related structures, the write, ahead log protocol, check pointing, recovering from a system crash, media recovery, other approaches and interaction with concurrency control.

Text Books:

[1] Elmasri Navrate, Database Management System, Pearson Education, 2008.

[2] Raghurama Krishnan, Johannes Gehrke, Database Management Systems, TMH, 3rd edition, 2008.

References Books:

[3] C. J. Date, Introduction to Database Systems, Pearson Education, 2009.

[4] Silberschatz, Korth, Database System Concepts, McGraw hill, 5th edition, 2005.

[5] Rob, Coronel & Thomson, Database Systems Design: Implementation and Management, 2009.

WEB DEVELOPMENT (PHP)				
Course Code:	**IT206**	**Course Credits:**	**3**	
Course Category:	**CC**	**Course (U / P)**	**U**	
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**	
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**	
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**	
COURSE OBJECTIVES				
1.Describe fundamentals of web				
2. Introduce the creation of static webpage using HTML				
3. Describe the function of JavaScript as a dynamic webpage creating tool				
4. Outline the principles behind using MySQL as a backend DBMS with PHP				
5.Describe the importance of CSS in web development				
COURSE OUTCOMES				
At the end of the course the students should be able to:				
1.Learn and use DHTML and AJAX. Learn the basics of JQuery.				
2. Learn about the major vulnerabilities facing web sites and some simple ways to reduce their likelihood				
3.Use a MySQL database with PHP to create database applications				
4.Design HTML pages and use basic JavaScript code to enhance the pages				
5.Develop a complete market-ready database-driven website with PHP and JavaScript and go through the basic phases of the software life cycle				

UNIT I INTRODUCTION

Internet Standards, Introduction to WWW, WWW Architecture, client and server, web server, web application basic pieces, working of a website, Internet Protocols, Overview of HTTP, HTTP request – response, Generations of dynamic web pages, Front end and backend web development, web content management systems: Wordpress, Joomla, web development life cycle, Guidelines for Indian Government websites.

UNIT II BASICS OF HTML,CSS, JAVASCIPT

HTML and HTML5: Introduction, TML Tags, Formatting and Fonts, Commenting Code, Anchors, Backgrounds, Images, Hyperlinks, Lists, Tables, Frames, HTML Forms. Cascading Style Sheet (CSS): Introduction, Basics of CSS, style types. JavaScript: Introduction, variables, operators, conditionals, looping and validation. Introduction to Jquery, Ajax and XML.

UNIT III INTRODUCTION TO PHP

PHP structure: basic syntax, variables, operators, multiline commands. Expression and control flow in PHP, PHP dynamic linking. PHP functions and Objects, PHP arrays, Practical PHP: Date and time functions, file handling, system calls. Accessing and manipulating databases using PHP, Error handling in PHP, generating images with PHP. Cookies, sessions and authentication.

UNIT IV INTRODUCTION TO FRAMEWORK

Introduction of MVC pattern models, MVC works, Configuration Codelgniter, setting up Codelgniter with apache, Environment eg. Enable mod_rewrite, Fetching data, saving and updating data, Deleting data, user defined function in model, Data Validation, controller function, interacting with views, controller variables and parameters, Redirection, Getting post data, working with configuration layout, creating custom layout, Element and helpers, storing data in cake session, Reading a session data, Delete data from session

UNIT V MYSQL

Databases, Tables, Columns, MySQL Data Type, SELECT, UPDATE and DELETE Statements,PHP and MySQL: Connecting from PHP to MySQL Database, Executing SQL Queries from PHP.

Text Book

1.Learning PHP, MySQL & JavaScript with JQUERY, CSS & HTML5: Robin Nixon (O’Reilly)
2.Learning Web Design: A Beginner’s Guide to (X)HTML, Style Sheets and Web
Graphics:
Jennifer Niederst Robbins (O’Reilly).

Reference Books:

4. E. Balaguruswamy, Programming with JAVA, Tata McGraw Hill, 1998.
5. Christian Nagel, Professional C# and .NET 4, Wrox, 2010.
6. Karli Watson, Beginning Microsoft Visual C#, Wrox, 2

DISCRETE MATHEMATICS			
Course Code:	**IT208**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.			
2.Express a logic sentence in terms of predicates, quantifiers, and logical connectives			
3.Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.			
4.Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.			
5.Apply rules of inference, tests for validity, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. To express a logic sentence in terms of predicates, quantifiers, and logical connectives.			
2. Apply the rules of inference, proof by contradiction, and mathematical induction.			
3. Students will be able to evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.			
4.Students will be able to learn about predicates, quantifiers, and logical connectives			
5. Student will be able to use tree and graph algorithms to solve problems.			

UNIT I MATHEMATICAL LOGIC

Statements and notations, connectives, well formed formulas, truth tables, tautology, equivalence implication, normal forms, predicates: predicate logic, free & bound variables, rules of inference, consistency, proof of contradiction, automatic theorem proving.

UNIT II SET THEORY

Properties of binary relations, equivalence, compatibility and partial ordering relations, hasse diagram. functions: inverse function comports of functions, recursive functions, lattice and its properties, pigeon hole principles and its application, algebraic structures: algebraic systems examples and general properties, semi groups and monads, groups sub groups' homomorphism, isomorphism.

UNIT III ELEMENTARY COMBINATORICS

Basis of counting, combinations & permutations, with repetitions, constrained repetitions, binomial coefficients, binomial multinomial theorems, the principles of inclusion – exclusion.

UNIT IV RECURRENCE RELATION

Generating functions, function of sequences calculating coefficient of generating function, recurrence relations, solving recurrence relation by substitution and generating funds, characteristics roots solution of inhomogeneous recurrence relation.

UNIT V GRAPH THEORY

Representation of graph, DFS, BFS, spanning trees, planar graphs. graph theory and applications, basic concepts isomorphism and sub graphs, multi graphs and euler circuits, hamiltonian graphs, chromatic numbers

Text Books:

[1].Ralph. P.Grimaldi, Discrete and Combinational Mathematics- An Applied Introduction-5th Edition, Pearson Education

[2].Trembly J.P. & ManoharP. Discrete Mathematical Structures with applications to computer science, TMH

[3].Kenneth H. Rosen, Discrete Mathematics and its Applications, Fifth Edition.TMH.

Reference Books:

[4].Thomas Koshy, Discrete Mathematics with Applications, , Elsevier

[5].Bernand Kolman, Roberty C. Busby, Sharn Cutter Ross,Discrete Mathematical Structures, Pearson Education/PHI.

[6]. Garry Haggard and others, Discrete Mathematics for Computer science,, Thomson.

[7] J.L. Mott, A. Kandel, T.P. Baker,Discrete Mathematics for Computer Scientists & Mathematicians, Prentice Hall.

FUNDAMENTALS OF DIGITAL ELECTRONICS CIRCUITS			
Course Code:	**IT210**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**3 U**
No. of Lectures + Tutorials (Hrs/Week):3	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):45	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Simplify and evaluate basic logic statements including compound statements, implications, inverses, converses, and contrapositives using truth tables and the properties of logic.			
2.Express a logic sentence in terms of predicates, quantifiers, and logical connectives			
3.Apply the operations of sets and use Venn diagrams to solve applied problems; solve problems using the principle of inclusion-exclusion.			
4.Determine the domain and range of a discrete or non-discrete function, graph functions, identify one-to-one functions, perform the composition of functions, find and/or graph the inverse of a function, and apply the properties of functions to application problems.			
5.Apply rules of inference, tests for validity, proof by contradiction, proof by cases, and mathematical induction and write proofs using symbolic logic and Boolean Algebra.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. To express a logic sentence in terms of predicates, quantifiers, and logical connectives.			
2. Apply the rules of inference, proof by contradiction, and mathematical induction.			
3. Students will be able to evaluate Boolean functions and simplify expressions using the properties of Boolean algebra.			
4.Students will be able to learn about predicates, quantifiers, and logical connectives			
5. Student will be able to use tree and graph algorithms to solve problems.			

UNIT I

Number systems & codes, Binary arithmetic Boolean algebra and switching function. Minimization of switching function, concept of prime implecant etc. Karnaugh's map method, Quine & McCluskey's method, cases with don't care terms and multiple outputs switching function. Logic gates, NAND, NOR realization of switching function; half-adder half-subtractor full-adders full-subtractor circuits. Series & parallel addition and BCD adders, look-ahead carry generator.

UNIT II

Linear wave shaping circuits, Bistable, monostable & astable multivibrators, Schmitt trigger circuits .Introduction to D/A converters. Various types of Analog to Digital & Digital to Analog converters sample & hold circuits and V-F converters.

UNIT III

Logic families: RTL, DTL, all types of TTL circuits, ECL, 12 L and PMOS, NMOS & CMOS logic etc. Gated flip-flops and gated multivibrators etc; Interfacing between TTL to MOS, vice-versa.

UNIT IV

Introduction to shift registers / ring counters synchronous & asynchronous counters and designing of combinational circuits like code converters & counters etc.

UNIT V

Semiconductor memories & designing with ROM and PLA: Decoders Encoders multiplexers & demultiplexers**.**

Text Books:

[1] Tocci, "Digital Systems Principles & Applications".
[2] M. Mano, "Digital Logic & Computer Design", (PHI).
[3] Dr. A K Gautam, Digital Electronics, Khanna Publication

Reference Books:

[1] John F. Wakerly, Digital Design: Principles & Practices, Pearson Education.2003
[2] Richard F.Tinder, Engineering Digital Design, 2/e, Harcourt India Private Ltd., 2001
[3] William I. Fletcher, An Engineering Approach to Digital Design, Pearson Education
[4] William H.Gothmann, Digital Electronics: An Introduction to Theory and Practice, Eastern Economy Edition, Prentice-Hall of India Private Limited, New Delhi. 2001.
[5] Jacob Millman & Herbert Taub,Pulse,Digitaland Switching Waveforms,13th Reprint,Tata McGraw Hill Publishing Company Ltd., 1999

DIGITAL COMMUNICATION & CODING

Course Code:	**IT212**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**1U**
No. of Lectures + Tutorials (Hrs. /Week):	**02+00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**30**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1 To understand the building blocks of a digital communication system.

2 To prepare a mathematical background for communication signal analysis.

3 To understand and analyze the signal flow in a digital communication system.

4 · To analyze error performance of a digital communication system in presence of noise and other interference.

5 To understand the concept of a spread spectrum communication system.

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Understand the working of waveform coding techniques and analyze their performance.

2 Analyze the performance of a baseband and passband digital communication system in terms of error rate and spectral efficiency.

3.Perform the time and frequency domain analysis of the signals in a digital communication system.

4 Design of digital communication system

5 Understand the working of a spread spectrum communication system and analyze its performance.

Unit I:

Sampling of Signal, Sampling Theorem for Low Pass and Band Pass Signals, Aliasing, Pulse Amplitude Modulation, Time Division Multiplexing, Channel Bandwidth for PAM-TDM Signal, Types of Sampling, Instantaneous, Aperture Effect,

Unit II

Pulse Code Modulation: Quantization, Quantization Error, Pulse Code Modulation, Companding, Data Rate and Bandwidth of Multiplexed PCM Signal, Inter symbol Interference, Eye Diagram, Line Coding ,Differential PCM, Delta Modulation, Granular Noise ,Comparison of various system in terms of Bandwidth and Signal To-Noise Ratio.

Unit III:

Digital Modulation Techniques:- Analysis, Generation and Detection , Spectrum and Bandwidth of Amplitude Shift Keying, Binary Phase Shift Keying, Differential Phase Shift Keying , Quadrature Phase Shift Keying.

Unit IV:

Probability of error, bit error rate, Comparison of digital modulation techniques on the basis of probability of error, Matched Filter.

Unit V:

Line Coding: Unipolar RZ and NRZ, Bipolar RZ and NRZ, AMI, Split Phase etc. Properties for the selection of Line Codes, HDB Signaling , B8ZS Signaling, Inter-symbol Interference, Nyquist Criteria for Zero ISI

DATA STRUCTURE LAB			
Course Code:	**IT284**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Lectures + Tutorials (Hrs. /Week):	**03+00**	**Mid Sem. Exam Hours:**	**–**
Total No. of Lectures (L + T):	**10**	**End Sem. Exam Hours:**	**2**
COURSE OBJECTIVES			
1. Introduce the concept of data structures through ADT including List, Stack, Queues .			
1. To design and implement various data structure algorithms.			
1. To introduce various techniques for representation of the data in the real world.			
1. To develop application using data structure algorithms			
1. Compute the complexity of various algorithms.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Select appropriate data structures as applied to specified problem definition			
1. Implement operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.			
1. Students will be able to implement Linear and Non-Linear data structures.			
1. Implement appropriate sorting/searching technique for given problem.			
1. Design advanced data structure using Non-Linear data structure			

NOTE: Suggested list of experiments but not limited to these only.
List of Experiments:

1. Run time analysis of Fibonacci Series

2 Study and Application of various data Structure

3. Study and Implementation of Array Based Program

    a. Searching (Linear Search, Binary Search)

    b. Sorting (Bubble, Insertion, Selection, Quick, Merge etc)

    c. Merging

4. Implementation of Link List

    a. Creation of Singly link list, Doubly Linked list
    b. Concatenation of Link list
    c. Insertion and Deletion of node in link list
    d. Splitting the link list into two link list

5. Implementation of STACK and QUEUE with the help of

    a. Array
    b. Link List

6. Implementation of Binary Tree, Binary Search Tree, Height Balance Tree

7. Write a program to simulate various traversing Technique

8. Representation and Implementation of Graph

    a. Depth First Search
    b. Breadth First Search
    c. Prims Algorithm
    d. Kruskal's Algorithms

9. Implementation of Hash Table.

DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**IT284**	**COURSE CREDITS:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Labs(Hrs/Week):	**2(3 hrs)**		
Total No. of Labs	**10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Explain basic database concepts, applications, data models, schemas and instances.			
2.Demonstrate the use of constraints and relational algebra operations.			
3.Emphasize the importance of normalization in databases.			
4.Facilitate students in Database design			
5.Familiarize issues of concurrency control and transaction management.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1.Students get practical knowledge on designing and creating relational database systems.			
2.Understand various advanced queries execution such as relational constraints, joins, set operations, aggregate functions, trigger, views and embedded SQL.			
3.Design a commercial relational database system (Oracle, MySQL) by writing SQL using the system			
4.Use the basics of SQL and construct queries using SQL in database creation and interaction.			
5.Analyze and Select storage and recovery techniques of database system.			

NOTE: Suggested list of experiments but not limited to these only.
List of Experiments:

1. Introduction to MySQL, an exercise of data types in MySQL & Data Definition Language Commands
2. Exercise on Data Manipulation Language and Transaction Control Commands
3. Exercise on Types of Data Constraints
4. Exercise on JOINS (Single-Table) Using Normalization
5. Exercise on JOINS (Multiple-Table) Using Normalization
6. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic
7. Exercise on different Functions (Aggregate, Math and String)
8. Exercise on different types of sub queries

9. Procedures
10. View
11. Triggers

WEB DEVELOPMENT USING PHP LAB			
Course Code:	**IT286**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Labs (Hrs/Week):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. Understand best technologies for solving web client/server problems using PHP			
2. Analyze & design real time web applications			
3. Use PHP for dynamic effects and to validate form input entry			
4.Analyze & Develop to Use appropriate client-side or Server-side applications			
5. To develop and deploy real time web applications in web servers and in the cloud			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Develop a dynamic webpage by the use of java script and DHTML.			
2. Write a well formed / valid XML document using PHP			
3. Connect a java program to a DBMS and perform insert, update and delete operations on DBMS table using PHP.			
4. Draft a server side application called Servlet to catch form data sent from client, process it and store it on database using PHP			
5. Create a server side application to catch form data sent from client and store it on database using PHP			

NOTE: Suggested list of experiments but not limited to these only.

Note: Experiments of this lab will be based on Implementation through PHP/ .NET.

List of Experiments

1. WAP to demonstrate the string handling.
2. WAP to demonstrate array handling.
3. WAP to demonstrate the form handling.
4. WAP to demonstrate the file handling and uploading.

5. WAP to demonstrate the exception handling.

6. WAP to demonstrate the cookie handling and session handling.

7. WAP to demonstrate the E-mail sending.

8. WAP to demonstrate the database connectivity (MS-Access, Sql Server, MySQL).

9. WAP to demonstrate the use of filter in PHP.

10. WAP to demonstrate the OOPs concepts.

11. WAP to create a login page and authenticate login credentials with the backend.

12. Design a web page using PHP and host it to a hosting server (may be used as a hostinger server).

DIGITAL ELECTRONIC CIRCUITS LAB			
Course Code:	**IT288**	**Course Credits:**	**2**
Course Category:	**CC-P**	**Course (U / P)**	**U**
Course Year (U / P):	**2U**	**Course Semester (U / P):**	**4U**
No. of Labs (Hrs/Week):	**02(3 hrs)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To acquire the basic knowledge of digital logic levels and application of knowledge to understand digital electronics circuits.
2. To prepare students to perform the analysis and design of various digital electronic circuits.
3. To develop skill to build, and troubleshoot digital circuits.
4. An ability to construct, analyze, and troubleshoot simple sequential circuits.
5. An ability to design and troubleshoot a simple state machine.

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Have a thorough understanding of the fundamental concepts and techniques used in digital electronics
2. To understand and examine the structure of various number systems and its application in digital design.
3. The ability to understand, analyze and design various combinational and sequential circuits.
4.Ability to identify basic requirements for a design application and propose a cost effective solution.
5. The ability to identify and prevent various hazards and timing problems in a digital design

NOTE: Suggested list of experiments but not limited to these only.

List of Experiments:

1. To verify the De-Morgan’s theorems using NAND/NOR gates.
2. To design the full adder and half adder using AND, OR and X- OR gates.
3. To implement the logic circuits using decoder.
4. To implement the logic circuits using multiplexer.
5. To design parity generator and checker circuits.
6. To design and implement RS FLIP FLOP using basic latches.
7. Realization and testing of basic logic gates using discrete components.
8. Realization and testing of CMOS IC characteristics.

9. Realization and testing of TTL IC characteristics.

10. Realization and testing of RAM circuit using IC 7489.

11. Realization and testing of Interfacing of CMOS- TTL and TTL- CMOS ICS.

SEMESTER V

SEMESTER V

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT301	Theory of Automata	3	0	0	3	CC11
2	IT303	Computer Networks	3	0	0	3	CC12
3	IT305	Compiler Design	3	1	0	4	CC13
4	IT307	Soft Computing Techniques	3	0	0	3	CC14 / SEC
5	IT 315	Elective 1	3	0	0	3	E1 / DSE
6	IT 319	Elective 2	3	0	0	3	E2 / DSE
7	IT381	Computer Networks Lab	0	0	3	2	CC-L9
8	IT383	Compiler Design Lab	0	0	3	2	CC-L10
9	IT385	Soft Computing Techniques Lab	0	0	3	2	CC-L11 / SEC
10	GP	General Proficiency	Non Credit				

Theory of Automata

Course Code	**IT301**	**Course Credit**	**03**
Course Category	**CC**	**Course(U/P)**	**U**
No of Lectures + Tutorials(Hrs./Week)	**03+00**	**Mid Semester Exam Hours:**	**01**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
Total no of Lectures(L+T)	**45+00**	**End Term Exam Hours:**	**03**

COURSE OBJECTIVES

1. The objective of this course is to introduce students to the foundation of computability theory.

2. Application of mathematical techniques and logical reasoning to important problem.

3. Develop a strong background in reasoning about finite state automata and formal language.

4. This course is to explore the theoretical foundations of computer science from the perspective of formal language and classify machines by their power to recognize languages.

5. the basic theory of computer science and formal methods of computation like automation theory, formal language, grammars, Turing machine

Course Outcomes

At the end of the course the student should be able to understand the :

1. Under the basic property of regular grammar and design automata

2.Language accepted by an automata i.e. DFA(Deterministic Finite Automata)/NDFA(Non deterministic finite automata).

3. Understand the regular expression(RE) ,Kleen closure ,positive closure, RE to FA and FA to RE

4.Closure property of different language and Decidability /Undesirability property of different languages.

5.Define the various categories of language grammars in the Chomsky hierarchy and variants of Turing machine

UNIT I AUTOMATA

Introduction; alphabets, strings and languages; automata and grammars, deterministic finite automata (DFA)-formal definition, simplified notation: state transition graph, transition table, language of DFA, Nondeterministic finite Automata (NFA), NFA with epsilon transition, language of NFA, equivalence of NFA and DFA, minimization of finite automata, distinguishing one string from other, Myhill-Nerode Theorem

UNIT II REGULAR EXPRESSIONS AND LANGUAGES

Regular expression (RE), definition, operators of regular expression and their precedence, algebraic laws for regular expressions, Kleen‟s theorem, regular expression to FA, DFA to regular expression, arden theorem, non-regular languages, pumping lemma for regular languages. Application of pumping lemma, closure properties of regular languages, decision properties of regular languages, FA with output: moore and mealy machine, equivalence of moore and mealy machine, applications and limitation of FA.

UNIT III CONTEXT-FREE GRAMMAR AND LANGUAGES

Context Free Grammar (CFG) and Context Free Languages (CFL): definition, examples, derivation, derivation trees, ambiguity in grammar, inherent ambiguity, ambiguous to unambiguous CFG, useless symbols, simplification of CFGs, normal forms for CFGs: CNF and GNF, closure properties of CFLs, decision properties of CFLs: emptiness, finiteness and membership, pumping lemma for CFLs.

UNIT IV PUSH DOWN AUTOMATA

Push Down Automata (PDA): description and definition, instantaneous description, language of PDA, acceptance by final state, acceptance by empty stack, deterministic PDA, equivalence of PDA and CFG, CFG to PDA and PDA to CFG, two stack PDA

UNIT V TURING MACHINES (TM)

Basic model, definition and representation, instantaneous description, language acceptance by TM, variants of turing machine, TM as computer of integer functions, universal TM, church"s thesis recursive and recursively enumerable languages, halting problem, introduction to undecidability, undecidable problems about TMs. Post Correspondence Problem (PCP), modified PCP, introduction to recursive function theory.

Text Books:

1. Hopcroft, Ullman, "Introduction to Automata Theory, Languages and Computation", Pearson Education
2. K.L.P. Mishra and N.Chandrasekaran, "Theory of Computer Science : Automata, Languages and Computation", PHI

References Books:

3. Martin J. C., "Introduction to Languages and Theory of Computations", TMH
4. Papadimitrou, C. and Lewis, C.L., "Elements of the Theory of Computation", PHI

COMPUTER NETWORKS

Course Code:	**IT 303**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester**	**5P**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. Understanding of computer networks and its components
2. A general understanding of switching and OSI layers
3. Understanding of concept of congestion in the network
4. Understanding of protocols used in computer networks
5. Understanding of addressing in the computer network

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Understand network scenario
2. Understand OSI and TCP/IP layers
3. Understand the concept of congestion in the network
4. Understand various protocols used in each layer
5. Able to create network with the use of IP address

UNIT I INTRODUCTION AND PHYSICAL LAYER

Key concepts of computer network, transmission media, network devices, network topology, topology design issues, types of networks: LAN, MAN, WAN, PAN, ISDN systems and ATM network, OSI-reference model, open system standards, characteristics of network, TCP/IP model, protocols and standards, encoding technique.

UNIT II SWITCHING ANDDATALINKLAYER

Circuit switching, packet switching, message switching, hybrid switching, and ATM switching, multiplexing techniques: TDMA, FDMA, WDMA, CDMA, data link layer: LLC &MAC level protocols and design issues, issues IEEE 802 LAN Standards, framing, CRC, error control, flow control, HDLC, ALOHA and performance issues. Frames relay networks and performance parameters.

UNIT III NETWORK LAYER

Network layer design issues, overview of IPv4 and IPv6, addressing: class full and classless, static and dynamic, subnet and super net, auto configuration through DHCP, routing protocols: RIP, DVR,LSR, OSFP, BGP, congestion control algorithm, subnet concept, virtual LAN, ICMP, multicasting, mobile IP.

UNIT IV TRANSPORT LAYER

Port addressing schemes, connectionless and connection oriented services: TCP and UDP, wireless TCP, Congestion control, queue management, NAT, PAT, socket format at transport level, socket interface and programming.

UNIT V APPLICATION LAYER

Client server architecture, domain name services, application services: HTTP, TELNET, RLOGIN, FTP, CBR, NFS, SMTP, POP, IMAP, MIME, voice and video over IP, social issues- privacy, freedom of speech, copyright.

.

Text Books:

1.S. Tanenbaum, Computer Networks, 4th edition, Prentice Hall, 2008
2.Forouzan, B.A., Data Communication and Networking, Tata McGraw-Hill.

References Books:

1.W. Stallings, Data and Computer Communications, 8th edition, Prentice Hall, 2007
2.Douglus E. ComerTCP/IP Principles, Protocols and Architecture, Pearson Education
3.F. Haball ,Data Communication, Computer network & open systems - Computer Networks : An Engineering approach - S. Keshav
4.Kurose, J.F. & Ross, K.W., Computer Networking: A Top-Down Approach Featuring the Internet, Addison Wesley.

COMPILER DESIGN			
Course Code:	**IT305**	**Course Credits:**	**4**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 01**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 15**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. Understand the basic principles of compiler design, its various constituent parts, algorithms and data structures required to be used in the compiler.
2. Understand relations between computer architecture and how its understanding is useful in design of a compiler.
3. How to construct efficient algorithms for compilers.
4. Provide an understanding of the fundamental principles in compiler design.
5. Learn the process of translating a modern high-level-language to executable code required for compiler construction.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Acquire knowledge of different phases and passes of the compiler and also able to use the compiler tools like LEX, YACC, etc. Students will also be able to design different types of compiler tools to meet the requirements of the realistic constraints of compilers.
2. Understand the parser and its types i.e. Top-Down and Bottom-up parsers and construction of LL, SLR, CLR, and LALR parsing table.
3. Implement the compiler using syntax-directed translation method and get knowledge about the synthesized and inherited attributes.
4. Acquire knowledge about run time data structure like symbol table organization and different techniques used in that.
5. Understand the target machine's run time environment, its instruction set for code generation and techniques used for code optimization.

UNIT I

Introduction to Compiler: Phases and passes, Bootstrapping, Finite state machines and regular expressions and their applications to lexical analysis, Optimization of DFA-Based Pattern Matchers implementation of lexical analyzers, lexical-analyzer generator, LEX compiler, Formal grammars and their application to syntax analysis, BNF notation, ambiguity, YACC. The syntactic specification of programming languages: Context free grammars, derivation and parse trees, Capabilities of CFG.

UNIT II

Basic Parsing Techniques: Parsers, Shift reduce parsing, operator precedence parsing, top down parsing, predictive parsers Automatic Construction of efficient Parsers: LR parsers, the canonical Collection of LR(0) items, constructing SLR parsing tables, constructing Canonical LR parsing tables, Constructing LALR parsing tables, using ambiguous grammars, an automatic parser generator, implementation of LR parsing tables.

UNIT III

Syntax-directed Translation: Syntax-directed Translation schemes, Implementation of Syntax- directed Translators, Intermediate code, postfix notation, Parse trees & syntax trees, three address code, quadruple & triples, translation of assignment statements, Boolean expressions, statements that alter the flow of control, postfix translation, translation with a top down parser. More about translation: Array references in arithmetic expressions, procedures call, declarations and case statements.

UNIT IV

Symbol Tables: Data structure for symbols tables, representing scope information. Run-Time Administration: Implementation of simple stack allocation scheme, storage allocation in block structured language. Error Detection & Recovery: Lexical Phase errors, syntactic phase errors semantic errors.

UNIT V

Code Generation: Design Issues, the Target Language. Addresses in the Target Code, Basic Blocks and Flow Graphs, Optimization of Basic Blocks, Code Generator. Code optimization: Machine-Independent Optimizations, Loop optimization, DAG representation of basic blocks, value numbers and algebraic laws, Global Data-Flow analysis.

Text books:

1. K. Muneeswaran,Compiler Design,First Edition,Oxford University Press.
2. J.P. Bennet, “Introduction to Compiler Techniques”, Second Edition, Tata McGraw-Hill,2003.
3. Henk Alblas and Albert Nymeyer, “Practice and Principles of Compiler Building with C”, PHI, 2001.
4. Aho, Sethi & Ullman, "Compilers: Principles, Techniques and Tools”, Pearson Education
5. V Raghvan, “ Principles of Compiler Design”, TMH

Soft Computing Techniques			
Course Code:	**IT307**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.
2 Introduce students to artificial neural networks and fuzzy theory from an engineering perspective.
3 Understanding of various learning paradigms.
4 Develop the skills to gain understanding applications of soft computing.
5 Understanding the concept of genetic algorithm.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Comprehend the fuzzy logic and the concept of fuzziness involved in various systems and fuzzy set theory.
2 Understand the concepts of fuzzy sets, knowledge representation using fuzzy rules and fuzzy inference system.
3To understand the fundamental theory and concepts of neural networks and neural network architectures.
4 Understand appropriate learning rules for each of the architectures and working of genetic algorithm.
5 Reveal different applications of these models to solve engineering and other problems.

UNIT I FUZZY LOGIC

Introduction to fuzzy logic, classical and fuzzy sets, overview of fuzzy sets, membership function, fuzzy rule generation, operations on fuzzy sets: complement, intersection, union, combinations on operations, aggregation operation.

UNIT II FUZZY ARITHMETIC

Fuzzy numbers, linguistic variables, arithmetic operations on intervals & numbers, uncertainty based information, information and uncertainty, no specificity of fuzzy and crisp sets, fuzziness of fuzzy sets.

UNIT III NEURAL NETWORK

Overview of biological neurons, computational neurons, mathematical models of neurons, ANN architecture, single layer and multilayer architectures, activation function, threshold value, self learning and forced learning algorithms, feed forward and feedback architectures.

UNIT IV LEARNING FUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning, ANN training, algorithms perceptions, training rules, delta, back propagation algorithm, multilayer perceptron model, Hopfield networks, associative memories, applications of artificial neural networks,

UNIT V GENETIC ALGORITHMS

History of genetic algorithm, terminology of genetic algorithm, biological background, creation of offspring, working principles of genetic algorithms, fitness function, reproduction: Roulette wheel selection, Boltzmann selection, crossover mutation, inversion, deletion, and duplication, generation cycle.

Reference Books:

1. Artificial Neural Networks: An introduction to ANN Theory and Practice, Peteus J. Braspenning, PHI publication, 2005.
2. Fuzzy Logic: A spectrum of Theoretical and Practical issues, Paul P. Wang, pearson publication 2004.
3. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J. Kilr, Bo yuan, 2005.
4. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conference proceeding, 2005.
5. Neural Networks Theory, Particia Melin, Oxford University press, 2003.

COMPUTER BASED NUMERICAL & STATISTICAL TECHNIQUES			
Course Code:	IT315	**Course Credits:**	**3**
Course Category:	**E1**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Present the basic of Computer Based Numerical & Statistical Techniques			
2 Helps in learning step by step approach.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of Computer Based Numerical & Statistical Techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of Computer Based Numerical & Statistical Techniques.			
2 Apply methods.			
3 learn various techniques.			
4 Practical Approach			

UNIT-I

Floating point Arithmetic: Representation of floating point numbers, Operations, Normalization, Pitfalls of floating point representation, Errors in numerical computation Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method, Iteration Method, Regula-Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods.

UNIT-II

Simultaneous Linear Equations: Solutions of system of Linear equations, Gauss Elimination direct method and pivoting, Ill Conditioned system of equations, Refinement of solution. Gauss Seidel iterative method, Rate of Convergence Interpolation and approximation: Finite Differences, Difference tables Polynomial Interpolation: Newton’s forward and backward formula Central Difference Formulae: Gauss forward and backward formula, Stirling’s, Bessel’s, Everett’s formula. Interpolation with unequal intervals: Lagrange's Interpolation, Newton

Divided difference formula, Hermite's Interpolation Approximation of function by Taylor's series and Chebyshev polynomial.

UNIT-III

Numerical Differentiation and Integration: Introduction, Numerical Differentiation, Numerical Integration, Trapezoidal rule, Simpson's rules, Boole's Rule, Weddle's Rule Euler- Maclaurin Formula Solution of differential equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta methods, Predictor-corrector method, Automatic error monitoring, stability of solution.

UNIT-IV

Curve fitting, Cubic Spline and Approximation: Method of least squares, fitting of straight lines, polynomials, exponential curves etc. Frequency Chart: Different frequency charts like Histogram, Frequency curve, Pi-chart. Regression analysis: Linear and Non-linear regression, Multiple regression.

UNIT-V

Time series and forcasting: Moving averages, smoothening of curves, forecasting models and methods. Statistical Quality Controls methods Testing of Hypothesis: Test of significance, Chi-square test, t-test, ANOVA, F-Test Application to medicine, agriculture etc.

References Books:

[1] Rajaraman V., “Computer Oriented Numerical Methods”, PHI

[2] Gerald & Wheatley, “Applied Numerical Analyses”, AW

[3] Jain, Iyengar and Jain, “Numerical Methods for Scientific and Engineering Computations”, New Age Int.

[4] Grewal B. S., “Numerical methods in Engineering and Science”, Khanna Publishers, Delhi

[5] T. Veerarajan, T Ramachandran, “Theory and Problems in Numerical Methods”, TMH

[6] Pradip Niyogi, “Numerical Analysis and Algorithms”, TMH

[7] Francis Scheld, “Numerical Analysis”, TMH

[8] Gupta S. P., “Statistical Methods”, Sultan and Sons

SOFTWARE PROJECT MANAGEMENT			
Course Code:	**IT319**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Present the basic software project management approach			
2 Helps in learning step by step.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of paradigms of software project management techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of software project management.			
2 Apply methods.			
3 learn various techniques.			
4 Practical Approach			

UNIT-I: INTRODUCTION AND SOFTWARE PROJECT PLANNING

Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, Project Management Cycle, SPM Objectives, Management Spectrum, SPM Framework, Software Project Planning, Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan, Software project estimation, Estimation methods, Estimation models, Decision process.

UNIT-II: PROJECT ORGANIZATION AND SCHEDULING

Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project Lifecycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building the project schedule, Scheduling terminology and techniques, Network Diagrams: PERT, CPM, Bar Charts: Milestone Charts, Gantt Charts.

UNIT-III: PROJECT MONITORING AND CONTROL

Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: Budgeted Cost for Work Scheduled (BCWS), Cost Variance (CV), Schedule

Variance (SV), Cost Performance Index (CPI), Schedule Performance Index (SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types of Review: Inspections, Desk checks, Walkthroughs, Code Reviews, Pair Programming.

UNIT-IV: SOFTWARE QUALITY ASSURANCE AND TESTING

Testing Objectives, Testing Principles, Test Plans, Test Cases, Types of Testing, Levels of Testing, Test Strategies, Program Correctness, Program Verification & validation, Testing Automation & Testing Tools, Concept of Software Quality, Software Quality Attributes, Software Quality Metrics, and Indicators, The SEI Capability Maturity Model CMM), SQA Activities, Formal SQA Approaches: Proof of correctness, Statistical quality assurance, Cleanroom process.

UNIT-V: PROJECT MANAGEMENT AND PROJECT MANAGEMENT TOOLS

Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, Change Control, Change Requests Management, Version Control, Risk Management: Risks and risk types, Risk Breakdown Structure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, Cost Benefit Analysis, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

Textbooks:

[1]. M. Cotterell, Software Project Management, Tata McGraw-Hill Publication.

[2]. Royce, Software Project Management, Pearson Education

Reference Books:

[3]. Kieron Conway, Software Project Management, Dreamtech Press

[4]. S. A. Kelkar, Software Project Management, PHI Publication.

[5]. Harold R. Kerzner, Project Management “A Systems Approach to Planning, Scheduling, and Controlling” Wiley.

COMPUTER NETWORK LAB			
Course Code:	**IT381**	**Course Credits:**	**2**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 The objective of this lab course is to get practical knowledge of working principles of various communication protocols			

3.Implementing various network algorithms such as error control, error detection, routing, and security related algorithms.
4 Understanding of learning the process and application
5 The objective of this lab course is to get practical knowledge of working
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand the practical approach to network communication protocols.
2 Understand network layers, structure/format and role of each network layer
3 learn various techniques.
4 Practical Approach

List of Experiments:

1. Introduction to transmission media(CAT5, OFC, COAXIAL CABLE Wireless)
2. Introduces network interfaces(Wired and Wireless)
3. Configure and installing a Ethernet(10/100)
4. Performance evaluation of Ethernet(10/100)
5. Topology design(Ring, Bus)
6. Generation of data packet and measurement(CBR, VBR, Poison)
7. Router configuration
8. Switch configuration
9. Server configuration
10. Congestion control of network
11. QoS of network
12. Protocols and the configuration
13. Wireless systems
14. S3curity (WEP, WPA)

COMPILER DESIGN LAB

Course Code:	**IT383**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1 The objective of this lab course is to get teach the students the basic techniques that underlie the practice of Compiler Construction.

2 The objective of this lab course is to introduce the theory and tools that can be standard employed in order to perform syntax-directed translation of a high-level programming language into an executable code.

3 Implementing various syntax-directed code

4 Understanding of learning the process and application

5 Understanding of paradigms

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Understand the practical approach to introduce the major concept areas of language translation and compiler design

2 Develop an awareness of the function and complexity of modern compilers.

3 learn various techniques.

4 Practical Approach

List of Experiments:

15. Define LEX and YACC tools in detail.
16. Write a program to check whether a string belongs to the grammar or not.
17. Write a program to generate a parse tree.
18. WAP to convert regular expression into NFA.
19. WAP to generate tokens for a given grammar.
20. Write a program to find leading terminals.
21. Write a program to find trailing terminals.
22. Write a program to compute FIRST of non-terminals.

23. Write a program to compute FOLLOW of non-terminals.

24. Write a program to check whether a grammar is left recursive and remove left recursion.

25. Write a program to remove left factoring.

26. Write a program to check whether a grammar is Operator precedence.

27. Write a Program to implement Push Down Automata.

28. Write a program to implement Thomson’s construct

SOFT COMPUTING TECHNIQUES LAB			
Course Code:	**IT385**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
This course will enable students to:			
1. Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.			
2. Introduce students to artificial neural networks and fuzzy theory from an engineering perspective.			
3. To design and development of soft computing algorithms for solution of real time problem.			
4. Learn neural networks with back propagation and without preparation.			
5. Genetic algorithms, its applications and advances.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Understand components of Soft Computing and differentiate between hard and soft computing.			
2. Understand the difference between learning and programming and explore practical applications of Neural Networks (NN).			
3. Understand the concepts of fuzzy sets, knowledge representation using fuzzy rules, approximate reasoning, fuzzy inference systems, and fuzzy logic			

4. Understand appropriate learning rules for each of the architectures and learn several neural network paradigms and its applications
5. Understand the basics of genetic algorithm, use of GA operators and its applications

List of Experiments:

1. Implementation of operations and properties of classical sets.
2. Implementation of operations and properties of Fuzzy Sets.
3. Implementation of a Cartesian Product and Fuzzy Cartesian Product.
4. Implementation of composition of classical and fuzzy relations.
5. Implementing Simple perceptron model to show the effect of inputs, weights and bias on decision boundary.
6. Train the simple perceptron to show its limitation to solve only linearly separable boundary problems.
7. To design, implement and train a feedforward ANN to solve a pattern recognition problem.
8. Implementation of adaptive filters using ANN.
9. Implementing SGA to solve function optimization problems in a given range.
10. Implementation of genetic operators for Permutation Representation.
11. Implementation of Roulette Wheel Selection.
12. Implementation of Stochastic Universal Sampling.
13. Implementation of Rank Selection.
14. Implementation of Tournament Selection.
15. Illustrate the effect of Sigma Scaling.

CLOUD COMPUTING			
Course Code:	**IT302**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. The objective of this course is to introduce students to the foundation of cloud computing.

2. To provide students a sound foundation of the Cloud Computing so that they are able to start using and adopting Cloud Computing services and tools in their real life scenarios

3. To enable students exploring some important cloud computing driven commercial systems and applications.

4. To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.

2. Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.

3. Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.

4. Analyze various cloud programming models and apply them to solve problems on the cloud.

Unit 1: Introduction to Cloud Computing: Definition, Characteristics, Components, Cloud provider, SAAS, PAAS, IAAS and Others, Organizational scenarios of clouds, Administering & Monitoring cloud services, benefits and limitations, Deploy application over cloud, Comparison among SAAS, PAAS, IAAS Cloud computing platforms: Infrastructure as service: Amazon EC2,Platform as Service: Google App Engine, Microsoft Azure, Utility Computing, Elastic Computing

Unit 2: Introduction to Cloud Technologies: Study of Hypervisors Compare SOAP and REST Web Services, AJAX and mashups-Web services: SOAP and REST, SOAP versus REST, AJAX: asynchronous 'rich' interfaces, Mashups: user interface services Virtualization Technology: Virtual machine technology, virtualization applications in enterprises, Pitfalls of virtualization Multi Tenant software:Multi-entity support, Multi-schema approach, Multi-tenancy using cloud data stores, Data access control for enterprise applications,

Unit 3: Data in the cloud: Relational databases, Cloud file systems: GFS and HDFS, BigTable, HBase and Dynamo. Map-Reduce and extensions: Parallel computing, The map-Reduce model, Parallel efficiency of Map-Reduce, Relational operations using Map-Reduce, Enterprise batch processing using Map-Reduce, Introduction to cloud development, Example/Application of Mapreduce, Features and comparisons among GFS,HDFS etc, Map-Reduce model Cloud security fundamentals, Vulnerability assessment tool for cloud, Privacy and Security in cloud Cloud computing security architecture:Architectural Considerations- General Issues, Trusted Cloud computing, Secure Execution Environments and Communications, Micro-architectures; Identity Management and Access control-Identity management, Access control, Autonomic Security

Cloud computing security challenges: Virtualization security management- virtual threats, VM Security Recommendations, VM-Specific Security techniques, Secure Execution Environments and Communications in cloud

Unit 4: Issues in cloud computing, Implementing real time application over cloud platform Issues in Intercloud environments, QOS Issues in Cloud, Dependability, data migration, streaming in Cloud. Quality of Service (QoS) monitoring in a Cloud computing environment. Cloud Middleware. Mobile Cloud Computing. Inter Cloud issues. A grid of clouds, Sky computing, load balancing, resource optimization, resource dynamic reconfiguration, Monitoring in Cloud

Unit 5: Cloud computing platforms, Installing cloud platforms and performance evaluation Features and functions of cloud platforms: Xen Cloud Platform, Eucalyptus, OpenNebula, Nimbus, TPlatform, Apache Virtual Computing Lab (VCL), Enomaly Elastic Computing Platform

Text Books:

[1] Judith Hurwitz, R.Bloor, M.Kanfman, F.Halper, Cloud Computing for Dummies by (Wiley India Edition)

Reference Books:

[2]. Gautam Shroff, Enterprise Cloud Computing by, Cambridge

[3]. Ronald Krutz and Russell Dean Vines, Cloud Security by, Wiley-India

ANALYSIS & DESIGN OF ALGORITHMS			
Course Code:	**IT304**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
1. COURSE OBJECTIVES			
2. Analyze the asymptotic performance of algorithms.			
3. Write rigorous correctness proofs for algorithms.			
4. Demonstrate a familiarity with major algorithms and data structures.			

5. Apply important algorithmic design paradigms and methods of analysis.
6. Synthesize efficient algorithms in common engineering design situations.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Argue the correctness of algorithms using inductive proofs and invariant
2. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.
3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.
4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.
5. Analyze worst-case running times of algorithms using asymptotic analysis.

UNIT I BASIC CONCEPTS OF ALGORITHMS

Introduction, notion of algorithm, fundamentals of algorithmic solving, important problem types, Fundamentals of the analysis framework, asymptotic notations and basic efficiency classes.

UNIT II MATHEMATICAL ASPECTS AND ANALYSIS OF ALGORITHMS

Mathematical analysis of non-recursive algorithms, mathematical analysis of recursive algorithms, example: fibonacci numbers, empirical analysis of algorithms, algorithm visualization.

Unit III ANALYSIS OF SORTING AND SEARCHING ALGORITHMS

Brute force, selection sort and bubble sort, sequential search and brute-force string matching, divide and conquer, merge sort, quick sort, binary search, binary tree, traversal and related properties, decrease and conquer, insertion sort, depth first search and breadth first search.

UNIT IV ALGORITHMIC TECHNIQUES

Transform and conquer ,presorting, balanced search trees, avl trees, heaps and heap sort, dynamic programming, Warshall’s and Floyd’s algorithm, optimal binary search trees, greedy techniques, Prim’s
algorithm, Kruskal’s algorithm, Dijkstra’s algorithm, Huffman trees.

UNIT V ALGORITHM DESIGN METHODS

Backtracking, n-Queens problem, Hamiltonian circuit problem, subset-sum problem, branch and bound,
assignment problem, knapsack problem, traveling salesman problem.

Text Books:

[1]. Anany Levitin, “Introduction to the Design and Analysis of Algorithm”, Pearson Education Asia, 2003.

References Books:

[2]. T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, “Introduction to Algorithms”, PHI Pvt. Ltd., 2001

[3]. Sara Baase and Allen Van Gelder, “Computer Algorithms - Introduction to Design and Analysis”, Pearson Education Asia, 2003.

[4]. A.V.Aho, J.E. Hopcroft and J.D.Ullman, “The Design and Analysis of Computer Algorithms”,Pearson Education Asia, 2003.

COMPUTER ORGANIZATION			
Course Code:	**IT306**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Discuss the basic computer arithmetic and number system			
2.Understand concepts of register transfer logic and microoperation.			
3.Explain different types of processor design			
4.Learn the input-output organization			
5.Summarize the memory organization.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1.Define different number systems, binary addition and subtraction, 2’s complement representation and operations with this representation.			
2.Able to understand the theory and architecture of processor design			
3.Analyze some of input-output organization and different techniques			

UNIT I COMPUTER ARITHMETIC AND NUMBER SYSTEM

Number representation; number system, fixed and floating point number representation, arithmetic algorithms (addition, subtraction, booth multiplication).

UNIT II REGISTER TRANSFER AND MICRO OPERATION

Register transfer language, bus and memory transfers, bus architecture, bus arbitration, arithmetic logic, shift microoperation, arithmetic logic shift unit, design of fast address.

UNIT II PROCESSOR DESIGN

Processor organization: general register organization, stack organization, addressing mode, instruction format, data transfer & manipulations, program control, reduced instruction set computer.

UNIT IV INPUT-OUTPUT ORGANIZATION

I/O interface, synchronous and asynchronous data transfer, strobe, handshaking schemes, modes of transfer, interrupts & interrupt handling, direct memory access, input-output processor.

UNIT V MEMORY ORGANIZATION

Memory hierarchy, main memory (RAM and ROM Chips), organization of 2d and $2^{1/2}$d, auxiliary memory, cache memory, virtual memory, memory management hardware.

Books:

[1]. Patterson, Computer Organisation and Design, Elsevier Pub. 2009
[2]. William Stalling, “Computer Organization”, PHI

Reference Books:

[3]. Vravice,Hamacher & Zaky, “Computer Organization”, TMH
[4]. Mano,” Computer System Architecture”, PHI
[5]. John P Hays, “Computer Organization”, McGraw Hill
[6]. Tannenbaum,” Structured Computer Organization’, PHI
[7]. P Pal chaudhry, ‘Computer Organization & Design’, PHI

Information & Network Security			
Course Code:	**IT308**	**Course Credits:**	**4**
Course Category:	**C**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**3U**
No. of Lectures + Tutorials (Hrs. /Week):	**03+02**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45+15**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1: To develop a fundamental understanding of data, information and the security requirements.			

2: To create awareness about information security vulnerabilities, threats, principles, assets and risk management.
3: To learn and understand encryption algorithms and related cryptographic operations.
4: To develop the ability to understand entity authentication requirements in networks and web security protocols and applications
5: To Acquire understanding of information security related policies, violations, cybercrimes, laws and standards.
COURSE OUTCOMES
At the end of the course the students should be able to:
1: Understand the information and the security requirements fundamentals.
2: Understand the security principles and risk management procedures.
3: Understand the cryptographic algorithms with their applications.
4: Understand network and web security protocols and malicious codes, firewalls etc.
5: Understand the requirements of policies, standards, cyber security crimes and laws.

UNIT I: Introduction to Information Security: Definition of information, security, need of information security, CIA triad, principles of information security, Vulnerability, Threat, Organizational Assets-definition, types of assets, asset classification, Risk management, Security goals, attacks, types of attacks, services and mechanisms, Physical security, cryptography: Classical encryption techniques-substitution ciphers and transposition ciphers and their types.

UNIT II: Cryptography: Stream and block ciphers. Shannon's theory of confusion and diffusion, fiestal structure, Data encryption standard (DES), Triple DES, Introduction to Group, Field, Finite Field of the form GF(p), modular arithmetic, prime and relative prime numbers, Euclidean Algorithm, Advanced Encryption Standard (AES) encryption and decryption Fermat's and Euler's theorem, Primarily testing, Principals of public key crypto systems, RSA & DHKE algorithm.

UNIT III: Message Authentication: Authentication requirements, authentication code & functions, message authentication code, hash functions, security of hash functions, Secure hash algorithm (SHA) Digital Signatures: Digital signature standards (DSS), Key Management and distribution: Symmetric key distribution, Public key distribution, Public key Infrastructure.

UNIT IV: Network Security: Authentication Applications: Kerberos, Electronic mail security: pretty good privacy (PGP), S/MIME. IP Security: Architecture, Authentication header, Encapsulating security payloads, combining security associations. Secure Socket

Layer, Secure electronic transaction (SET) System Security: Intrusion & Intrusion detection, Viruses and related threats, firewalls.

UNIT V: Information Security Standards & Laws: Policy, Types of policies, Need of an Information Security Policy, Standards, Procedures, Guidelines; Information Security Management System (ISMS), ISO 27001Standard. Cyber Warfare-Cyber Crime-Cyber terrorism-Cyber Espionage, Types of cyber-crimes, IT ACT 2000, Evidence Act 1872-Admisibility electronic evidence.

Text Books:

[1] William Stallings, "Cryptography and Network Security: Principals and Practice", Pearson Education.

[2] Behrouz A. Frouzan: Cryptography and Network Security, Tata McGraw Hill.

Reference Books:

[1] Merkow, "Information Security Principles & Practices"

[2] Christof Paar & Jan Pelzel, Understanding Cryptography, Springer.

[3] Bare Act Information Technology ACT 2000.

[4] C K Shyamala, N Harini, Dr. T.R. Padmnabhan Cryptography and Security, Wiley.

[5] Bruce Schiener, "Applied Cryptography". John Wiley & Sons.

[6] Bernard Menezes," Network Security and Cryptography", Cengage Learning.

[7] Atul Kahate, "Cryptography and Network Security", Tata McGraw Hill.

[8] Thomas R. Peltier, Justin Peltier, John Blackley, Information Security Fundamentals.

[9] John Robertson, Ahmad Diab, Ericsson Marin, Eric Nunes, VivinPaliath, Jana Shakarian, Paulo Shakarian, DarkWeb Cyber Threat Intelligence Mining Cambridge University Press.

Computer Based Numerical & Statistical Techniques			
Course Code:	**IT315**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**1U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.Discuss the working understanding of numerical methods for the basic problems of numerical analysis.			
2.Understand concepts of implementation of numerical methods using a computer..			

3.Explain different types of various significant and fundamental concepts to inculcate in the students an adequate understanding of the application of Statistical Methods.
4.Learn the concepts of numerical methods used for different applications
5.Summarize the different oriented numerical and statistical techniques
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Understanding and Learning of numerical methods for numerical analysis.
2. Understanding the implementation of numerical methods using a computer.
3. Learning of tracing errors in Numerical methods and analyze and predict it.

Unit-I

Floating point Arithmetic: Representation of floating point numbers, Operations, Normalization, Pitfalls of floating point representation, Errors in numerical computation Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method, Iteration Method, Regula-Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods.

Unit-II

Simultaneous Linear Equations: Solutions of system of Linear equations, Gauss Elimination direct method and pivoting, Ill Conditioned system of equations, Refinement of solution. Gauss Seidal iterative method, Rate of Convergence Interpolation and approximation: Finite Differences, Difference tables Polynomial Interpolation: Newton’s forward and backward formula Central Difference Formulae: Gauss forward and backward formula, Stirling’s, Bessel’s, Everett’s formula. Interpolation with unequal intervals: Langrange’s Interpolation, Newton Divided difference formula, Hermite’s Interpolation Approximation of function by Taylor’s series and Chebyshev polynomial

Unit-III

Numerical Differentiation and Integration: Introduction, Numerical Differentiation, Numerical Integration, Trapezoidal rule, Simpson’s rules, Boole’s Rule, Weddle’s Rule Euler- Maclaurin Formula Solution of differential equations: Picard’s Method, Euler’s Method, Taylor’s Method, Runge-Kutta methods, Predictor-corrector method, Automatic error monitoring, stability of solution.

Unit-IV

Curve fitting, Cubic Spline and Approximation: Method of least squares, fitting of straight lines, polynomials, exponential curves etc. Frequency Chart: Different frequency chart like Histogram, Frequency curve, Pi-chart. Regression analysis: Linear and Non-linear regression, Multiple regression

Unit-V

Time series and forcasting: Moving averages, smoothening of curves, forecasting models and methods. Statistical Quality Controls methods Testing of Hypothesis: Test of significance, Chi-square test, t-test, ANOVA, F-Test Application to medicine, agriculture etc.

References Books:

[1] Rajaraman V., "Computer Oriented Numerical Methods", PHI
[2] Gerald & Wheatley, "Applied Numerical Analyses", AW
[3] Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
[4] Grewal B. S., "Numerical methods in Engineering and Science", Khanna Publishers, Delhi
[5] T. Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods", TMH
[6] Pradip Niyogi, "Numerical Analysis and Algorithms", TMH
[7] Francis Scheld, "Numerical Analysis", TMH
[8] Gupta S. P., "Statistical Methods", Sultan and Sons

CLOUD COMPUTING LAB			
Course Code:	**IT382**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. The objective of this course is to introduce students to the foundation of cloud computing.			
2. To provide students a sound foundation of the Cloud Computing so that they are able to start using and adopting Cloud Computing services and tools in their real life scenarios			
3. To enable students exploring some important cloud computing driven commercial systems and applications.			
4. To expose the students to frontier areas of Cloud Computing and information systems, while providing sufficient foundations to enable further study and research.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.			

2. Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
3. Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.
4. Analyze various cloud programming models and apply them to solve problems on the cloud.

List of Experiments:

1. Study of NIST model of cloud computing. Objective: Understand deployment models, service models, advantages of cloud computing.
2. Install Virtualbox/VMware Workstation with different flavours of linux or windows OS on top of windows7 or 8.
3. Install a C compiler in the virtual machine created using virtual box and execute Simple Programs
4. Install Google App Engine. Create hello world app and other simple web applications using python/java.
5. Use GAE launcher to launch the web applications.
6. Virtualization: Understand different types of virtualizations, Host and bare metal hypervisors and implement horizontal scalability. Technology: XEN/ Vmwares EXSi
7. Infrastructure as a Service: Implement IaaS using your resources. Technology: Open Stack / Eucalyptus
8. Identity Management in Cloud Concept: Simulate identity management in your private cloud. Technology: Open Stack
9. Storage as a Service: To explore Storage as a Service for remote file access using web interface. Technology: own Cloud
10. Cloud Security: To Understand security of web server and data directory. Technology: ownCloud
11. Platform as a Service: To Deploy web applications on commercial cloud. Technology: Google appEngine/ Windows Azure
12. Amazon Web Service: To create and access VM instances and demonstrate various.

ANALYSIS & DESIGN OF ALGORITHMS LAB			
Course Code:	**IT384**	**Course Credits:**	**2**
Course Category:	**CCL**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	
Total No. of Lectures (L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**
1. COURSE OBJECTIVES			

2. Analyze the asymptotic performance of algorithms.

3. Write rigorous correctness proofs for algorithms.

4. Demonstrate a familiarity with major algorithms and data structures.

5. Apply important algorithmic design paradigms and methods of analysis.

6. Synthesize efficient algorithms in common engineering design situations.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Argue the correctness of algorithms using inductive proofs and invariant

2. Explain the major graph algorithms and their analyses. Employ graphs to model engineering problems, when appropriate. Synthesize new graph algorithms and algorithms that employ graph computations as key components, and analyze them.

3. Describe the divide-and-conquer paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize divide-and-conquer algorithms. Derive and solve recurrences describing the performance of divide-and-conquer algorithms.

4. Describe the dynamic-programming paradigm and explain when an algorithmic design situation calls for it. Recite algorithms that employ this paradigm. Synthesize dynamic-programming algorithms, and analyze them.

5. Analyze worst-case running times of algorithms using asymptotic analysis.

List of Experiments:

1. Implement the minimum cost spanning tree algorithm.
2. Implement the single source shortest path algorithm.
3. Implement the algorithm for optimal binary search tree.
4. Implement the algorithm for Job sequencing with deadlines.
5. Implement the algorithm for sum of subsets problem.
6. Implement the algorithm for traveling sales person problem.
7. Implement the algorithm for knapsack problem.
8. Implement the algorithm for n-queen problem.
9. Implement the algorithm for graph coloring.
10. Implement the algorithm for all pair shortest path.
11. Implement all types of sorting techniques and analyze time complexity.
12. Implement matrix multiplication.

INFORMATION & NETWORK SECURITY LAB			
Course Code:	**IT386**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lab Sessions(L + T):	**12 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To learn different cipher techniques.			
2. To implement the algorithms DES, RSA			
3. To implement the algorithm MD5			
4. To implement the algorithm SHA-1			
5. To use network security tools and vulnerability assessment tools			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Develop code for classical Encryption Techniques to solve the problems.			
2. Build cryptosystems by applying symmetric and public key encryption algorithms.			
3. Construct code for authentication algorithms.			
4. Develop a signature scheme using Digital signature standard.			
5. Demonstrate the network security system using open source tools.			

LIST OF EXPERIMENTS:

1. Perform encryption, decryption using the following substitution techniques

i. Caesar Cipher

ii. Playfair Cipher

2. Perform encryption, decryption using the following substitution techniques

i. Hill Cipher

ii. Vigenere Cipher

3. Perform encryption, decryption using the following transposition techniques

i. Rail fence- Row & Column Transformation

4. Apply the DES algorithm for practical applications.

5. Apply the AES algorithm for practical applications.

6. Implement RSA Algorithm using HTML and JavaScript.

7. Implement the Diffie-Hellman Key Exchange algorithm for a given problem.

8. Calculate the message digest of a text using the SHA-1 algorithm.

9. Implement the SIGNATURE SCHEME- Digital Signature Standard.

10. Demonstrate Intrusion Detection System(IDs) using any tool(Snort or any other software.)

SEMESTER VII

SEMESTER VII							
S. No.	**Course Code**	**Course Name**	**L**	**T**	**P**	**Credits**	**Types**
1	MA401	Modeling and simulation	3	0	0	3	
2	IT401	Sensor Networks and IoT	3	0	0	3	
3	IT403	Soft Computing Techniques	3	0	0	3	
4	IT405	Big Data Analytics	3	0	0	3	
5		Elective 5	3	0	0	3	
6	IT481	Sensor Networks and IoT Lab	0	0	3	2	
7	IT491	Industrial Training	0	0	6	3	
8	IT493	Minor Project	0	0	10	5	
9	GP	General Proficiency	Non Credit				
Total Hours and Credits			**15**	**0**	**19**	**25**	

Modeling and simulation			
Course Code:	**MA401**	**Course Credits:**	**3**
Course Category:	**C**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs. /Week):	**03**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1: To introduce with the various system simulation and modeling techniques, and highlight their applications.			
2: To introduce modeling, design, simulation, planning, verification and validation in the areas of simulation.			
3: To develop skills among the learners of system simulation..			
4: To make them able to solve real world problems, which cannot be solved by mathematical approaches.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1: Understand the different modelling terms by analyzing the system or the data that is present..			
2: Understand the different mathematical model and their application in simulation.			
3: Understand the model and from the results check for the correctness of the assumptions..			
4: Understand outcomes and make predictions.			
5: Understand the requirement of policies, standards, cyber security crimes and laws.			

Unit 1:

Introduction to simulation and modelling, simulation process, advantages and disadvantages of simulation techniques, limitations of simulation techniques, comparison of simulation and analytical methods, fixed time step vs even to even model, analog vs digital simulation.

Unit 2:

Simulation of continuous systems, simulation of water reservoir system, simulation of servo system, simulation of an autopilot

Unit 3:

Discrete system simulation, queuing models, characteristics of queuing models, behavior of arrivals, pattern of arrival at the system, the arrival time distribution, queuing process, queuing discipline, service process, distribution of service time, performance measures of a queuing system, classification of queuing models, single server queuing models and its simulation, inventory model, deterministic inventory models with and without shortage cost model, simulation of inventory models

Unit 4:

Monte Carlo simulation, real time simulation, hybrid simulation, distributed log models, cobweb model, generation of random numbers, test for randomness, confidence interval, statistical methods, CPM and PERT networks characteristics, critical path computation

Unit 5:

System dynamics, system dynamic diagrams, exponential growth models, exponential decay models, modified exponential growth models, verification and validation of simulation models, techniques for verification and validation of model, time series approach

Sensor Network and IOT			
Course Code:	**IT 401**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**3 + 0**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 0**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To create understanding of various concepts of Sensor Network and Internet of things (IOT)			
2 To learn the architectures of Sensor network and IOT			
3 To Understand and analyze various protocols in the domain area			
4 Conceptualize the application of Sensor and IOT			
COURSE OUTCOME			
At the end of the course the students should be able to:			
1 Differentiate between traditional, adhoc and wireless sensor network and IOT			
2 Use the concepts for conceptualizing modern day networks			
3 Explore various research domains in Sensor, IOT and their protocols			
4 Develop the architectures of Sensor IOT based applications			

Unit I INTRODUCTION TO WIRELESS SENSOR NETWORK AND ARCHITECTURE

Single node architecture: hardware and software components of a sensor node – Wireless Sensor Network (WSN) architecture: typical network architectures- Different types of Sensor Nodes.

UNIT I COMMUNICATION PROTOCOLS AND SUPPORT PROCESSES

PHYSICAL LAYER, MAC Layer protocols, Link Layer protocols, Routing Protocols, Transport layer protocols; **Support Processes**- Clustering, Data Aggregation, Naming and Addressing, Localization and Positioning, Topology Control, Applications

UNIT III Basics of IoT and Architecture

Definition, building blocks of IoT, conceptual framework, M2M communication and IoT, IoT connectivity and communication APIs, role of sensor networks in IoT (sensing and actuation), vision of 5G and Next generation networks, Layered Architecture of IOT

UNIT IV Protocols of IoT and Computing

layer wise description of IoT protocols, IEEE standards, wired and wireless protocols for enabling IoT, Concept of Data analytics; Data Analytics and Computing- Introduction to Big Data, Introduction of Machine Learning for IoT, Fog Computing, Edge Computing and Cloud computing

UNIT V Sensor and IoT Applications

Sensor network Applications – Environmental monitoring, Event based monitoring; IOT Applications- Smart Home, Smart Healthcare, Smart Environmental monitoring, Structural Health Monitoring (SHM) in IoT, Smart Metering, Smart Transportation and logistics, Smart Industrial applications, Sensor network and IoT Challenges & Open Research Issues

Text Books

- Holger Karl and Andreas Willig "Protocols and Architectures for Wireless Sensor Networks", Wiley, 2005
- Vidushi Sharma, and Anuradha Pughat, Eds. "Energy Efficient Wireless Sensor Networks", CRC Press (Taylor & Francis), Florida, USA, 2017, pp 1-276. ISBN:978-1498783347, 2018
- David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things", First edition, Pearson Education (Cisco Press Indian Reprint)
- Arshdeep Bahga and Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press, First edition,2014

REFERENCE BOOKS:

- Carlos De Morais Cordeiro, Dharma Prakash Agrawal "Ad Hoc & Sensor Networks: Theory and Applications", World Scientific Publishing Company, 2006.
- Feng Zhao and Leonides Guibas, "Wireless Sensor Networks", Elsevier Publication – 2002.
- Kazem Sohraby, Daniel Minoli, & Taieb Znati, "Wireless Sensor Networks-Technology, Protocols, and Applications", John Wiley, 2007
- Honbu Zhou, "The Internet of Things in the Cloud: A Middleware Perspective", CRC press, First edition, 2012
- Olivier Hersent, David Boswarthick, Omar Elloumi, "The Internet of Things- Key applications and protocols", Wiley, 2012
- Michael Miller, "The Internet of Things", QUE Publishing, 2015
- Peter Waher, "Learning Internet of Things", Packt Publishing, 2015

SOFT Computing Techniques

Course Code:	**IT-403**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. The primary objective of this course is to provide an introduction to the basic principles, techniques, and applications of soft computing

2. Develop the skills to gain a basic understanding of neural network theory and fuzzy logic theory.

3. Introduce students to artificial neural networks and fuzzy theory from an engineering perspective

4. Provide the mathematical background for carrying out the optimization associated with neural network learning

5. Aim of this course is to develop some familiarity with current research problems and research methods in Soft Computing by working on a research or design project.

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Describe human intelligence and AI
2. Explain how intelligent system works
3. Apply basics of Fuzzy logic and neural networks..
4. Discuss the ideas of fuzzy sets, fuzzy logic and use of heuristics based on human experience
5. Describe with genetic algorithms and other random search procedures useful while seeking global optimum in self-learning situations

UNIT I Neural network:

Overview of biological neurons, computational neuron, mathematical model of neurons, ANN architecture, single layer and multilayer architectures, activation function, threshold value, self-learning and forced learning algorithms, feed forward and feedback architectures.

UNIT II Learning fundamentals:

Learning paradigms, supervised and unsupervised learning, reinforced learning, ANN training, algorithms perceptions, training rules, delta, back propagation algorithm, multilayer perception model, Hopfield networks, associative memories, applications of artificial neural networks.

UNIT III Fuzzy logic:

Introduction to fuzzy logic, classical and fuzzy sets, overview of fuzzy sets, membership function, fuzzy rule generation, operations on fuzzy sets: compliment, intersection, union, combinations on operations, aggregation operation.

UNIT IV Fuzzy arithmetic:

Fuzzy numbers, linguistic variables, arithmetic operations on intervals & numbers, uncertainty based information, information and uncertainty, no specificity of fuzzy and crisp sets, fuzziness of fuzzy sets.

UNIT V Genetic algorithms:

History of genetic algorithm, terminology of genetic algorithm, biological background, creation of offspring, working principles of genetic algorithms, fitness function, reproduction: Roulette wheel selection, Boltzmann selection, cross over mutation, inversion, deletion, and duplication, generation cycle.

Text Books:

[1]. Peteus J. Braspenning, Artificial Neural Networks: An introduction to ANN Theory and Practice, PHI publication, 2005.

[2]. Paul P. Wang, Fuzzy Logic: A spectrum of Theoretical and Practical issues, Pearson publication 2004.

Reference Books:

[1]. Lotfi, Asker Zadeh, George J. Kilr, Bo yuan , Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- 2005.

[2]. Foundations of Fuzzy logic and Soft Computing: 12th International Fuzzy conference proceeding, 2005.

[3]. Neural Networks Theory, Particia Melin, Oxford University press, 2003.

[4]. Oscar Castillo, Neural Networks Theory and Application, Wiley Eastern publication.

Big Data Analytics			
Course Code:	**IT-405**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Understand the Big Data Platform and its Use cases			
2 Provide an overview of Apache Hadoop			
3 Provide HDFS Concepts and Interfacing with HDFS			
4 Provide hands on Hodoop Eco System			
5 Apply analytics on Structured, Unstructured Data..			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
At the end of the course the students should be able to:			
1 Identify Big Data and its Business Implications.			
2 List the components of Hadoop and Hadoop Eco-System			
3 Access and Process Data on Distributed File System			
4 Manage Job Execution in Hadoop Environment			
5 Develop Big Data Solutions using Hadoop Eco System			

UNIT I: UNDERSTANDING BIG DATA:

What is big data, why big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT II: NOSQL DATA MANAGEMENT:

Introduction to NoSQL , aggregate data models ,aggregates ,keyvalue and document data models, relationships, graph databases, schema less databases ,materialized views, distribution models, sharding, master slavenreplication, peer-peer replication, sharding and replication, consistency, relaxing consistency, version stamps, mapreduce, partitioning and combining, composing mapreduce calculations.

UNIT III: BASICS OF HADOOP:

Data format, analyzing data with Hadoop, scaling out, Hadoop streaming, Hadoop pipes, design of Hadoop distributed file system (HDFS), HDFS concepts, Java interface, data flow, Hadoop I/O, data integrity, compression , serialization, Avro file-based data structures

UNIT IV: MAP REDUCE APPLICATIONS :

Map Reduce workflows , unit tests with MRUnit , test data and local tests – anatomy of Map Reduce job run , classic Map-reduce , YARN , failures in classic Map-reduce and YARN , job scheduling , shuffle and sort , task execution , MapReduce types , input formats , output formats.

UNIT V: HADOOP RELATED TOOLS:

Hbase,data model and implementations, Hbase clients, Hbase examples – praxis. Cassandra, cassandra data model, cassandra examples, cassandra clients, Hadoop integration. Pig, Grunt, pig data model, Pig Latin, developing and testing Pig Latin scripts. Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation – HiveQL queries

Text Books:

1. Michael Minelli, Michelle Chambers, and Ambiga Dhiraj, "Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses", Wiley, 2013.
2. Big-Data Black Book, DT Editorial Services, Wily India

Reference Books:

3. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence", Addison-Wesley Professional, 2012.
4. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012. 5. Eric Sammer, "Hadoop Operations", O'Reilley, 2012.
5. E. Capriolo, D. Wampler, and J. Rutherglen, "Programming Hive", O'Reilley, 2012. 7. Lars George, "HBase: The Definitive Guide", O'Reilley, 2011.
6. Eben Hewitt, "Cassandra: The Definitive Guide", O'Reilley, 2010. [7]. Alan Gates, "Programming Pig", O'Reilley, 2011

GRAPH THEORY & AlGORITHMS

Course Code:	**IT-407**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1 To understand the concept of vertex connectivity and edge connectivity in graphs.

2 To develop the under-standing of Geometric duals in Planar Graphs

3To understand the concept of digraphs, Euler digraphs and Hamiltonian digraphs

4 To understand the concept of matrices in graphs like Incidence matrix, Adjacency matrix, Cycle matrix etc

5 To introduce the idea of coloring in graphs.

COURSE OUTCOMES

At the end of the course the students should be able to:

1 Understand and explain the basic concepts of graph theory.

2 Apply the basic concepts of mathematical logic.

3 Compute the maximum flow in networks and find its application

4 Understanding the concepts of connectivity and edge connectivity.

5 Identify maximum matching in a bipartite graph

UNIT-I: INTRODUCTION

Graphs, sub graphs, vertex degrees, walks, path, cycles and trails, connected graphs, disconnected graphs and components, matrix representation of graphs, isomorphism, Euler graphs, Hamiltonian paths and circuits, bipartite graphs.

UNIT-II: TREES AND CONNECTIVITY

Trees−rooted, binary trees and spanning trees, bridges, fundamental circuits, distance, center, diameter, eccentricity, radius and pendent vertices, Prim's, Kruskal's and Dijkstra's Algorithms, cut vertices, blocks and connectivity,

UNIT-III: PLANARITY, EULER TOURS AND HAMILTONIAN CYCLES

Planer graphs – Different representation of a planer graph, discussion on criterion of planarity, thickness and crossings, Euler's formula, Platonic bodies, combinatorial and geometric dual: Kuratowski's graphs, detection of planarity, geometric dual, Euler tours, Hamiltonian cycles and travelling salesman problem.

UNIT –IV: DIRECTED GRAPH AND COLORING

Directed graphs−definitions, in-degree, out-degree, orientations and tournaments, Coloring−vertex coloring, edge coloring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, and four color problem.

UNIT –V: EXTREMAL PROBLEMS

Enumeration of graphs, Ramsey's theorem, Ramsey numbers, edge Ramsey numbers, a generalization of party problem, Sperner's lemma and their applications.

Text Books:

1. West, Douglas Brent. Introduction to graph theory. Vol. 2. Upper Saddle River: Prentice hall, 2001.
2. Clark, John, and Derek Allan Holton. A first look at graph theory. Vol. 1. Teaneck, NJ: World Scientific, 1991.

Reference Books:

1. Deo, Narsingh. Graph theory with applications to engineering and computer science. Courier Dover Publications, 2016.

2. Chartrand, Gary. Introduction to graph theory. Tata McGraw-Hill Education, 2006.

MOBILE COMPUTING			
Course Code:	**IT-412**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To understand the concept of different mobile applications and basic design of embedded system			
2 To develop the under-standing of advanced design			
3To understand the android technology			
4 To understand the ios technology			
5 To introduce the idea of coloring in graphs.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Understand and explain the basic concepts of mobile applications			
2 Understand the basic concepts of embedded system OS			
3 Understand the basic concepts of ANDROID			
4 Understand the basic concepts of IOS			

UNIT I

INTRODUCTION: Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements Gathering and validation for mobile applications

UNIT II

BASIC DESIGN: Introduction – Basics of embedded systems design – Embedded OS – Design Constraints for mobile applications, both hardware and software related – Architecting mobile applications – User interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

UNIT III

ADVANCED DESIGN: Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in cloud computing environment – Design patterns for mobile applications.

UNIT IV

TECHNOLOGY I – ANDROID: Introduction – Establishing the development environment – Android Architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment – Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration With social media applications.

UNIT V

TECHNOLOGY II – iOS: Introduction to Objective C – iOS features – UI implementation – Touchframeworks – Data persistence using Core Data and SQLite – Location aware applications using CoreLocation and Map Kit – Integrating calendar and address book with social media application – Using Wifi- iPhone marketplace. Swift: Introduction to Swift features of swift.

TEXT BOOKS:

[1]. Charlie Collins, Michael Galpin and Matthias Kappler, “Android in Practice”, DreamTech, 2012 [2]. AnubhavPradhan , Anil V Despande Composing Mobile Apps,Learn ,explore,apply.

Reference Books:

[3]. James Dovey and Ash Furrow, “Beginning Objective C”, Apress, 2012

[4]. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012

[5]. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, “Beginning iOS 6 Development:Exploring the iOS SDK”, Apress, 2013.

SENSOR NETWORK & IOT LAB

Course Code:	**IT481**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lab Sessions(L + T):	**10 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. To learn different sensors, network and different protocols
2. To implement the different network topologies
3. To implement different wireless sensor networks

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Understand the concepts of Internet of Things
2. Analyze basic protocols in wireless sensor network.
3. Design IoT applications in different domain and be able to analyze their performance
4. Implement basic IoT applications on embedded platform

List of Experiments:-

1. Basics of sensor networks, IoT, 6LoWPAN nodes (used in IoT applications), OS Contiki, Network Simulator COOJA, Download and Installation of Contiki (OS for IoT), Creation of Virtual Machine, Download and Installation of VM Player.
2. Initialization of Network Simulator COOJA, Understanding of all windows on simulator, study the Mote Configuration, Program the Motes so that all motes display “Hello World” on the output window, Change the values in files to display any desired output by all the motes.
3. Create a network topology having 5 motes of similar configuration. Program them to broadcast the data. Capture the broadcasted packets and analyze the values of various

headers like IPv6, using analyzer. Repeat the program by changing the transmission range of all motes and observe the effect.

4. Create a complete wireless sensor network (WSN) topology having 6 motes. Configure 1 mote as Border Router and rest of the 5 motes as sender Motes. Go to the browser and check for the values of the routing table of your WSN.
5. Create a Client-Server network topology having 8 motes. Configure 2 motes as server and 6 motes as client. Capture the packets and generate its pcap files. Analyze the captured packets using packet analyzer tool Wireshark.
6. Study of MoteWorks Network Landscape for deployment of wireless sensor network. Study of TinyOS and nesC.
7. Study of MoteConfig and MoteView. Program the motes using MoteConfig and create a live sensor network. Connect the live sensor network to a local PC and analyze the results using Moteview.
8. To study and verify the self-healing property of wireless sensor network.
9. Introduction to Cisco Packet Tracer and configuring various network devices, hosts & transmission media. Study of components of IoT and basic implementation of IoT network on packet tracer.
10. To deploy a home automation application and perform remote monitoring and control of home appliances.

Mini Project (Not Compulsory): Create an IoT application of Street Light System where the LEDs will glow automatically when luminosity is less than desired value and switch off when sufficient light is present.

INDUSTRIAL TRAINING

Course Code:	**IT491**	**Course Credits:**	**5**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**4U**	**Course Semester**	**7U**

No. of lectures (Hrs/Week):	**-**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**-**	**End Sem. Exam Hours:**	**-**
COURSE OBJECTIVES			
COURSE OUTCOMES			

MINOR PROJECT

Course Code:	**IT493**	**Course Credits:**	**5**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**4U**	**Course Semester**	**7U**
No. of lectures (Hrs/Week):	**-**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**-**	**End Sem. Exam Hours:**	**-**

COURSE OBJECTIVES

COURSE OUTCOMES

SEMESTER-VIII

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT490	Seminar	0	0	3	2	
2	IT492	Major Project	0	0	16	8	
3	IT494	Intenship	0	0	30	15	
4	IT490	Seminar	0	0	3	2	
5	GP	General Proficiency	Non Credit				
Total Hours and Credits			**00**	**0**	**49**	**25**	

SEMINAR

Course Code:	**IT490**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	
Course Year (U / P):	**4U**	**Course Semester**	**8U**
No. of lectures (Hrs/Week):	**-**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**-**	**End Sem. Exam Hours:**	**-**

COURSE OBJECTIVES

COURSE OUTCOMES

MAJOR PROJECT

Course Code:	**IT492**	**Course Credits:**	**8**
Course Category:	**CC**	**Course (U / P)**	
Course Year (U / P):	**4U**	**Course Semester**	**8U**
No. of lectures (Hrs/Week):	**-**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**-**	**End Sem. Exam Hours:**	**-**

COURSE OBJECTIVES

COURSE OUTCOMES

INTERNSHIP			
Course Code:	**IT494**	**Course Credits:**	**15**
Course Category:	**CC**	**Course (U / P)**	
Course Year (U / P):	**4U**	**Course Semester**	**8U**
No. of lectures (Hrs/Week):	**-**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lectures (L + T):	**-**	**End Sem. Exam Hours:**	**-**
COURSE OBJECTIVES			
COURSE OUTCOMES			

ELECTIVES

S. No.	Course Code	Course Name	L	T	P	Credits	Types
1	IT309	IT Forensics	3	0	0	3	E1
2	IT311	Advanced Communication Systems	3	0	0	3	E1
3	IT313	Bio-Informatics	3	0	0	3	E1
4	IT315	Computer Based Numerical & Statistical Techniques	3	0	0	3	E1
5	IT317	Data Compression	3	0	0	3	E2
6	IT319	Software Project Management	3	0	0	3	E2
7	IT321	Information Retrieval System	3	0	0	3	E2
8	IT323	Android	3	0	0	3	E2
9	IT310	Digital Image Processing	3	0	0	3	E3
10	IT312	Distributed Data Bases	3	0	0	3	E3
11	IT314	Expert Systems	3	0	0	3	E3
12	IT316	High Speed Networks	3	0	0	3	E3
13	IT318	Computer Security	3	0	0	3	E4
14	IT320	Management Information System	3	0	0	3	E4
15	IT322	Evolutionary Computation	3	0	0	3	E4
16	IT324	Information Retrieval and Management	3	0	0	3	E4
17	IT407	Graph Theory and Algorithms	3	0	0	3	E5
18	IT409	Mobile Computing	3	0	0	3	E5
19	IT411	Data Warehousing and Data Mining	3	0	0	3	E5
20	IT413	Visual Basic using asp.net	3	0	0	3	E5

OPEN ELECTIVES

S. No.	Course Code	Course Name	L	T	P	Credits	Types
1	ES101	Environmental Studies	3	1	0	4	OE1
2	SS102	History of Science & Technology	3	0	0	3	OE1
3	BSC101	Human Values & Buddhist Ethics	3	0	0	3	OE1
4	BSC201	Introduction to Buddhist Mediation: Theories & Practices	3	0	0	3	OE1
5	BSC305	Buddhist Art & Architecture	3	0	0	3	OE1

** ELECTIVES**

IT Forensics

Course Code:	**IT309**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**0**
Total No. of Lab Sessions(L + T):	**45+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

4. To learn different sensors, network and different protocols

5. To implement the different network topologies

6. To implement different wireless sensor networks

COURSE OUTCOMES	
At the end of the course the students should be able to:	
5.	Understand the concepts of Internet of Things
6.	Analyze basic protocols in wireless sensor network.
7.	Design IoT applications in different domain and be able to analyze their performance
8.	Implement basic IoT applications on embedded platform

UNIT I

Overview of Biometrics, Biometric Identification, Biometric Verification, Biometric Enrollment, Biometric System Security. Authentication and Biometrics: Secure Authentication Protocols, Access Control Security Services, Matching Biometric Samples, and Verification by humans. Common biometrics: Finger Print Recognition, Face Recognition, Speaker Recognition, Iris Recognition, Hand Geometry, Signature Verification

UNIT II

Introduction to Information Hiding: Technical Steganography, Linguistic Steganography, Copy Right Enforcement, Wisdom from Cryptography Principles of Steganography: Framework for Secret Communication, Security of Steganography System, Information Hiding in Noisy Data , Adaptive versus non-Adaptive Algorithms, Active and Malicious Attackers, Information hiding in Written Text.

UNIT III

A Survey of Steganographic Techniques: Substitution systems and Bit Plane Tools, Transform Domain Techniques: - Spread Spectrum and Information hiding, Statistical Steganography, Distortion Techniques, Cover Generation Techniques. Steganalysis: Looking for Signatures: - Extracting hidden Information, Disabling Hidden Information.

UNIT IV

Watermarking and Copyright Protection: Basic Watermarking, Watermarking Applications, Requirements and Algorithmic Design Issues, Evaluation and Benchmarking of Watermarking system. Transform Methods: Fourier Transformation, Fast Fourier Transformation, Discrete Cosine Transformation, Mellin-Fourier Transformation, Wavelets, Split Images in Perceptual Bands. Applications of Transformation in Steganography.

UNIT V

Computer Forensics, Rules of evidence, Evidence dynamics, Evidence collection, Data recovery, Preservation of digital evidence, surveillance tools for future warfare,

Text Books:

[1]. Katzendbisser, Petitcolas, “Information Hiding Techniques for Steganography and Digital Watermarking", Artech House.
[2]. Peter Wayner, "Disappearing Cryptography: Information Hiding, Steganography and Watermarking 2/e", Elsevier

Reference Books:

[3]. Bolle, Connell et. al., "Guide to Biometrics", Springer
[4]. John Vecca, “Computer Forensics: Crime scene Investigation”, Firewall Media 5. Christopher L.T. Brown, “Computer Evidence: Collection and Preservation”, Firewall Media

Advanced Communication Systems			
Course Code:	**IT311**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lab Sessions(L + T):	**45+ 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To learn different communication system like digital, optical and satellite			
2. To implement the different communication system			
3. To learn different spectrums and techniques.			

Advanced Communication Systems			
Course Code:	**IT311**	**Course Credits:**	**3**
COURSE OUTCOMES			
At the end of the course the students should be able to:			
4.Understand the concepts of different communication system			
5.Analyze basic digital system and utility of different methods like Modulation and De-modulation.			
6.Design different communication system according to the requirements			

Unit I

Introduction: Electromagnetic Spectrum, Need of Communication systems, Types of communication systems, Advantages and drawbacks of wireless and wired communication system.

Unit II

Digital Communication Systems:

Baseband modulation and demodulation: Detection of binary signals in Gaussian noise, ISI, Equalization, Carrier and symbol synchronization, Signal design for band limited channels. Band pass modulation and demodulation: Modulation techniques, Coherent and Non coherent detection, Error performance for binary system, Symbol error performance, Communication link Analysis: Link budget analysis, Simple link analysis, System trade-offs. Modulation and coding trade-offs.

Unit III

Satellite communication systems

Origin of Satellite Communications, Historical Back-ground, Basic Concepts of Satellite Communications, Frequency allocations for Satellite Services, Applications, Future Trends of Satellite Communications. INTELSAT Series, INSAT, VSAT, Mobile satellite services: GSM, GPS, INMARSAT, LEO, MEO, Satellite Navigational System. Direct Broadcast satellites (DBS)- Direct to home Broadcast (DTH), Digital audio broadcast (DAB)- Worldspace services, Business TV(BTV), GRAMSAT, Specialized services – E –mail, Video conferencing, Internet

Unit IV

Optical Communication Systems: Optical fibre-step index, graded index, material, preparation, measurement of propagation, properties, jointing, connectors and couplers. Fibre optic communication systems. System model. Optical channel-space, fibre optic, sources-lasers, LEDs. Fibre laser for optical communication through guided media.

Modulation techniques—direct modulation and indirect modulation—injection modulation, A/O, E/O modulation techniques. Optical detection—PIN diodes and APDs. Optical communication systems

Unit V

Advanced Communication networks: Mobile Communication system, Wireless Communication, Optical communication Networks, Hybrid communication systems, Spread Spectrum.

Text Books:

[1]. Bernard Sklar, Digital Communication.

[2]. Simon Haykin, Digital Communication.

Reference Books:

[3]. Satellite Communication by D.C.Aggarwal

[4]. Optical Communication by John M Senior .

BIO-INFORMATICS			
Course Code:	**IT313**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lab Sessions(L + T):	**45+ 00**	**End Sem. Exam Hours:**	**3**

BIO-INFORMATICS

Course Code:	**IT313**	**Course Credits:**	**3**

COURSE OBJECTIVES

1. The objective of this course is to demonstrate significant cell biological principles, quantitative and analytical approaches that enable the students to translate the theoretical foundation in cell biology and genetics into practical understanding
2. To analyze structure of DNA and RNA.
3. To learn Pearl basics

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Understand the concepts of cell, DNA, RNA, pearl basics, and Genome
2. Analyze basic of macromolecular compound, different algorithms and sequences.
3. Design different general data retrieval techniques: indices, Boolean search, fuzzy search

UNIT I: Bioinformatics objectives and overviews, Interdisciplinary nature of Bioinformatics, Data integration, Data analysis, Major Bioinformatics databases and tools. Metadata: Summary & reference systems, finding new type of data online.

Molecular Biology and Bioinformatics: Systems approach in biology, Central dogma of molecular biology, problems in molecular approach and the bioinformatics approach, overview of the bioinformatics applications.

UNIT II: Basic chemistry of nucleic acids, Structure of DNA, Structure of RNA, DNA Replication, Transcription-Translation, Genes- the functional elements in DNA, Analyzing

DNA,DNA sequencing. Proteins: Amino acids, Protein structure, Secondary, Tertiary and Quaternary structure, Protein folding and function, Nucleic acid-Protein interaction.

UNIT III: Perl Basics, Perl applications for bioinformatics- Bioperl, Linux Operating System, mounting/ unmounting files, tar, gzip / gunzip, telnet, ftp, developing applications on Linux

OS, Understanding and Using Biological Databases, Overview of Java, CORBA, XML, Web deployment concepts.

UNIT IV: Genome, Genomic sequencing, expressed sequence tags, gene expression, transcription factor binding sites and single nucleotide polymorphism. Computational representations of molecular biological data storage techniques: databases (flat, relational and object oriented), and controlled vocabularies, general data retrieval techniques: indices, Boolean search, fuzzy search and neighboring, application to biological data warehouses.

UNIT V: Macromolecular structures, chemical compounds, generic variability and its connection to clinical data. Representation of patterns and relationships: sequence alignment algorithms, regular expressions, hierarchies and graphical models, Phylogenetics. BLAST.

Text Books:

[1]. D E Krane & M L Raymer," Fundamental concepts of Bioinformatics", Perason Education.

[2]. Rastogi, Mendiratta, Rastogi, "Bioinformatics Methods & applications, Genomics, Proteomics & Drug Discovery" PHI, New Delhi

Reference Books:

[3]. Shubha Gopal et.al. "Bioinformatics: with fundamentals of genomics and proteomics", Mc Graw Hill.

[4]. O'Reilly, "Developing Bio informatics computer skills", CBS

[5]. Forsdyke, "Evolutionary Bioinformatics",

Computer Based Numerical & Statistical Techniques

Course Code:	**IT315**	**Course Credits:**	**3**
Course Category:	**E1**	**Course (U / P)**	**P**
Course Year (U / P):	**3U**	**Course Semester**	**5U**
No. of Lab (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lab Sessions(L + T):	**45+ 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

1. The objective of this course to Obtain an intuitive and working understanding of numerical methods for the basic problems of numerical analysis.
2. To Gain experience in the implementation of numerical methods using a computer.
3. To learn various significant and fundamental concepts to inculcate in the students an adequate understanding of the application of Statistical Methods

COURSE OUTCOMES

At the end of the course the students should be able to:

1. Trace errors in Numerical methods and analyze and predict it.
2. Analyze basic of application of Statistical methods.
3. Discuss concepts of numerical methods used for different applications

Unit-I

Floating point Arithmetic: Representation of floating point numbers, Operations, Normalization, Pitfalls of floating point representation, Errors in numerical computation Iterative Methods: Zeros of a single transcendental equation and zeros of polynomial using Bisection Method, Iteration Method, Regula-Falsi method, Newton Raphson method, Secant method, Rate of convergence of iterative methods.

Unit-II

Simultaneous Linear Equations: Solutions of system of Linear equations, Gauss Elimination direct method and pivoting, Ill Conditioned system of equations, Refinement of solution. Gauss

Seidel iterative method, Rate of Convergence Interpolation and approximation: Finite Differences, Difference tables Polynomial Interpolation: Newton's forward and backward formula Central Difference Formulae: Gauss forward and backward formula, Stirling's, Bessel's, Everett's formula. Interpolation with unequal intervals: Lagrange's Interpolation, Newton Divided difference formula, Hermite's Interpolation Approximation of function by Taylor's series and Chebyshev polynomial

Unit-III

Numerical Differentiation and Integration: Introduction, Numerical Differentiation, Numerical Integration, Trapezoidal rule, Simpson's rules, Boole's Rule, Weddle's Rule Euler- Maclaurin Formula Solution of differential equations: Picard's Method, Euler's Method, Taylor's Method, Runge-Kutta methods, Predictor-corrector method, Automatic error monitoring, stability of solution.

Unit-IV

Curve fitting, Cubic Spline and Approximation: Method of least squares, fitting of straight lines, polynomials, exponential curves etc. Frequency Chart: Different frequency charts like Histogram, Frequency curve, Pi-chart. Regression analysis: Linear and Non-linear regression, Multiple regression

Unit-V

Time series and forecasting: Moving averages, smoothening of curves, forecasting models and methods. Statistical Quality Controls methods Testing of Hypothesis: Test of significance, Chi-square test, t-test, ANOVA, F-Test Application to medicine, agriculture etc.

References Books:

[1] Rajaraman V., "Computer Oriented Numerical Methods", PHI
[2]Gerald & Wheatley, "Applied Numerical Analyses", AW
[3]Jain, Iyengar and Jain, "Numerical Methods for Scientific and Engineering Computations", New Age Int.
[4] Grewal B. S., "Numerical methods in Engineering and Science", Khanna Publishers, Delhi
[5] T. Veerarajan, T Ramachandran, "Theory and Problems in Numerical Methods", TMH
[6] Pradip Niyogi, "Numerical Analysis and Algorithms", TMH
[9] Francis Scheld, "Numerical Analysis", TMH
[10] Gupta S. P., "Statistical Methods", Sultan and Sons

DATA COMPRESSION			
Course Code:	**IT317**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Present the basics of data compression approach			
2 Helps in learning step by step.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of paradigms of data compression techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of data compression			
2 Apply methods.			
3 learn various techniques.			
4 Practical Approach			

Unit - I:

Compression Techniques: Loss less compression, Lossy Compression, Measures of performance, Modeling and coding, Mathematical Preliminaries for Lossless compression: A brief introduction to information theory, Models: Physical models, Probability models, Markov models, composite source model, Coding: uniquely decodable codes, Prefix codes.

Unit – II:

The Huffman coding algorithm: Minimum variance Huffman codes, Adaptive Huffman coding: Update procedure, Encoding procedure, Decoding procedure. Golomb codes, Rice codes, Tunstall codes, Applications of Hoffman coding: Loss less image compression, Text compression, Audio Compression.

Unit-III:

Coding a sequence, Generating a binary code, Comparison of Binary and Huffman coding, Applications: Bi-level image compression-The JBIG standard, JBIG2, Image compression. Dictionary Techniques: Introduction, Static Dictionary: Diagram Coding, Adaptive Dictionary. The LZ77 Approach, The LZ78 Approach, Applications: File

Compression-UNIX compress, Image Compression: The Graphics Interchange Format (GIF), Compression over Modems: V.42 bits, Predictive Coding: Prediction with Partial match (ppm): The basic algorithm, The ESCAPE SYMBOL, length of context, The Exclusion Principle, The Burrows-Wheeler Transform: Move-to-front coding, CALIC, JPEG-LS, Multi-resolution Approaches, Facsimile
Encoding, Dynamic Markoy Compression.

Unit – IV:
Distortion criteria, Models, Scalar Ouantization: The Quantization problem, Uniform Quantizer,
Adaptive Quantization, Non uniform Quantization.

Unit-V:
Advantages of Vector Quantization over Scalar Quantization, The Linde-Buzo-Gray Algorithm, Tree structured Vector Quantizers. Structured VectorQuantizers.

Text Books:
[1]. Khalid Sayood, Introduction to Data Compression, Morgan Kaufmann Publishers
[2]. Drozdek , Elements of Data Compression, Cengage Learning

Reference Books:
[3]. Data Compression: The Complete Reference 4th Edition by David Salomon, Springer

SOFTWARE PROJECT MANAGEMENT			
Course Code:	**IT319**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Present the basic software project management approach
2 Helps in learning step by step.
3 Understanding of Methods.
4 Understanding of learning the process and application
5 Understanding of paradigms of software project management techniques.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Will have basic knowledge and understanding of software project management.
2 Apply methods.
3 learn various techniques.
4 Practical Approach

UNIT-I: Introduction and Software Project Planning

Fundamentals of Software Project Management (SPM), Need Identification, Vision and Scope document, ProjectManagement Cycle, SPM Objectives, Management Spectrum, SPM Framework, Software Project Planning,Planning Objectives, Project Plan, Types of project plan, Structure of a Software Project Management Plan,Software project estimation, Estimation methods, Estimation models, Decision process.

UNIT-II: Project Organization and Scheduling

Project Elements, Work Breakdown Structure (WBS), Types of WBS, Functions, Activities and Tasks, Project LifeCycle and Product Life Cycle, Ways to Organize Personnel, Project schedule, Scheduling Objectives, Building theproject schedule, Scheduling terminology and techniques, Network Diagrams: PERT, CPM, Bar Charts: MilestoneCharts, Gantt Charts.

UNIT-III: Project Monitoring and Control

Dimensions of Project Monitoring & Control, Earned Value Analysis, Earned Value Indicators: Budgeted Cost forWork Scheduled (BCWS), Cost Variance (CV), Schedule Variance (SV), Cost Performance Index (CPI), SchedulePerformance Index (SPI), Interpretation of Earned Value Indicators, Error Tracking, Software Reviews, Types ofReview: Inspections, Deskchecks, Walkthroughs, Code Reviews, Pair Programming.

UNIT-IV: Software Quality Assurance and Testing

Testing Objectives, Testing Principles, Test Plans, Test Cases, Types of Testing, Levels of Testing, Test Strategies,Program Correctness, Program Verification & validation, Testing Automation & Testing Tools, Concept ofSoftware Quality, Software Quality Attributes, Software Quality Metrics and Indicators, The SEI CapabilityMaturity Model CMM), SQA

Activities, Formal SQA Approaches: Proof of correctness, Statistical qualityassurance, Cleanroom process.

UNIT-V: Project Management and Project Management Tools

Software Configuration Management: Software Configuration Items and tasks, Baselines, Plan for Change, ChangeControl, Change Requests Management, Version Control, Risk Management: Risks and risk types, Risk BreakdownStructure (RBS), Risk Management Process: Risk identification, Risk analysis, Risk planning, Risk monitoring, CostBenefit Analysis, Software Project Management Tools: CASE Tools, Planning and Scheduling Tools, MS-Project.

Text Books:

[1]. M. Cotterell, Software Project Management, Tata McGraw-Hill Publication.

[2]. Royce, Software Project Management, Pearson Education

Reference Books:

[3]. Kieron Conway, Software Project Management, Dreamtech Press

[4]. S. A. Kelkar, Software Project Management, PHI Publication.

[5]. Timothy C. Bell, T ext Compression1st Edition , Prentice Hall

INFORMATION RETRIEVAL SYSTEMS			
Course Code:	**IT321**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To understand the theoretical basis behind the standard models of IR (Boolean, Vector-space, Probabilistic and Logical models)			
2. To understand the difficulty of representing and retrieving documents, images, speech, etc.			
3.To understand the standard methods for Web indexing and retrieval			
4.To understand how techniques from natural language processing, artificial intelligence, human computer interaction, and visualization integrate with IR			
5.To be familiar with various algorithms and systems			

COURSE OUTCOMES
At the end of the course the students should be able to:
1.Learn the theories and techniques behind Web search engines, E-commerce recommendation systems, etc.
2.Get hands on project experience by developing real-world applications, such as intelligent tools for improving search accuracy from user feedback, email spam detection, recommendation system, or scientific literature organization and mining.
3.Learn tools and techniques to do cutting-edge research in the area of information retrieval or text mining.
4.be able to implement, run and test a standard IR system
5.Open the door to the amazing job opportunities in Search Technology and E-commerce companies such as Google, Microsoft, Yahoo!, and Amazon

UNIT I: Basic Concepts of IR, Data Retrieval & Information Retrieval, IR system block diagram. Automatic Text Analysis, Luhn's ideas, Conflation Algorithm, Indexing and Index Term Weighing, Probabilistic Indexing, Automatic Classification. Measures of Association, Different Matching Coefficient, Classification Methods, Cluster Hypothesis. Clustering Algorithms, Single Pass Algorithm, Single Link Algorithm, Rochhio's Algorithm and Dendograms

UNIT II: File Structures, Inverted file, Suffix trees &suffix arrays, Signature files, Ring Structure, IR Models, Basic concepts, Boolean Model, Vector Model, and Fuzzy Set Model. Search Strategies, Boolean search, serial search, and clusterbased retrieval, Matching Function. Performance Evaluation- Precision and recall, alternative measures reference collection (TREC Collection), Libraries & Bibliographical system- Online IR system, OPACs, Digital libraries - Architecture issues, document models, representation & access, Prototypes, projects & interfaces, standards.

UNIT III: Taxonomy and Ontology: Creating domain specific ontology, Ontology life cycle Distributed and Parallel IR: Relationships between documents, Identify appropriate networked collections, multiple distributed collections simultaneously, Parallel IR - MIMD Architectures, Distributed IR
Collection Partitioning, Source Selection, Query Processing.

UNIT IV: Multimedia IR models & languages- data modeling, Techniques to represent audio and visual document, query languages Indexing & searching- generic multimedia indexing approach, Query databases of multimedia documents, Display the results of multimedia searches, one dimensional time series, two dimensional color images, automatic feature extraction.

UNIT V : Searching the Web, Challenges, Characterizing the Web, Search Engines, Browsing, Mata searchers, Web crawlers, robot exclusion, Web data mining, Metacrawler, Collaborative filtering, Web agents (web shopping, bargain finder), Economic, ethical, legal and political issues.

Text Books:

[1]. Yates & Neto, "Modern Information Retrieval", Pearson Education, ISBN 81-297-0274-6

[2]. I. Witten, A. Moffat, and T. Bell, "Managing Gigabytes" 4. D. Grossman and O. Frieder "Information Retrieval: Algorithms and Heuristics"

Reference Books:

[3]. Mark leven, "Introduction to search engines and web navigation", John Wiley andsons Inc., ISBN 9780-170-52684-2.

[4]. V. S. Subrahamanian, Satish K. Tripathi "Multimedia information System", Kulwer Academic Publisher

[5]. Chabane Djeraba,"Multimedia mining A highway to intelligent multimedia

ANDROID			
Course Code:	**IT323**	**Course Credits:**	**3**
Course Category:	**E2**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**5U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 Present the basics of android approach			
2 Helps in learning step by step.			
3 Understanding of Methods.			
4 Understanding of learning the process and application			
5 Understanding of paradigms of android techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1 Will have basic knowledge and understanding of android technology.			
2 Apply methods.			

3 learn various techniques.
4 Practical Approach

Unit 1: Introduction: What is Android, Android versions and its feature set The various Android devices on the market , The Android Market application store , Android Development Environment - System Requirements, Android SDK, Installing Java, and ADT bundle - Eclipse Integrated Development Environment (IDE), Creating Android Virtual Devices (AVDs)

Unit 2: Android Architecture Overview and Creating an Example Android Application: The Android Software Stack, The Linux Kernel, Android Runtime - Dalvik Virtual Machine, Android Runtime – Core Libraries, Dalvik VM Specific Libraries, Java Interoperability Libraries, Android Libraries, Application Framework, Creating a New Android Project ,Defining the Project Name and SDK Settings, Project Configuration Settings, Configuring the Launcher Icon, Creating an Activity, Running the Application in the AVD, Stopping a Running Application, Modifying the Example Application, Reviewing the Layout and Resource Files,

Unit 3: Android Software Development Platform Understanding Java SE and the Dalvik Virtual Machine , The Directory Structure of an Android Project , Common Default Resources Folders, The Values Folder , Leveraging Android XML, Screen Sizes , Launching Your Application: The AndroidManifest.xml File, Creating Your First Android Application

Unit 4: Android Framework Overview: Android Application Components, Android Activities: Defining the UI, Android Services: Processing in the Background, Broadcast Receivers: Announcements and Notifications Content Providers: Data Management, Android Intent Objects: Messaging for Components Android Manifest XML: Declaring Your Components

UNIT 5: Text controls Button controls Images Supporting Multiple Screen, Activities. Application context. Intent WebView. List View. Spinner AutoComplete Textview. Multi Auto Complete Textview. Toast. Dialogue Notification. Status bar Notification. Option Menu. Context Menu. Google Map. File. Shared Preferences. Database (SQLite database) Creation of .apk files. CHAPTER 11: Web services. HTTP client. XML and JSON.

References:

www.tutorialspoint.com › android

DIGITAL IMAGE PROCESSING			
Course Code:	**IT310**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for image processing			
2. To study the image enhancement techniques			
3. To study image restoration procedures			
4. To study the image compression procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of a digital image processing system.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for image enhancement and image restoration			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

Unit 1: Digital Image Fundamentals and Transforms:

Elements of visual perception – Image sampling and quantization Basic relationship between pixels – Basic geometric transformations-Introduction to Fourier Transform and DFT – Properties of 2D Fourier Transform – FFT– Separable Image Transforms -Walsh – Hadamard – Discrete Cosine Transform, Haar, Slant – Karhunen – Loeve transforms.

UNIT II: Image Enhancement Techniques

Spatial Domain methods: Basic grey level transformation – Histogram equalization – Image subtraction – Image averaging –Spatial filtering: Smoothing, sharpening filters – Laplacian filters – Frequency domain filters: Smoothing – Sharpening filters – Homomorphic filtering.

UNIT III: Image Restoration and Compression

Model of Image Degradation/ restoration process – Noise models, restoration by spatial filtering, Inverse filtering -Least mean square filtering –Constrained least mean square filtering – Blind image restoration – Pseudo inverse – Singular value decomposition Wiener filtering, image reconstruction from Projection.

Unit IV: Image Compression

Fundamentals of data compression- basic compression methods: Huffman coding, Golomb coding, LZW coding, Run-Length coding, Symbol based coding. Digital Image Watermarking, Representation and Description- minimum perimeter polygons algorithm (MPP).

Unit V: Image Segmentation and Representation

Edge detection–Thresholding-Region Based segmentation – Boundary representation: chair codes- Polygonal approximation–Boundary segments – boundary descriptors: Simple descriptors-Fourier descriptors – Regional descriptors –Simple descriptors-Texture.

Text Books:

[1] Rafael C Gonzalez & Richard E Woods, Digital Image Processing, Pearson Education.

[2] William K Pratt, Digital Image Processing, John Willey.

Reference Books:

[3] A.K. Jain, Fundamentals of Digital Image Processing, PHI.

[4] Chanda Dutta Magundar, Digital Image Processing and Applications, Prentice Hall of India.

DISTRIBUTED DATABASE			
Course Code:	**IT312**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for distributed database			
2. To study the image enhancement techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of a digital image processing system.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for image enhancement and image restoration			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

UNIT I : Transaction and schedules, Concurrent Execution of transaction, Conflict and View Serializability, Testing for Serializability,Concepts in Recoverable and Cascadeless schedules.

UNIT II : Lock based protocols, time stamp based protocols, Multiple Granularity and Multiversion Techniques, Enforcing serializablity byLocks, Locking system with multiple lock modes, architecture forLocking scheduler.

UNIT III : Distributed Transactions Management, Data Distribution, 8Fragmentation and Replication Techniques, Distributed Commit,Distributed Locking schemes, Long duration transactions, MossConcurrency protocol.

UNITIV: Issues of Recovery and atomicity in Distributed Databases, Traditional recovery techniques, Log based recovery, Recovery withConcurrent Transactions, Recovery in Message passing systems,Checkpoints, Algorithms for recovery line, Concepts in Orphan andInconsistent Messages.

UNIT V : Distributed Query Processing, Multiway Joins, Semi joins, Cost based query optimization for distributed database, Updatingreplicated data, protocols for Distributed Deadlock Detection, Eagerand Lazy Replication Techniques.

Text Books:

[1]. Silberschatz,korth and Sudershan, Database System Concept', Mc Graw Hill
[2]. Ramakrishna and Gehrke,' Database Management System, Mc Graw Hill

References Books:

[3]. Garcia-Molina, Ullman,Widom,' Database System Implementation' Pearson Education
[4]. Ceei and Pelagatti,'Distributed Database', TMH
[5]. Singhal and Shivratri, 'Advance Concepts in Operating Systems' MC Graw Hill

EXPERT SYSTEMS			
Course Code:	**IT314**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			

4. To study the procedures.
5.To understand image segmentation and representation techniques.
COURSE OUTCOMES
At the end of the course the students should be able to:
1. Review the fundamental concepts of expert systems.
2. Analyze images in the frequency domain using various transforms.
3. Evaluate the techniques for fuzzy logic
4. Categorize various compression techniques
5. Interpret Image compression standards.

UNIT I: The meaning of an expert system, problem domain and knowledge domain, the advantages of an expert system, general stages in the development of an expert system, general characteristics of an expert system, history and uses of expert systems today, rule-based expert systems, procedural and nonprocedural paradigms, characteristics of artificial neural systems. -The study of logic, difference between formal logic and informal logic, meaning of knowledge, how knowledge can be represented.

UNIT II: Semantic nets, how to translate semantic nets into PROLOG, limitations of semantic nets, schemas, frames and their limitations, how to use logic and set symbols to represent knowledge, the meaning of propositional and first order predicate logic, quantifiers, imitations of propositional and predicate logic. Trees, lattices, and graphs, state and problem spaces, AND-OR trees and goals, methods of inference, rules of inference, limitations of propositional logic, logic systems, resolution rule of inference, resolution systems, and deduction, shallow and causal reasoning, applying resolution to first-order predicate logic, forward and backward chaining, additional methods of reference, Meta knowledge, the Markov decision process.

UNIT III: The meaning of uncertainty and theories devised to deal with it, types of errors attributed to uncertainty, errors associate, with induction, features of classical probability, experimental and subjective probabilities, compound and conditional probabilities, hypothetical reasoning and backward induction, temporal reasoning.

UNIT IV: Markov chains, odds of belief, sufficiency and necessity, role of uncertainty in inference chains, implications of combining evidence, role of inference nets in expert systems, how probabilities are propagated. Sources of uncertainty in rules, methods of dealing with uncertainty, Dempster-Shafer theory, theory of uncertainty based on fuzzylogic, commercial applications of fuzzy logic. How to select an appropriate problem,

UNIT V: Stages in the development of an expert system, types of errors to expect in the development stages, the role of the knowledge engineer in the building of expert systems, the expected life cycle of an expert system, how to do a life cycle model.

Textbook:

[1] J. Giarratano and G. Riley, "Expert Systems -- Principles and Programming". 4th Edition, PWS Publishing Company, 2004.

[2] Durkin, J., Expert systems Design and Development, Macmillan, 1994 2. Elias M. Awad, Building Expert Systems, West Publishing Company 1996

HIGHSPEED NETWORKS			
Course Code:	**IT316**	**Course Credits:**	**3**
Course Category:	**E3**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

UNIT I

Frame Relay Networks – Asynchronous transfer mode–ATM Protocol Architecture, ATM logical Connection – ATM Cell – ATM Service Categories – AAL. High Speed LAN's: Fast Ethernet – Gigabit Ethernet– Fiber Channel – Wireless LAN's, WiFiand WiMax Networks applications, requirements –Architecture of 802.11.

UNIT II

Queuing Analysis – Queuing Models – Single Server Queues – Effects of Congestion CongestionControl – Traffic Management – Congestion Control in Packet Switching Networks – FrameRelay Congestion Control.

UNIT III

TCP Flow control – TCP Congestion Control – Retransmission – Timer Management –Exponential RTObackoff – KARN's Algorithm – Window management – Performance ofTCP over ATM. Traffic andCongestion control in ATM – Requirements – Attributes –Traffic Management Frame work, TrafficnControl – ABR traffic Management – ABR ratecontrol, RM cell formats – ABR Capacity allocations –GFR traffic management.

UNIT IV

Integrated Services Architecture – Approach, Components, Services- Queuing Discipline– FQ – PS –BRFQ – GPS – WFQ – Random Early Detection – Differentiated Services.

UNIT V

RSVP – Goals & Characteristics, Data Flow, RSVP operations – Protocol Mechanisms– MultiprotocolTransfer Protocol– RTCP. **TOTAL** Label Switching – Operations, Label Stacking – Protocol details – RTP– Protocol Architecture – Data

.

Text Books:

[1]. William Stallings, "High speed networks and internet", Second Edition, Pearson Education, 2002

[2]. Warland, Pravin Varaiya, "High performance communication networks", Second Edition, JeanHarcourt Asia Pvt. Ltd., , 2001

Reference Books:

[3]. Irvan Pepelnjk, Jim Guichard, Jeff Apcar, "MPLS and VPN architecture", Cisco Press, Volume 1and 2, 2003.

[4]. Abhijit S. Pandya, Ercan Sea, "ATM Technology for Broad Band Telecommunication Networks", CRC Press, New York, 2004

COMPUTER SECURITY			
Course Code:	**IT318**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**
Course Year (U / P):	**3U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

MANAGEMENT INFORMATION SYSTEM			
Course Code:	**IT320**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

EVOLUTIONARY COMPUTING			
Course Code:	**IT322**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards			

INFORMATION RETRIEVAL & MANAGEMENT			
Course Code:	**IT324**	**Course Credits:**	**3**
Course Category:	**E4**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**6U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

UNIT I: Basic Concepts of IR, Data Retrieval & Information Retrieval, IR system block diagram. Automatic Text Analysis, Luhn's ideas, Conflation Algorithm, Indexing and Index Term Weighing, Probabilistic Indexing, Automatic Classification. Measures of Association, Different MatchingCoefficient, Classification Methods, Cluster Hypothesis. Clustering Algorithms, Single Pass Algorithm, Single Link Algorithm, Rochhio's Algorithm and Dendograms.

UNIT II: File Structures, Inverted file, Suffix trees &suffix arrays, Signature files, Ring Structure, IR Models, Basic concepts, Boolean Model, Vector Model, and Fuzzy Set Model. Search Strategies, Boolean search, serial search, and clusterbased retrieval, Matching Function. Performance Evaluation- Precision and recall, alternative measures reference collection (TREC Collection), Libraries & Bibliographical system- Online IR system, OPACs, Digital libraries - Architecture issues,document models, representation & access, Prototypes, projects & interfaces, standards.

UNIT III: Taxonomy and Ontology: Creating domain specific ontology, Ontology life cycle Distributed and Parallel IR: Relationships between documents, Identify appropriate networked collections, multiple distributed collections simultaneously, Parallel IR - MIMD Architectures, Distributed IR
Collection Partitioning, Source Selection, Query Processing.

UNIT IV: Multimedia IR models & languages- data modeling, Techniques to represent audio and visual document, query languages Indexing & searching- generic multimedia indexing approach, Query databases of multimedia documents, Display the results of multimedia searches, onedimensional time series, two dimensional color images, automatic feature extraction.

UNIT V : Searching the Web, Challenges, Characterizing the Web, Search Engines, Browsing, Mata searchers, Web crawlers, robot exclusion, Web data mining, Metacrawler, Collaborative filtering, Web agents (web shopping, bargain finder), Economic, ethical, legal and political issues.

Text Books:

[1]. Yates & Neto, "Modern Information Retrieval", Pearson Education, ISBN 81-297-0274-6

[2]. I. Witten, A. Moffat, and T. Bell, “Managing Gigabytes” 4. D. Grossman and O. Frieder “Information Retrieval: Algorithms and Heuristics”

Reference Books:

[3]. Mark leven, “Introduction to search engines and web navigation”, John Wiley andsons Inc., ISBN 9780-170-52684-2.

[4]. V. S. Subrahamanian, Satish K. Tripathi “Multimedia information System”,Kulwer Academic Publisher

[5]. Chabane Djeraba, ”Multimedia mining A highway to intelligent multimedia

GRAPH THEORY & AlGORITHMS			
Course Code:	**IT 407**	**Course Credits:**	**3**
Course Category:	**E5**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 + 00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1 To understand the concept of vertex connectivity and edge connectivity in graphs.			
2 To develop the under-standing of Geometric duals in Planar Graphs			
3To understand the concept of digraphs, Euler digraphs and Hamiltonian digraphs			
4 To understand the concept of matrices in graphs like Incidence matrix, Adjacency matrix, Cycle matrix etc			

5 To introduce the idea of coloring in graphs.
COURSE OUTCOMES
At the end of the course the students should be able to:
1 Understand and explain the basic concepts of graph theory.
2 Apply the basic concepts of mathematical logic.
3 Compute the maximum flow in networks and find its application
4 Understanding the concepts of connectivity and edge connectivity.
5 Identify maximum matching in a bipartite graph

Unit-I: INTRODUCTION

Graphs, sub graphs, vertex degrees, walks, path, cycles and trails, connected graphs, disconnected graphs and components, matrix representation of graphs, isomorphism, Euler graphs, Hamiltonian paths and circuits, bipartite graphs.

Unit-II: TREES AND CONNECTIVITY

Trees─rooted, binary trees and spanning trees, bridges, fundamental circuits, distance, center, diameter, eccentricity, radius and pendent vertices, Prim's, Kruskal's and Dijkstra's Algorithms, cut vertices, blocks and connectivity,

Unit-III: PLANARITY, EULER TOURS AND HAMILTONIAN CYCLES

Planer graphs – Different representation of a planer graph, discussion on criterion of planarity, thickness and crossings, Euler's formula, Platonic bodies, combinatorial and geometric dual: Kuratowski's graphs, detection of planarity, geometric dual, Euler tours, Hamiltonian cycles and travelling salesman problem.

Unit –IV: DIRECTED GRAPH AND COLORING

Directed graphs─definitions, in-degree, out-degree, orientations and tournaments, Coloring─vertex coloring, edge coloring, covering and partitioning of a graph, chromatic number, chromatic partitioning, chromatic polynomials, and four color problem.

Unit –V: EXTREMAL PROBLEMS

Enumeration of graphs, Ramsey's theorem, Ramsey numbers, edge Ramsey numbers, a generalization of party problem, Sperner's lemma and their applications.

Text Books:

1. West, Douglas Brent. *Introduction to graph theory*. Vol. 2. Upper Saddle River: Prentice hall, 2001.
2. Clark, John, and Derek Allan Holton. *A first look at graph theory*. Vol. 1. Teaneck, NJ: World Scientific, 1991.

Reference Books:

1. Deo, Narsingh. *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2016.
2. Chartrand, Gary. *Introduction to graph theory*. Tata McGraw-Hill Education, 2006.

MOBILE COMPUTING			
Course Code:	**IT409**	**Course Credits:**	**3**
Course Category:	**E5**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1.To study the image fundamentals and mathematical transforms necessary for expert system			
2. To study the expert system techniques			
3. To study diffrenent protocol procedures			
4. To study the procedures.			
5.To understand image segmentation and representation techniques.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Review the fundamental concepts of expert systems.			
2. Analyze images in the frequency domain using various transforms.			
3. Evaluate the techniques for fuzzy logic			
4. Categorize various compression techniques			
5. Interpret Image compression standards.			

UNIT I

INTRODUCTION: Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements Gathering and validation for mobile applications

UNIT II

BASIC DESIGN: Introduction – Basics of embedded systems design – Embedded OS – Design Constraints for mobile applications, both hardware and software related – Architecting mobile applications – User interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

UNIT III

ADVANCED DESIGN: Designing applications with multimedia and web access capabilities –

Integration with GPS and social media networking applications – Accessing applications hosted in cloud computing environments – Design patterns for mobile applications.

UNIT IV

TECHNOLOGY IN – ANDROID: Introduction – Establishing the development environment – Android Architecture – Activities and views – Interacting with UI – Persisting data using

SQLite – Packaging and deployment – Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration With social media applications.

UNIT V

TECHNOLOGY II – iOS: Introduction to Objective C – iOS features – UI implementation – Touch Frameworks – Data persistence using Core Data and SQLite – Location aware applications using CoreLocation and Map Kit – Integrating calendar and address book with social media application – Using Wifi- iPhone marketplace. Swift: Introduction to Swift features of swift.

TEXT BOOKS:

[1]. Charlie Collins, Michael Galpin and Matthias Kappler, "Android in Practice", DreamTech, 2012

[2]. AnubhavPradhan , Anil V Deshpande Composing Mobile Apps,Learn ,explore,apply.

Reference Books:

[3]. James Dovey and Ash Furrow, "Beginning Objective C", Apress, 2012

[4]. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012

[5]. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, "Beginning iOS 6 Development:Exploring the iOS SDK", Apress, 2013.

DATA WAREHOUSING & MINING			
Course Code:	**IT411**	**Course Credits:**	**3**
Course Category:	**E5**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To Understand Data mining principles and techniques.			
2. To Understand DM as a cutting edge business intelligence.			
3. To expose the students to the concepts of Data ware housing Architecture and Implementation.			
4. To study the overview of developing areas – Web mining, Text mining and ethical aspects of Data mining.			
5. To identify Business applications and Trends of Data mining.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Perform the preprocessing of data and apply mining techniques on it.			
2. Identify the association rules, classification, and clusters in large data sets.			

3. Solve real world problems in business and scientific information using data mining.
4. Use data analysis tools for scientific applications.
5. Implement various supervised machine learning algorithms.

Unit-1: Data warehousing Definition, usage and trends. DBMS vs. data warehouse, Data marts, Metadata, Multidimensional data mode, Data cubes, Schemas for Multidimensional Database: stars, snowflakes and fact constellations.

Unit-2: Data warehouse process & architecture, OLTP vs. OLAP, ROLAP vs. MOLAP, types of OLAP, servers, 3-Tier data warehouse architecture, distributed and virtual data warehouses, data warehouse manager.

Unit-3: Data warehouse implementation, computation of data cubes, modeling OLAP data, OLAP queries manager, data warehouse back end tools, complex aggregation at multiple granularities, tuning and testing of data warehouse.

Unit-4: Data mining definition & task, KDD versus data mining, data mining techniques, tools and applications. Data mining query languages, data specification, specifying knowledge, hierarchy specification, pattern presentation & visualization specification, data mining languages and standardization of data mining.

Unit-5: Data mining techniques: Association rules, Clustering techniques, Decision tree knowledge discovery through Neural Networks & Genetic Algorithm, Rough Sets, and Support Victor Machines and Fuzzy techniques. Mining complex data objects, Spatial databases, Multimedia databases, Time series and Sequence data; mining Text Databases and mining Word Wide Web.

TEXT BOOKS:

[1] Sam Anahory & Dennis Murray, Data Warehousing In the Real World, Pearson, 1997

[2] Jiawei Han & Micheline Kamber, Data Mining- Concepts & Techniques, Morgan Kaufmann, 2001.

[3] Arun Pujar, Data Mining Techniques, University Press; Hyderabad, 2001,.

REFERENCE BOOKS:

[4] Pieter Adriaans & Dolf Zantinge, Data Mining, Pearson, 1997.

[5] Alex Berson, Data Warehousing, Data Mining and OLTP, Mc Graw Hill, 1997. □

[6] Mallach, Data warehousing System, Mc Graw Hill, 2000.

[7] W.H. Inman, Building the Data Warehouse, John Wiley & Sons, 1996.

[8] W.H Ionhman, C.Klelly, Developing the Data Warehouses, John Wiley & Sons.

[9] W.H.Inman, C.L.Gassey, Managing the Data Warehouses, John Wiley & Sons.

VISUAL BASIC ASP .NET			
Course Code:	**IT413**	**Course Credits:**	**3**
Course Category:	**E5**	**Course (U / P)**	**U**
Course Year (U / P):	**4U**	**Course Semester (U / P):**	**7U**
No. of Lectures + Tutorials (Hrs/Week):	**03 +00**	**Mid Sem. Exam Hours:**	**1**
Total No. of Lectures (L + T):	**45 + 00**	**End Sem. Exam Hours:**	**3**
COURSE OBJECTIVES			
1. To Understand Data mining principles and techniques.			
2. To Understand DM as a cutting edge business intelligence.			
3. To expose the students to the concepts of Data warehousing Architecture and Implementation.			
4. To study the overview of developing areas – Web mining, Text mining and ethical aspects of Data mining.			
5. To identify Business applications and Trends of Data mining.			
COURSE OUTCOMES			
At the end of the course the students should be able to:			
1. Perform the preprocessing of data and apply mining techniques on it.			
2. Identify the association rules, classification, and clusters in large data sets.			
3. Solve real world problems in business and scientific information using data mining.			
4. Use data analysis tools for scientific applications.			
5. Implement various supervised machine learning algorithms.			