UNIVERSITY SCHOOL OF
INFORMATION AND COMMUNICATION TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAMME STRUCTURE

M.TECH. COMPUTER SCIENCE AND ENGINEERING

SPECIALIZATION: ARTIFICIAL INTELLIGENCE AND ROBOTICS

2021-2023

GAUTAM BUDDHA UNIVERSITY
GAUTAM BUDH NAGAR, GREATER NOIDA, UP, INDIA

SEMESTER I

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CS521	Advanced Data Base Management System	3	0	0	3	CC1
2	CS523	Design and Architecture for Software Systems	3	0	0	3	CC2 / FC
3	CS525	Advanced Data Structure and Algorithm	3	1	0	4	CC3
4	CS527	Research Techniques in ICT	3	0	0	3	CC4
5	CS529	Java Programming	3	0	0	3	CC5
6	ES415	Energy and Environment	3	0	0	3	OE1 /AECC
7	EN531	Language, Culture and Society	3	0	0	3	OE2 /AECC
8	CS581	Advanced Data Base Management System Lab	0	0	3	2	CC-L1
9	CS583	Java Programming Lab	0	0	3	2	CC-L2 / SEC
10	GP	General Proficiency	Non-Credit				
		Total Hours and Credits	**21**	**1**	**6**	**26**	

SEMESTER II

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CS522	Python Programming	3	0	0	3	CC6
2	CA522	Soft Computing Techniques	3	0	0	3	CC7
3	CA524	Knowledge Engineering	3	0	0	3	CC8
4	CA526	Machine Translation and Learning	3	0	0	3	CC9
5		Elective-1	3	0	0	3	E1 / DSE
6		Elective-2	3	0	0	3	E2 / DSE
7		Generic Elective	3	1	0	4	GE1
8	CS582	Python Programming lab	0	0	3	2	CC-L3 / SEC
9	CA582	Soft Computing lab	0	0	3	2	CC-L4
10	GP	General Proficiency	Non-Credit				
		Total Hours and Credits	**21**	**1**	**6**	**26**	

Summer Project in Industry will be done individually after Fourth Year of Integrated B.Tech. - M.Tech. CSE and Second Year of M.Tech. during the summer break and it will be of minimum 4 weeks. It will be evaluated as per University Examination Rules in III semester of M.Tech. and IX semester for Integrated B.Tech. - M.Tech. CSE.

SEMESTER III

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CA621	Expert System Design	3	0	0	3	CC10
2	CA623	Robotics	3	0	0	3	CC11
3		Elective-3	3	0	0	3	E3 / DSE
4		Elective-4	3	0	0	3	E4 / DSE
5	CA681	Robotics Lab	0	0	3	2	CC-L5
6	CA683	Summer Project	0	0	8	4	SP / E
7	CA691	Dissertation Part - I	0	0	16	8	DP1 / E
8	GP	General Proficiency	Non-Credit				
		Total Hours and Credits	**12**	**0**	**27**	**26**	

SEMESTER IV

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CA692	Dissertation Part - II	0	0	52	26	DP2 / E
2	GP	General Proficiency	Non-Credit				
		Total Hours and Credits	**0**	**0**	**52**	**26**	

GRAND TOTAL OF CREDITS = 104

Summer Project will be done individually, and it will be evaluated as per University Examination Rules.

Dissertation will be done individually, and it will be evaluated as per University Examination Rules.

USICT will provide a mentor/supervisor for summer project, and dissertation.

ELECTIVES FROM DCSE

S.No.	Course Code	Course Name	L	T	P	Credits	Types
1	CA528	Natural Language Processing	3	0	0	3	E1
2	CA530	Speech Processing and Systems	3	0	0	3	E1
3	CA532	Pattern Matching	3	0	0	3	E1
4	CA534	Digital Transformation using AI	3	0	0	3	E1
5	CA536	Intelligent Information Retrieval	3	0	0	3	E2
6	CA538	Evolutionary Computation	3	0	0	3	E2
7	CA540	Fuzzy Set Theory	3	0	0	3	E2
8	CA542	Adavanced Computer Vision	3	0	0	3	E2
9	CA625	Deep Learning	3	0	0	3	E3
10	CA627	Predictive Analysis	3	0	0	3	E3
11	CA629	R Programming	3	0	0	3	E3
12	CA631	Robot Kinematics	3	0	0	3	E3
13	CA633	Embedded System and Design	3	0	0	3	E4
14	CA635	AI Enabled Cyber Security	3	0	0	3	E4
15	CA637	Human Machine Interation	3	0	0	3	E4
16	CA639	Robotics Planing Algorithms	3	0	0	3	E4

OPEN AND GENERIC ELECTIVES FROM OTHER SCHOOLS

17	ES415	Energy and Environment	3	0	0	3	OE1
18	EN531	Language, Culture and Society	3	0	0	3	OE2
19	MA402	Modeling and Simulation	3	1	0	4	GE1
20	MA416	Probability and Stochastic Process	3	1	0	4	GE1

CA **C**omputer Science and **A**rtificial Intelligence for Course Code

CS Computer Science for Course Code

CC Core Course from USICT for Type of Course

CC-L Core Course Lab from USICT for Type of Course

GE General Elective from related discipline of other Deptt. /School

AECC Ability Enhancement Compulsory Course

OE Open Elective from other discipline of other Deptt. /School

FC Foundation Course

DSC Discipline Specific Course

SEC Skill Enhancement Course

E Elective

DP1 Dissertation Part 1

DP2 Dissertation Part 2

SP Summer Project

ADVANCED DATABASE MANAGEMENT SYSTEM			
Course Code:	**CS521**	**Course Credits:**	**3**
Course Category:	**CC1**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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 Knowledge of database design
2 A general understanding of database, design and dependency
3 Understanding of different types of databases
4 Knowledge of databases on the internet
5 Application on enhanced database

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of ER diagram and UML class diagram.
2 Ability to apply functionality and Normalization in relational databases.
3 Recognize and fetch data from object oriented, parallel and distributed databases.
4 Use XML and understand unstructured data
5 Implement concept and deduction of enhanced database on different applications

UNIT I INTRODUCTION TO DATABASE DESIGN

Entities, Attributes, Entity Sets, Relationships, Key Constraints, Participation Constraints, Weak Entities, UML Class Diagrams, Subclasses, Superclasses, Inheritance,Specialization, Generalization, Constraints and Characteristics of Specialization and Generalization Hierarchies, Modeling of UNION Types Using Categories, Representing Specialization and Generalization In UML Class Diagrams, Data Abstraction, Knowledge Representation and Ontology Concepts.

UNIT II DATABASES DESIGN THEORY

Problems Caused by Redundancy, Decompositions, Problems Related to Decomposition, Reasoning About FD's, FIRST, SECOND, THIRD Normal Form, BCNF, Fourth Normal Form, Lossless Join Decomposition, Dependency Preserving Decomposition, Schema Refinement in DataBase Design, Multi Valued Dependencies.

UNIT III OBJECT- ORIENTED, PARALLEL AND DISTRIBUTED DATABASES

Overview of Object-Oriented Concepts, Object Identity, Object Structure, Type Constructor, Encapsulation of Operations, Methods and Persistence; Architectures for Parallel Databases, Parallel Query Evaluation, Parallelizing Individual Operations, Sorting Joins, Distributed Database Concepts, Data Fragmentation, Replication and Allocation Techniques for Distributed Database Design, Query Processing in Distributed Databases, Concurrency Control and Recovery in Distributed Databases.

UNIT IV DATABASES ON THE WEB AND SEMI–STRUCTURED DATA

Web interface, XML, structure of XML data, querying XML data, storage of XML data, XML applications, semi-structured data model, indexes for text data.

UNIT V ENHANCED DATA MODELS FOR ADVANCED APPLICATIONS

Active database concepts, temporal database concepts, spatial databases: concept and architecture, deductive databases and query processing, mobile databases, Geographic Information Systems (GIS).

Textbooks:

1. Elmasri and Navathe, Fundamentals of Database Systems,
2. Ramakrishnan and Gehrke, Database Management Systems,

References Books:

3. Korth, Silberschatz, Sudarshan, Database System Concepts,
4. Rob and Coronel, Database Systems: Design, Implementation and Management,
5. Date and Longman, Introduction to Database Systems,

DESIGN AND ARCHITECTURE OF SOFTWARE SYSTEM			
Course Code:	**CS-523**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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 Perceive the creational and structural patterns.
2 Be capable of applying his knowledge to create an architecture for a given application.
3 Be able to explain the role of analysing architectures.
4 Acknowledging software design.
5 Be able to identify different structural patterns.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Recognize the architecture, creating it and moving from one to any, different structural patterns.
2 Ability to analyse the architecture and build the system from the components.
3 Design creational and structural patterns.
4 Learn about behavioural patterns.
5 Do a case study in utilizing architectural structures.

UNIT I SOFTWARE ARCHITECTURE

Foundations of software architecture, goals of software architecture limitations, role of software architect, types of architecture, qualities attributes, qualities scenario, architectural styles, common architectural design, architectural design process, key architecture principles, key design principles, functional and non-functional properties of software architectures, heterogeneous architectures, virtual machine architecture, data flow architecture, service-oriented architecture.

UNIT II DESIGN FUNDAMENTALS AND METHODOLOGIES

Nature of design process: objectives, building modules, constructs, design qualities, assessing the design, design viewpoints for software, design strategies: top down and bottom up, organizational methods and design, Jackson structural programming, Jackson system development, models for software architecture

UNIT III SOFTWARE ARCHITECTURE DESIGN

Architectural design and mapping, architecture design patterns, module architecture view, styles of the module view type, execution architecture view, code architecture view, component-and-connector view type, styles of component-and-connector view type, allocation view type and styles, object-oriented architecture, user interface architecture, quantified design space, formalizing architectural description language, first class connectors, tools for architectural design: Unicon, A4; exploiting style in architectural design, architectural interconnection.

UNIT IV INTERACTION ORIENTED SOFTWARE ARCHITECTURE AND DESIGN

Model-View-Controller (MVC), Presentation-Abstraction-Control (PAC) architecture, distributed architecture: client server architecture, multi-tire, service-oriented architecture (SOA). Design principles, traditional approach to design, Structured Analysis Design Technique (SADT), Structures System Analysis and Design Method (SSADM), user interface design; human factor, human computer interaction, interface design guidelines, standards, object-oriented analysis and design.

UNIT V PATTERNS

Design patterns, creational patterns, access control patterns, service variation patterns, service extension patterns, archetypes patterns, model driven architecture with archetype patterns, literate modelling, Customer Relationship Management (CRM) archetype pattern, product archetype pattern, quantity archetype pattern, rule archetype pattern, layering, organizing domain logic, mapping to relational databases, web presentation, domain logic patterns, data source architectural patterns, object-relational behavioural patterns, object relational structural patterns, object-relational metadata mapping patterns, web presentation patterns, distribution patterns, offline concurrency patterns.

Text Books:

1. Software Architecture Perspectives on an Emerging Discipline, M. Shaw Prentice-Hall, 1996.
2. Software Architecture Design: Methodology and Styles, Lixin Tao, Xiang Fu and Kai Qian, Stipes Publishing L.L.C., 2006.
3. Software Architecture in Practice, Len Bass, Paul Clements, Rick Kazman, Pearson Education Asia,2003.

References Books:

4. Software Design, David Budgen, Addison-Wesley, 1994.
5. Software Engineering, Pressman R.S, McGraw Hill Inc., 1996.

ADVANCED DATA STRUCTURE AND ALGORITHM			
Course Code:	**CS-525**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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. Recognizing the appropriate data structures, ADT libraries, and using it to design algorithms for a specific problem.
2. Be capable of solving problems using abstraction techniques.
3. Be able to choose appropriate algorithms for a specific problem.
4. Be able to analyse algorithms in terms of their efficiency and correctness.
5. To accept the recent developments in the area of algorithm design.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Design and analyse programming problem statements.
2. Choose appropriate data structures and algorithms for a specific problem.
3. Gain the necessary mathematical abstraction to solve problems.
4. Come up with an analysis of efficiency and proof of correctness.
5. Comprehend and select algorithm design approaches in a problem specific manner.

UNIT I INTRODUCTION

Review of Basic Concepts: Abstract data types, Data structures, Algorithms, Big-Oh, Small-Oh, Omega, Small-Omega and Theta Notations, finding time complexity of programs, **Recurrence Relations:** Solving Recurrence Relations, Substitution Method, Master Theorem.

UNIT II

Hashing: Review of Hashing, Hash Function, Collision Resolution Techniques in Hashing, Separate Chaining, Open Addressing, Linear Probing, Quadratic Probing, Double Hashing, Rehashing, Extendible Hashing, Recent Trends in Hashing.

UNIT III TREES & GRAPH

Trees: Binary Search Trees, AVL Trees, Red Black Trees, 2-3 Trees, B-Trees, Splay Trees, Minimum Spanning Tree (MST), Kruskal's Algorithm and Prim's Algorithm, Applications to MST.
Graph: Graph, Breadth First Search, Depth First Search, Shortest path in edge-weighted case (Dijkstra's), Bellman Ford Algorithms, Topological Sorting.

UNIT IV SELECTED TOPICS

Strassen's Matrix Multiplication, Greedy method VS Dynamic Programming, Job sequencing with deadlines, Fractional Knapsack Problem, 0/1 Knapsack Problem, Travelling Salesman Problem, Huffman coding, Pre order, Post order, Inorder traversal, Postfix to infix notation, Infix to Postfix notation.

UNIT V

Linear Programming: Geometry of the feasibility region and Simplex algorithm
NP-completeness: Examples, proof of NP-hardness and NP-completeness.
Recent Trends: Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.

Text Books:

1. Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
2. Algorithms Unlocked: Thomas H. Cormen.
3. The Algorithm Design Manual, Steven S. Skiena.

References Books:

1. Algorithms: Robert Sedgewick and Kevin Wayne.
2. Advanced Data Structures: Peter Brass.

RESEARCH TECHNIQUES IN ICT			
Course Code:	**CS527**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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 know what research is.
2. To make the students aware and apply ICT in Research.
3. To make the students recognize Scientific Research Methodology.
4. To equip the students to examine the philosophical and socio-cultural context of research and relate it to the contemporary paradigm shift.
5. To explore different traditions such as empiricism, rationalism, and constructivism etc.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Discuss issues of research and knowledge creation
2. Apply appropriate research methods to research problems
3. Critique research methodologies
4. Able to write research papers so as to develop a great career in the field of research.
5. Design a research program.

UNIT I INTRODUCTION TO RESEARCH TECHNIQUES

Meaning of research, objectives of research, motivation in research, types of research, characteristics and prerequisites of research, significance of research, research process, sources of research problem, criteria of identifying the problem, necessity of defining the problem, errors in selecting research problem, technique involved in defining the problem, report, and paper writing.

UNIT II DATA ANALYSIS ANDSTATISTICAL TECHNIQUES

Data and their analyses, quantitative methods and techniques, Measure of central tendency, measures of variation, frequency distribution, analysis of variance, methods, Correlation analysis, regression analysis, time series and forecasting, introduction to discriminant analysis, factor analysis, cluster analysis, conjoint analysis, probability distribution, binomial distribution, poisson distribution, uniform distribution, exponential distribution, and normal distribution, sampling methods, test of hypothesis.

UNIT III MATHEMATICAL MODELING

Steps of modelling, operations research models like queuing theory, stochastic processes, application of models, conceptual framework development and validation techniques, optimization techniques.

UNIT IV ALGORITHMIC RESEARCH

Algorithmic research problems, types of algorithmic research, types of solution procedure, steps of algorithm development, steps of algorithmic research, design of experiments.

UNIT V SIMULATION AND SOFT COMPUTING TECHNIQUES

Introduction to soft computing, artificial neural network, genetic algorithm, fuzzy logic and their applications, tools of soft computing, need for simulation, types of simulation, simulation language, fitting the problem to simulation study, simulation models, output analysis, data simulation packages like MATLAB, NS2, ANSYS, Cadence.

Text books:

1. Research Methodology: Methods and Techniques, C.R. Kothari

Reference Books:

1. Research Methodologies, R. Panneerselvam, Prentice Hall, 2007.
2. Research in Education, Best John V. and James V Kahn, Wiley eastern, 2005.
3. Elements of Educational Research, Sukhia, S.P., P.V. Mehrotra, and R.N. Mehrotra, PHI publication, 2003.
4. Methodology of Research Education, K. Setia, IEEE publication, 2004.
5. Research methodology, Methods and Techniques, Kothari, C.R., 2000.

JAVA PROGRAMMING			
Course Code:	**CS529**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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 Knowledge of basic Object-Oriented paradigm, practices and application.
2 A general understanding of class, object and methods.
3 Understanding of multithreading and applets.
4 Basic knowledge of swings and Beans with implementation.
5 Understanding of Servlet programming.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of object-oriented programming.
2 Ability to apply OOPs concept in real life problems.
3 Ability to design, develop, maintain and evaluate large-scale software systems.
4 To produce efficient, reliable, robust and cost-effective software solutions using Java.
5 Ability to perform independent research and analysis.

UNIT I OBJECT-ORIENTED PROGRAMMING

Concept of object-oriented programming (OOP), benefits of OOP, application of OOP, Java history, Java features, Data types, Java key words, identifiers, constants, variables, declaration and scope of the variable, symbolic constant, type casting, arrays, strings, vectors, wrappers classes, operator, expressions, program control statements: decision making and branching: if, if…. else, else…. if, else if ladder, switch, decision making and looping while, do…. while, for.

UNIT II 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, finalizer methods, abstract methods and classes, visibility control, exception handling fundamental, Interface and Packages.

UNIT III MULTITHREADING AND APPLET PROGRAMMING

Multithreading programming: creating threads, thread class and runnable interface extending the thread class, stopping and blocking a thread, life cycle of a thread, thread methods, thread exceptions, thread priority, synchronization, thread communication using notify(), wait(), and notify all(), applet programming : 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 to applets, applets and HTML.

UNIT IV SWING AND BEANS

Introduction to Swing, Differences between AWT Controls & Swing Controls, JApplet, Swing Button: JButton, JToggleButton, CheckBoxes, Radio Button, JComboBox, Text Boxes etc., Icons, Labels, JTabbed Pains, JScroll Pains, JList, JTrees, JTables Java Beans: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Developing a Home page using Applet & Swing.

UNIT V SERVLET PROGRAMMING

Introduction to Servlets: Lifecycle of a Servlet, The Servlet API, The javax. Servlet Package, Reading Servelet parameters, Reading Initialization parameters; The javax.servlet HTTP package, Handling Http Request & Responses, Security Issues Introduction to JSP, Problem with Servlet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC Setting Up and JSP Environment: Installing the Java Software Development Kit, Tomcat Server & Testing Tomcat. U

REFERENCE BOOKS:

1. Programming with JAVA, E. Balagurusawamy, Tata McGraw Hill, 1998.
2. JAVA Beginner's guide, Herbert Schildt, Tata McGraw Hill, 2007.
3. Java How to Program, Deitel & Deitel, Prentice-Hall, 1999.
4. The Complete Reference JAVA 2, Herbert Schildt, 5th Edition, Tata McGraw Hill, 2002.
5. The Complete Reference JAVA 2, Herbert Schildt, 7th Edition, Tata McGraw Hill, 2009.
6. The Java Programming Language, Ken Arnold, James Gosling, Addison-Wesley, 1996.
7. How to Program Java, Peter Coffee, Ziff-Davis Press, 1996.

ADVANCED DATABASE MANAGEMENT SYSTEM LAB			
Course Code:	**CS581**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U / P**
Course Year (U / P):	**4U / 1P**	**Course Semester (U / P):**	**8U / 2P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To explore the features of a Database Management Systems
2. To interface a database with front end tools
3. To comprehend the internals of a database system
4. To provide a strong foundation in advanced database concepts from an industry perspective.
5. To learn query processing and transaction management concepts for object-relational database and distributed database

COURSE OUTCOME
At the end of the course the students should be able to:
1. Develop and apply critical thinking skills.
2. Design and present Lab as well as project reports
3. Built appropriate methods for the analysis of raw data
4. Perform logical troubleshooting as and when required.
5. Verify and implement the concepts and theory learnt in class.

1. Introdution to MySQL, Postgre Sql, Microsoft Sql softwares.
2. An exercise of data types in PostGresql & Data Definition Language Commands
3. Exercise on Data Manipulation Language and Transaction Control Commands using PostgreSql.
4. Exercise on Types of Data Constraints using PostgreSql.
5. Exercise on JOINS (Single-Table) Using Normalization
6. Exercise on JOINS (Multiple-Table) Using Normalization
7. Exercise on GROUP BY/ORDER BY Clause and Date Arithmetic using PostgreSql.
8. Exercise on different Functions (Aggregate, Math and String)
9. Exercise on different types of sub queries
10. Procedures, View and Triggers

JAVA PROGRAMMING LAB			
Course Code:	**CS583**	**Course Credits:**	**2**
Course Category:	**CC-L2**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
No. of Labs (Hrs./Week):	**01 (3 Hr)**		
Total No. of Labs:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1 Knowledge of basic Object-Oriented paradigm, practices, and application.
2 A general understanding of class, object, and methods.
3 Understanding of multithreading and applet.
4 Basic knowledge of Swings and Beans with implementation.
5 Understanding of Servlet programming.

COURSE OUTCOMES
At the end of the course the students should be able to:
1 Basic knowledge and understanding of object-oriented programming.
2 Ability to apply OOPs concept in real life problem.
3 Ability to design, develop, maintain, and evaluate large-scale software systems.
4 To produce efficient, reliable, robust, and cost-effective software solutions using Java.
5 Ability to perform independent research and analysis.

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, Operators, Decision statement, Loops statement and Branch statements and Exception handling

3. Write a separate Java Code to implement each of the following OOP's concepts:
   Abstraction, Encapsulation, Inheritance, Polymorphism, Method Overloading and Method Overriding

4. Write a separate Java Code to implement each of the following:
   Class, Object, Constructors, Method, and Visibility Controls: Private, Public and Protected

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

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

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

8. Write a separate Java Code to implement each of the following:
   Swing and Beans

9. Write a separate Java Code to implement each of the following:
   Swing Button: JButton, JToggleButton, CheckBoxes, Radio Button, JComboBox, Text Boxes

10. Write a separate Java Code to implement each of the following:
    Servlet and JSP

ENERGY AND ENVIRONMENT			
Course Code:	**ES 415**	**Course Credits:**	**(L–T–P): 3 (3-0-0)**
Course Category:	**OE1**	**Course (C/P):**	**P**
Course Year (U/P):	**1P**	**Course Year (U/P):**	**1P**
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 OBJECTIVE	
1.	To provide in-depth knowledge of renewable and non-renewable energy resources
2.	To provide knowledge about energy harnessing techniques
3.	To study the energy-environment issues

COURSE OUTCOME	
At the end of the course the students can use the knowledge so gathered and utilized to	
1.	Meet the challenges of energy *vis-a-vis* environmental security

UNIT I SUN AS SOURCE OF ENERGY

Nature of its radiation, solar radiation and its spectral characteristics; Conventional energy sources (coal, oil, biomass and natural gas), non-conventional energy sources (hydro-electric power, tidal, wind, geothermal, solar, nuclear magneto-hydrodynamic power MHD); Energy use pattern in India and parts of world, Energy security.

UNIT II FOSSIL FUELS

Classification, composition, physico-chemical characteristics; Calorific value – gross and net; Energy content of coal, petroleum and natural gas, shale oil, coal bed methane, gas hydrates Concept of Green Energy; Principles of generation of hydropower, tidal energy, ocean thermal energy conversion, wind power, geothermal energy, solar energy (solar collectors, photo-voltaic modules, solar ponds).

UNIT III NUCLEAR ENERGY

Fission and fusion, nuclear fuels, nuclear reactor – principles and types; Mechanism of radiation action on living organisms - Stochastic and Non-stochastic effects, delayed effects; Radioactivity from nuclear reactors, fuel processing and radioactive waste, hazards related to power plants

UNIT IV BIOENERGY

Types, importance, methods of energy production from biomass

UNIT V ENVIRONMENTAL IMPLICATIONS OF ENERGY USE

CO_2 emission and atmosphere –scenario in developed and developing world (and India), Global warming, Radiative forcing, Impacts of large-scale exploitation of solar, wind, hydro, nuclear and bioenergy sources, National Solar Mission, National Mission for Enhanced Energy Efficiency, case studies.

SUGGESTED READINGS

Fay, J.A. and Golomb, D.S. 2011. *Energy and the Environment*, Oxford University Press, New Delhi.
Iqbal, M. 1983. *An Introduction to Solar Radiation*. Academic Press, New York.
Kaushika, N.D. and Kaushik, K. 2004. *Energy, Ecology and Environment: A Technological Approach*, Capital Publications, New Delhi. ***Website*** *- https://nptel.ac.in/course.html*

Sem- II

Soft Computing Techniques			
Course Code:	**CA522**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures (Hrs/Week) / Total No. of Lectures	**03 / 45**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To expose the concepts of feed forward neural networks.
2. To provide adequate knowledge about feedback neural networks.
3. To teach about the concept of fuzziness involved in various systems.
4. To expose the ideas about genetic algorithm
5. To provide adequate knowledge about of FLC and NN toolbox

COURSE OUTCOME
At the end of the course the students should be able to:
1. Learn and understand Fuzzy logic and its applications.
2. Comprehend Artificial neural networks and its applications.
3. Solving single-objective optimization problems using GAs.
4. Solving multi-objective optimization problems using Evolutionary algorithms
5. Applications of Soft computing to solve problems in varieties of application domains.

UNIT I INTRODUCTION

Introduction to Soft Computing; Definition, requirement, necessity, and adequacy; various dialects of soft computing – Evolutionary Algorithms, Fuzzy Sets and Fuzzy Logic, Artificial Neural Networks - their suitability in Searching, optimization, decision matching and pattern related problems; potential areas of applications.

UNIT II FUZZY SETS AND FUZZY LOGIC

Introduction to fuzzy sets and fuzzy logic; difference between classical and fuzzy sets; chance vs fuzziness; limitations of fuzzy systems; typical shapes of membership functions and their usage; operations on fuzzy sets: compliment, intersection, union; combinations on operations, aggregation operation.

UNIT III FUZZY RELATIONS AND FUZZY SYSTEMS

Cartesian Product; Classical Relations and Fuzzy Relations; Cardinality, operations, and properties of crisp and fuzzy relations; Composition of operations, Fuzzy cartesian product; The linguistic variables, Reasoning in fuzzy logic, Fuzzification and defuzzification; Mamdani and Sugano Fuzzy Inference Systems.

UNIT IV NEURAL NETWORK

Overview of biological neurons; McCulloch-Pitts model, Rosenblatt's Perceptron model, difference, capabilities, and limitations; Model of generic computational neuron; Basic activation functions; Basic Learning laws of neurons; Single layer and multilayer architectures; Feedforward and feedback networks.

UNIT V LEARNING FUNDAMENTALS

Learning paradigms, supervised and unsupervised learning, reinforced learning; back propagation algorithm; Radial basis neurons, Generalized Regression Neural network, Probabilistic Neural Networks; Competitive learning; Self Organizing Features Map, Hopfield networks, associative memories, applications of artificial neural networks. Elasticity vs plasticity dilemma, preprocessing, post processing, early stopping.

UNIT VI EVOLUTIONARY ALGORITHMS

Problems suitable and not suitable for applying evolutionary algorithms; Various dialects of evolutionary Algorithms; Terminology of Genetic Algorithms; Canonical Genetic Algorithm; Common representations and related reproduction operators; premature convergence, schema theorem, minimal deceptive problem and Royal Road function; fitness function, Roulette wheel selection, Rank selection, Tournament Selection; termination criteria, survivor selection, population models; parallel implementations.

Textbooks:

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. An Introduction to Genetic Algorithms, Milanie Mitchell, MIT Press 1998.
4. A Genetic Algorithm Tutorial, Darrell Whitley.

Reference Books:

1. Fuzzy Sets, Fuzzy logic, and Fuzzy Systems: Selected Papers- Lotfi Asker Zadeh, George J. Kilr, Bo yuan, 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. Neural Networks Theory and Application, Oscar Castillo, Wiley Eastern publication
5. Genetic Algorithms in Search, Optimization and Machine Learning, David E Goldberg, Eddison-Wesley, 1988.

Knowledge Engineering			
Course Code:	**CA524**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures (Hrs/Week) / Total No. of Lectures	**03 / 45**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To learn knowledge management in different organizations
2. To discover about knowledge economy and knowledge management ontology
3. To get introduced to the basic knowledge representation, problem solving, and learning methods of Artificial Intelligence
4. to be able to implement the expertise model as a prototype
5. To familiarize with Fuzzy Logic and knowledge processing in expert systems

COURSE OUTCOME
At the end of the course the students should be able to:
1. know the fundamental concepts of Artificial Intelligence such as knowledge representation, problem solving, fuzzy set and expert systems
2. Develop knowledge on management systems in various organizations.
3. Ability to implement production systems, frames, inheritance systems and approaches to handle uncertain or incomplete knowledge.
4. Handle Knowledge Transfer and Knowledge Sharing
5. Test and deploy knowledge in the organizations

UNIT I

Knowledge, Representation, Reasoning, why knowledge representation and reasoning, Role of logic, Historical background, Representing knowledge in logic, Varieties of logic, Name, Type, Measures, Unity Amidst diversity

UNIT II

Ontology: Ontological categories, Philosophical background, Top-level categories, describing physical entities, Defining abstractions, Sets, Collections, Types and Categories, Space and Time

UNIT III

Knowledge Representations: Knowledge Engineering, Representing structure in frames, Rules and data, Object-oriented systems, Natural language Semantics, Levels of representation

UNIT IV

Processes: Times, Events and Situations, Classification of processes, Procedures, Processes and Histories, Concurrent processes, Computation, Constraint satisfaction, Change Contexts: Syntax of contexts, Semantics of contexts, First-order reasoning in contexts, Modal reasoning in contexts, Encapsulating objects in contexts.

UNIT V

Knowledge Soup: Vagueness, Uncertainty, Randomness and Ignorance, Limitations of logic, Fuzzy logic, Nonmonotonic Logic, Theories, Models and the world, Semiotics Knowledge Acquisition and Sharing: Sharing Ontologies, Conceptual schema, accommodating multiple paradigms, Relating different knowledge representations, Language patterns, Tools for knowledge acquisition

Textbooks:

1. Knowledge Representation logical, Philosophical, and Computational Foundations by John F. Sowa, Thomson Learning.
2. Knowledge Representation and Reasoning by Ronald J. Brachman, Hector J. Levesque, Elsevier.

Machine Translation and Learning			
Course Code:	**CA526**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lab (Hrs./Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To enable students to grasp the history of machine translation from a computational linguistic point of view as well as from the translation perspective
2. Recognize the different approaches adopted for machine translation show how research efforts have been spent to solve translation problems using the computer
3. how the technology has evolved due to the deepening understanding of computational linguistics and translation studies, and the advent of new tools
4. To design and implement a function based on high-level description such as pseudocode or a precise mathematical statement of what the function computes
5. to describe various types of applications of machine translations.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. explain standard models and the foundations of machine translation strategies (such as rule-based machine translation and statistical machine translation) as well as to describe differences between these strategies;
2. analyse and critically review scientific publications in the field of machine translation
3. explain, apply, and assess manual and automatic evaluation methods for machine translation;
4. build translation models based on existing tools for statistical machine translation and one's own training data as well as evaluate and analyse the results of translations produced by those models
5. formulate and critically discuss the methodological assumptions made by the techniques discussed in the course and to present the results in a professionally adequate way.

UNIT I OVERVIEW OF MACHINE TRANSLATION

Definition, fundamental concepts, applications, language similarities and differences, cross language information retrieval, computer aided human translation, sublanguage classification, concept of polysynthetic language, Sapir – Whorf hypothesis, concept of ontology, direct vs. indirect translation, statistical model of machine translation.

UNIT II DIFFERENT MACHINE TRANSLATION MODELS

Graphical models: belief networks, Bayesian networks, Hidden Markov models, incremental learning, reinforcement learning, machine learning applications.

UNIT III MACHINE TRANSLATION METHODOLOGIES

Decision trees, linear discrimination, Neural networks, support vector machines (SVMs), quantifying fluency and faithfulness, usability and system development, direct transfer, quantifying fluency, and faithfulness, boosting and bagging, naive Bayes classifiers, gradient-descent, Qlearning.

UNIT IV MACHINE-LEARNING FUNDAMENTALS

Classification, regression, and clustering; noisy, noise-free and incomplete data; supervised and unsupervised learning; hypothesis classes, model complexity, model selection, Ockham's razor and bias-variance dilemma, dynamic environments, reinforcement learning and the exploration exploitation dilemma.

UNIT V BASIC LEARNING METHODS

Unsupervised learning: K-means, vector quantization, self-organizing neural networks. Supervised learning: K nearest neighbor, learning vector quantization, decision tree, supervised neural networks, the transfer metaphor, syntactic transformations, lexical transformations.

Textbooks:

1. Machine Translation of Languages, Wilen Sky R. Planning and understanding, Addison Wisely, Reading MA, 2003.
2. Artificial Neural Networks: An Introduction to ANN Theory and Practice by Peteus J. Braspenning, PHI publication, 2005.

References Books:

1. A New Approach to Machine Translation, Russel S, and Norvig P, Pearson education publication, 2003.
2. Evolutionary Language Understanding, Sampson G. Mc. Graw Hill publication, 2002.
3. The Machine Translation perspective, Schank, R.C., PHI publication 2000.

Speech Processing Systems			
Course Code:	**CA530**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Lectures (Hrs/Week) / Total No. of Lectures	**03 / 45**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. To explain how DSP techniques could be used for solving problems in speech communication
2. To recognize the speech production mechanism and the various speech analysis techniques and speech models
3. To know the speech compression techniques
4. To appreciate the speech recognition techniques
5. To know the speaker recognition and text to speech synthesis techniques

COURSE OUTCOME
At the end of the course the students should be able to:
1. Identify the various temporal, spectral and cepstral features required for identifying speech units – phoneme, syllable, and word
2. Determine and apply Mel-frequency cepstral coefficients for processing all types of signals
3. Justify the use of formant and concatenative approaches to speech synthesis
4. Identify the apt approach of speech synthesis depending on the language to be processed
5. Determine the various encoding techniques for representing speech.

UNIT I SPEECH SIGNAL CHARACTERISTICS & ANALYSIS

Speech production process - speech sounds and features- - Phonetic Representation of Speech -- representing= speech in time and frequency domains - Short-Time Analysis of Speech - Short- Time Energy and Zero-Crossing Rate - Short-Time Autocorrelation Function - Short-Time Fourier Transform (STFT) - Speech Spectrum - Cepstrum - Mel-Frequency Cepstrum Coefficients - Hearing and Auditory Perception - Perception of Loudness - Critical Bands - Pitch Perception

UNIT II SPEECH COMPRESSION

Sampling and Quantization of Speech (PCM) - Adaptive differential PCM - Delta Modulation - Vector Quantization- Linear predictive coding (LPC) - Code excited Linear predictive Coding (CELP)

UNIT III SPEECH RECOGNITION

LPC for speech recognition- Hidden Markov Model (HMM)- training procedure for HMM- subword unit model based on HMM- language models for large vocabulary speech recognition - Overall recognition system based on subword units - Context dependent subword units- Semantic post processor for speech recognition

UNIT IV SPEAKER RECOGNITION

Acoustic parameters for speaker verification- Feature space for speaker recognition-similarity measures- Text dependent speaker verification-Text independent speaker verification techniques

UNIT V SPEAKER RECOGNITION AND TEXT TO SPEECH SYNTHESIS

Text to speech synthesis (TTS)-Concatenative and waveform synthesis methods, sub-word units for TTS, intelligibility, and naturalness-role of prosody

TEXTBOOKS:

1. L. R. Rabiner and R. W. Schafer, Introduction to Digital Signal Processing, Foundations and Trendsin Signal Processing Vol. 1, Nos. 1–2 (2007) 1–194
2. Ben Gold and Nelson Morgan ―Speech and Audio signal processing- processing and perception of speech and music‖, John Wiley and sons 2006

REFERENCES

1. Lawrence Rabiner, Biiing and– Hwang Juang and B.Yegnanarayana ―Fundamentals of Speech Recognition‖, Pearson Education, 2009
2. Claudio Becchetti and Lucio Prina Ricotti, ―Speech Recognition‖, John Wiley and Sons, 1999
3. Donglos O shanhnessy ―Speech Communication: Human and Machine ―, 2nd Ed. University press 2001.

Evolutionary Computation			
Course Code:	**CA538**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**P**
Course Year	**2P**	**Course Semester (U / P):**	**3P**
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 Introduce the main concepts, techniques, and applications in the field of evolutionary computation.
2. Give students some practical experience on when evolutionary computation techniques are useful, how to use them in practice and how to implement them with different programming languages.
3. Develop an understanding of issues in computing using evolutionary algorithms, in particular genetic algorithms.
4. Learn how the knowledge of evolutionary computing can be used in application areas.
5. To learn multi objective optimization

COURSE OUTCOMES
On completion of this course, the student should be able to:
1. Understand the implementation issues of evolutionary algorithms.
2. Determine the appropriate parameter settings to make different evolutionary algorithms work well.
3. Design new evolutionary operators, representations, and fitness functions for specific practical and scientific applications.
4. Effectively present an evolutionary computation article to an audience.
5. Review and critique evolutionary computation articles.

UNIT-I–FUNDAMENTAL CONCEPTS AND DIALECTS

METAHEURISTICS, REQUIREMENTS OF METAHEURISTICS IN OPTIMIZATION PROBLEMS, CHARACTERISTICS OF PROBLEMS SUITABLE FOR APPLICABILITY OF METAHEURISTICS; EVOLUTIONARY COMPUTING METAPHOR AND INSPIRATION FROM BIOLOGY; APPLICATIONS, PROS, AND CONS OF EVOLUTIONARY COMPUTATIONS; COMPONENTS OF EVOLUTIONARY COMPUTATION, EXAMPLE APPLICATIONS – EIGHT QUEEN PROBLEM, KNAPSACK PROBLEM; VARIOUS DIALECTS OF EVOLUTIONARY COMPUTATION.

UNIT-II–GENETIC ALGORITHMS

INTRODUCTION; CANONICAL GA; BINARY, INTEGER, REAL-VALUED, AND PERMUTATION REPRESENTATIONS AND VARIATION OPERATORS FOR THEM; POPULATION MODELS; PARENT SELECTION – FITNESS PROPORTIONATE SELECTION, RANK BASED SELECTION, IMPLEMENTING SELECTION PROBABILITIES, TOURNAMENT SELECTION; SURVIVOR STRATEGY. IMPLEMENTATION ISSUES, PARAMETERS' CONTROL AND EFFECT ON GA DYNAMICS.

UNIT-III– THEORETICAL FOUNDATION OF GENETIC ALGORITHMS

SCHEMAS AND HYPERPLANE SAMPLING; SCHEMATA THEOREM, LIMITATIONS AND BUILDING BLOCK HYPOTHESIS; TWO-ARMED BANDIT PROBLEM; DECEIVING A GA; MINIMAL DECEPTIVE PROBLEM; ROYAL ROADS FUNCTIONS; SAHC; NAHC; RMHC; HITCHHIKING; EXACT MATHEMATICAL MODELS OF SGA; STATISTICAL-MECHANICS APPROACHES.

UNIT-IV– PROBLEM SOLVING USING EC

EVOLVING COMPUTER PROGRAMS – EVOLVING LISP PROGRAMS, EVOLVING CELLULAR AUTOMATA; DATA ANALYSIS AND PREDICTION – PREDICTING DYNAMICAL SYSTEM, PREDICTING PROTEIN STRUCTURE; EVOLVING NEURAL NETWORKS – EVOLVING WEIGHTS, ARCHITECTURE (DIRECT ENCODING AND GRAMMATICAL ENCODING), AND LEARNING RULES; BALDWIN EFFECT AND EVOLUTIONARY REINFORCED LEARNING.

UNIT-V– PARALLEL IMPLEMENTATION AND OTHER DIALECTS OF EC

PARALLEL IMPLEMENTATION OF GA; GENETIC PROGRAMMING; EVOLUTIONARY PROGRAMMING; EVOLUTIONARY STRATEGIES; LEARNING CLASSIFIER SYSTEMS; MEMETIC ALGORITHMS; INTRODUCTION TO MOGA. INTRODUCTION TO SIMULATED ANNEALING, ARTIFICIAL BEE COLONY ALGORITHMS, ANT COLONY OPTIMIZATION ALGORITHMS, AND SWARM INTELLIGENCE. EVOLUTIONARY COMPUTATION

REFERENCE BOOKS:

1. AN INTRODUCTION TO GENETIC ALGORITHMS, MILANIE MITCHELL, MIT PRESS.
2. GENETIC ALGORITHMS, DAVID E. GOLDBERG, PEARSON EDUCATION.
3. HANDBOOK OF GENETIC ALGORITHMS, LOWRENCE DAVIS, VAN NOSTRAND REINHOLD.
4. MULTI-OBJECTIVE OPTIMIZATION USING EVOLUTIONARY ALGORITHMS, KALYANMOY DEB, WILEY
5. ANT COLONY OPTIMIZATION, MARCO DORIGO AND THOMAS STUTZLE, MIT PRESS, LONDON.
6. SWARM INTELLIGENCE: INTRODUCTION AND APPLICATIONS, C. BLUM AND D. MERKLE; SPRINGER

SOFT COMPUTING TECHNIQUES LAB			
Course Code:	**CA582**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
No. of Labs (Hrs/Week):	**03**	**Mid Sem. Exam Hours:**	
Total No. of Lab:	**10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Provide an understanding of the basic mathematical elements of fuzzy sets.
2. To use fuzzy set operations to implement current computing techniques used in fuzzy computing
3. To implement applications using the fuzzy set operations.
4. Cover fuzzy logic inference with emphasis on their use in the design of intelligent or humanistic systems.
5. To become familiar with neural networks learning algorithms from available examples.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. To analyze the applications using fuzzy set which uses current techniques, skills, and tools necessary for computing.
2. To identity and solves the engineering problems using the fuzzy set theory and identify the differences and similarities between fuzzy sets and classical sets theories
3. Learn mathematical basis as well as the general principles of various soft computing techniques
4. Become familiar with fuzzy relations and the properties of these relations
5. Identify, analyze, design, and solve the problem, implement, and validate the solution including both hardware and software.

List of Experiments: (Any 10 of the following)

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 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 filter using ANN.
9. Implementing SGA to solve function optimization problem 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.

PYTHON PROGRAMMING			
Course Code:	**CS522**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**2P**
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 learn and understand Python programming basics and paradigm
2. To learn and understand python looping, control statements and string manipulations.
3. Students should be made familiar with the concepts of GUI controls and designing GUI applications.
4. To learn and know the concepts of file handling, exception handling and database connectivity.
5. To learn and understand database connectivity in python programming language.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. To read and write simple Python programs.
2. To develop Python programs with conditionals and loops.
3. To define Python functions and to use Python data structures –- lists, tuples, dictionaries
4. To do input/output with files in Python
5. To do searching, sorting and merging in Python

UNIT I INTRODUCTION

The Programming Cycle for Python, Python IDE, Interacting with Python Programs, Elements of Python, Type Conversion. Basics: Expressions, Assignment Statement, Arithmetic Operators, Operator Precedence, Boolean Expression.

UNIT II CONDITIONALS

Conditional statement in Python (if-else statement, its working and execution), Nested-if statement and Elif statement in Python, Expression Evaluation & Float Representation. Loops: Purpose and working of loops, while loop including its working, For Loop, Nested Loops, Break and Continue.

UNIT III FUNCTION

Parts of A Function, Execution of A Function, Keyword and Default Arguments, Scope Rules. Strings: Length of the string and perform Concatenation and Repeat operations in it. Indexing and Slicing of Strings. Python Data Structure: Tuples, Unpacking Sequences, Lists, Mutable Sequences, List Comprehension, Sets, Dictionaries Higher Order Functions: Treat functions as first-class Objects, Lambda Expressions

UNIT IV FILE I/O

File input and output operations in Python Programming Exceptions and Assertions Modules: Introduction, Importing Modules, Abstract Data Types: Abstract data types and ADT interface in Python Programming. Classes: Class definition and other operations in the classes, Special Methods (such as _init_, _str_, comparison methods and Arithmetic methods etc.), Class Example, Inheritance, Inheritance and OOP.

UNIT V ITERATORS & RECURSION:

Recursive Fibonacci, Tower of Hanoi Search: Simple Search and Estimating Search Time, Binary Search and Estimating Binary Search Time Sorting & Merging: Selection Sort, Merge List, Merge Sort, Higher Order Sort

Textbooks:

1. Allen B. Downey, ``Think Python: How to Think Like a Computer Scientist ‘‘, 2nd edition, Updated for Python 3, Shroff/O ‘Reilly Publishers, 2016 (http://greenteapress.com/wp/thinkpython/)
2. Guido van Rossum and Fred L. Drake Jr, —An Introduction to Python – Revised and updated for Python 3.2, Network Theory Ltd., 2011.
3. John V Guttag, —Introduction to Computation and Programming Using Python ‘‘, Revised and expanded Edition, MIT Press, 2013

PYTHON PROGRAMMING LAB			
Course Code:	**CS582**	**Course Credits:**	**2**
Course Category:	**CC**	**Course (U / P)**	**U / P**
Course Year (U / P):	**4U / 1P**	**Course Semester (U / P):**	**8U / 2P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
1. Interpret the use of procedural statements like assignments, conditional statements, loops and function calls.
2. Infer the supported data structures like lists, dictionaries and tuples in Python
3. Illustrate the application of matrices and regular expressions in building the Python programs.
4. Discover the use of external modules in creating excel files and navigating the file systems.
5. Describe the need for Object-oriented programming concepts in Python.

COURSE OUTCOME
At the end of the course the students should be able to:
1. To write, test, and debug simple Python programs.
2. To implement Python programs with conditionals and loops.
3. Use functions for structuring Python programs.
4. Represent compound data using Python lists, tuples, dictionaries
5. Read and write data from/to files in Python.

1. Write a python program find the maximum of a list of numbers.
2. Write a python program to perform Matrix Multiplication.
3. Write a python program first n prime numbers
4. Write a python program selection sort.
5. write a python program to compute the GCD of two numbers.
6. Write a python program to find the most frequent words in a text file.
7. Write a Python program to create a scientific calculator
8. Write a Python program to print all the Disarium numbers between 1 and 100.
9. Write a Python program to encrypt the text using Caesar Cipher technique. Display the encrypted text. Prompt the user for input and the shift pattern.
10. Write a Python program to construct a linked list. Prompt the user for input. Remove any duplicate numbers from the linked list.
11. Perform the following file operations using Python

Sem-III

EXPERT SYSTEM DESIGN			
Course Code:	**CA621**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**T**
Course Year (U / P):	**1P**	**Course Semester (U / P):**	**1P**
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 Knowledge of basic AI and expert system
2 A general understanding of inference, trees etc.
3 Understanding of how to develop expert system
4 Understanding of different types of models used in expert system
5 It is an application of AI that provide system the ability to automatically learn and improve from experience

COURSE OUTCOMES
At the end of the course the students should be able to:
1 fundamental understanding of artificial intelligence (AI) and expert systems.
2 Apply basic principles of inference, perception, knowledge representation, and learning
3 Ability to design, development, stage of expert system.
4 Demonstrate proficiency in applying scientific method to models of machine learning.
5 Describe the use of statistical and reinforcement learning.

UNIT I EXPERT SYSTEM AND KNOWLEDGE REPRESENTATION

Introduction, Advantages, General Concepts, Characteristics, Development of Expert System Technology, Applications and Domain, Language, Shells, Tools, Elements, Production Systems, Procedural Paradigms, Artificial Neural Systems, Meaning of Knowledge, Productions, Semantic Nets, Object-Attribute-Value Triples, Prolog and Semantic Nets, Difficulties With Semantic Nets, Schemata, Frames, Difficulties With Frames, Logic and Sets, Propositional Logic, First Order Predicate Logic, Universal Quantifier, Existential Quantifier and Limitations of Predicate Logic

UNIT II INFERENCE

Introduction, Trees, Lattice, Graphs, State and Problem Spaces, And-Or Trees and Goals, Rule of Inference, Limitations of Propositional Logic, Logic Systems, Resolution, Resolution System and Deduction, Shallow and Casual Knowledge, Forward and Backward Knowledge, Other Method of Inference, Metaknowledge, Hidden Markov Models.

UNITIII DESIGN OF EXPERT SYSTEM

Introduction, Selection of Appropriate Problem, Stages in the Development of Expert System, Errors in Development Stages, Software Engineering and Expert Systems, The Expert System Life Cycle, A Detailed Life Cycle Model.

UNIT IV LEARNING AND CLASSIFIER

Linear Regression (with one variable and multiple variables), Decision Trees and issue in decision tree, Clustering (K-means, Hierarchical, etc.), Dimensionality reduction, Principal Component Analysis, Anomaly detection, Feasibility of learning, Reinforcement learning

UNIT V LEARNING AND COMMUNICATION

Statistical Learning Methods: Introduction to neural networks, Perceptron's, Multi-layer feed forward network, Application of ANN. Reinforcement Learning: Passive reinforcement learning - Active reinforcement learning - Generalization in reinforcement learning

Reference Books:

1. Measuring and Managing Knowledge for Expert System, Mc. Graw- hill Boston, 2001.
2. Dendral: Expert System, Feigenbaum et al, by PHI publication, 1992.
3. Modal Operators in expert systems, Berners Lee, Mc Garw hill publication, 2002.
4. The Frame Based Knowledge Representation in Expert Systems, Mc Carthy and Hays, PHI publication, 2003.
5. Decision Theoretic Expert Systems, Russel, Wiley Eastern publication, 2002.

ROBOTICS			
Course Code:	**CA623**	**Course Credits:**	**3**
Course Category:	**CC**	**Course (U / P)**	**P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
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. Knowledge of Robotics engineering, manipulator and different sensors
2. Appreciate the concepts of linear control
3. Realize the concepts of different kind of sensor
4. Study of different driver and actuator of robotics science
5. Understanding the various method of system stability

COURSE OUTCOMES
At the end of the course the students should be able to:
1. Know the history, Evolution and Anatomy of Robot
2. Realize the concepts of system modeling and control strategy
3. Apply and understand the concepts of manipulator
4. To produce efficient, reliable, robust, and cost-effective Robotic strategy
5. Ability to perform independent research and analysis.

UNIT I- Introduction of Robotics

History of robots, Classification of robots, Present status and future trends. Basic components of robotic system. Basic terminology- Accuracy, Repeatability, Resolution, Degree of freedom. Mechanisms and transmission, Design and control issues, Manipulation and control, Sensors and vision, Programming robot, Future aspect.

UNIT II- Drive systems and Sensors

Drive system- hydraulic, pneumatic and electric systems Sensors in robot – Touch sensors, Tactile sensor, Proximity and range sensors, Robotic vision sensor, Force sensor, Light sensors, Pressure sensors.

UNIT III- Kinematics and Dynamics of Robots

2D, 3D Transformation, Scaling, Rotation, Translation, Homogeneous coordinates, multiple transformation, Simple problems. Matrix representation, Forward and Reverse Kinematics of Three Degree of Freedom, Homogeneous Transformations, Inverse kinematics of Robot, Robot Arm dynamics, D-H representation of robots, Basics of Trajectory Planning.

UNIT IV-Robot Control, Programming and Applications

Robot controls-Point to point control, Continuous path control, Intelligent robot, Control system for robot joint, Control actions, Feedback devices, Encoder, Resolver, LVDT, Motion Interpolations, Adaptive control. Introduction to Robotic Programming, On-line and off-line programming, programming examples. Robot

applications-Material handling, Machine loading and unloading, assembly, Inspection, Welding, Spray painting.

UNIT V- Artificial Intelligence in Robotics

History, state of the art, Need for AI in Robotics. Thinking and acting humanly, intelligent agents, structure of agents. Solving problems by searching, informed search and exploration, constraint satisfaction problems, knowledge and reasoning, knowledge representation, first order logic.

Text Books:

1. Mikell P Groover, Nicholas G Odrey, Mitchel Weiss, Roger N Nagel, Ashish Dutta, “Industrial Robotics, Technology programming and Applications", McGraw Hill, 2012.
2. Craig. J. J. “Introduction to Robotics- mechanics and control”, Addison- Wesley, 1999.

Reference Books:

1. S.R. Deb, “Robotics Technology and flexible automation”, Tata McGraw-Hill Education., 2009.
2. Richard D. Klafter, Thomas. A, ChriElewski, Michael Negin, "Robotics Engineering an Integrated Approach", PHI Learning., 2009.
3. Francis N. Nagy, Andras Siegler, "Engineering foundation of Robotics", Prentice Hall Inc., 1987.
4. P.A. Janaki Raman, "Robotics and Image Processing an Introduction", Tata McGraw Hill Publishing company Ltd., 1995.
5. Carl D. Crane and Joseph Duffy, "Kinematic Analysis of Robot manipulators", Cambridge University press, 2008.
6. Fu. K. S., Gonzalez. R. C. & Lee C.S.G., “Robotics control, sensing, vision and intelligence”, McGraw Hill Book co, 1987
7. Ray Asfahl. C., “Robots and Manufacturing Automation”, John Wiley & Sons Inc.,1985

R Programming			
Course Code:	**CA629**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
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 Know about the Nature of Data
2. To Extract the useful information from Data
3. How to represent data into meaningful information.
4. To learn some basics of statistics
5. How to handle tolerance in the Data.

COURSE OUTCOMES
At the end of the course the students should be able to:
1. The course would enable the ability to understand and critically assess available data using machine learning methods
2. Learn the basic concepts and techniques of Data Science and discover trends in both structured and unstructured data
3. Know the concepts of supervised and unsupervised Learning.
4. Analyse complex problems using advanced analytics tools.
5. Identify the use of large volume data by extracting useful information and patterns and provide predictive insights.

UNIT I: Introduction to R and RStudio

Background, Getting Started, History of R and S, installing R and RStudio, Basic data types in R, Functions for reading and writing data. Using R for calculations. Using R to calculate summary statistics on data. Using R to generate random numbers. Variable types in R. Numeric variables, strings and factors, Statistics with R Analysing Data: Summary Statistics, Correlation and Covariance, Principal Components Analysis, Factor Analysis, Bootstrap Resampling. Probability Distributions: Normal Distribution, Common Distribution-Type Arguments, Distribution Function Families. Statistical Tests for Continuous and Discrete Data, Power Tests: Experimental Design Example, t-Test Design, Proportion Test Design, ANOVA Test Design.

UNIT II: Data structures: vectors, matrices, lists and data frames

Programming in R, Control structures- if-else, for loops, while, break, Repeat, next, Functions, Symbol binding, Scoping rules, Dates and times, the core data structures vectors, matrices, arrays, lists and data frames. subsetting vectors, slicing arrays and drilling down on lists. lapply functions,

UNIT III: Reading data into R from various data sources

Loop functions- lapply, apply, mapply, tapply, split, Basic tools, Using the tools, reading from flat files (plain text), reading from database connections, and reading from web sources, join command

UNIT IV: Statistical modelling functions

Simulation and profiling, Simulation- Generating Random numbers, Simulating a Linear model, Random sampling, R profiler, Line

UNIT V: Writing your own functions

R function syntax, passing of variables into the function, and argument handling, brute force approaches, function evaluations, notation that allows arguments to be passed on to other functions, functions that themselves take other functions as arguments.

Textbooks

1. Hands-On Programming with R: Write Your Own Functions and Simulations 1st Edition, Kindle Edition

Reference Books

1. R for Everyone: Advanced Analytics and Graphics

Embedded System and Design			
Course Code:	**CA633**	**Course Credits:**	**3**
Course Category:	**E**	**Course (U / P)**	**U / P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES
6. Building Blocks of Embedded System
7. Various Embedded Development Strategies
8. Bus Communication in processors, Input/output interfacing.
9. Various processor scheduling algorithms.
10. Basics of Real time operating system and example tutorials to discuss on one real time operating system too.

COURSE OUTCOME
At the end of the course the students should be able to:
1. understand and analyse Embedded systems.
2. study about the bus Communication in processors.
3. operate various Embedded Development Strategies
4. acquire knowledge on various processor scheduling algorithms.
5. suggest an embedded system for a given application

UNIT I INTRODUCTION TO EMBEDDED SYSTEMS

Introduction to Embedded Systems –Structural units in Embedded processor, selection of processor & memory devices- DMA – Memory management methods- Timer and Counting devices, Watchdog Timer, Real Time Clock, In circuit emulator, Target Hardware Debugging**.**

UNIT II EMBEDDED NETWORKING

Embedded Networking: Introduction, I/O Device Ports & Buses– Serial Bus communication protocols RS232 standard – RS422 – RS 485 - CAN Bus -Serial Peripheral Interface (SPI) – Inter Integrated Circuits (I2C) –need for device drivers.

UNIT III EMBEDDED FIRMWARE DEVELOPMENT ENVIRONMENT

Embedded Product Development Life Cycle- objectives, different phases of EDLC, Modelling of EDLC; issues in Hardware-software Co-design, Data Flow Graph, state machine model, STUDENTSFOCUS.COM Sequential Program Model, concurrent Model, object-oriented Model.

UNIT IV RTOS BASED EMBEDDED SYSTEM DESIGN

Introduction to basic concepts of RTOS- Task, process & threads, interrupt routines in RTOS, Multiprocessing and Multitasking, Pre-emptive and non-pre-emptive scheduling, Task communication shared memory, message passing-, Inter process Communication – synchronization between processes-semaphores, Mailbox, pipes, priority inversion, priority inheritance.

UNIT V EMBEDDED SYSTEM APPLICATION AND DEVELOPMENT

Case Study of Washing Machine- Automotive Application- Smart card System Application-ATM machine – Digital camera

TEXTBOOKS:

1. Peckol, "Embedded system Design", John Wiley & Sons,2010
2. Lyla B Das," Embedded Systems-An Integrated Approach", Pearson, 2013
3. Shibu. K.V, "Introduction to Embedded Systems", 2e, Mc graw Hill, 2017.

REFERENCES

1. Raj Kamal, 'Embedded System-Architecture, Programming, Design', Mc Graw Hill, 2013.
2. C.R.Sarma, "Embedded Systems Engineering", University Press (India) Pvt. Ltd, 2013.
3. Tammy Noergaard, "Embedded Systems Architecture", Elsevier, 2006.
4. Han-Way Huang, "Embedded system Design Using C8051", Cengage Learning, 2009.
5. Rajib Mall "Real-Time systems Theory and Practice" Pearson Education, 2007.

ROBOTICS PROGRAMMING LAB			
Course Code:	**CA681**	**Course Credits:**	**2**
Course Category:	**CC-L**	**Course (U / P)**	**U / P**
Course Year (U / P):	**2P**	**Course Semester (U / P):**	**3P**
No. of Lab (Hrs/Week) / Total No. of Lab	**03 / 10**	**End Sem. Exam Hours:**	**3**

COURSE OBJECTIVES

6. To write programming for simple operations
7. To introduce different types of robotics and demonstrate them to identify different parts and components.
8. To introduce the concepts of robotic manufacturing system and work cells
9. Familiarize with the robot programming and control
10. To learn robot programming and simulation for industrial application.

COURSE OUTCOME

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

1. Develop the robot programming for the given application
2. perform the singularity analysis of robotics
3. Interface the vision system with robotic arm to develop the machine vision applications.
4. To learn robot programming and simulation for any industrial process
5. use of any robotic simulation software to model the different types of robots and calculate work volume for different robots

1. Design and develop the manufacturing cell using virtual robot simulator.
2. Develop a TCP and work-object for Industrial Robot using Robot simulator.
3. Develop a work-object for Industrial Robot using Robot simulator.
4. Develop the robot programming for pick and place of objects.
5. Develop the robot programming for material handling applications.
6. Develop the robot programming for welding process.
7. Singularity analysis using Robot simulator.
8. Interface and configure the vision system with Industrial Robot.
9. Part identification based on color & pattern and separate the components using vision system and Robot.
10. Quality control using Industrial Robot with vision system.