VIRUDHUNAGAR HINDU NADARS' SENTHIKUMARA NADAR COLLEGE
(An Autonomous Institution Affiliated to Madurai Kamaraj University)
[Re-accredited with 'A' Grade by NAAC]
Virudhunagar – 626 001.

Course Name: Master of Computer Applications
Discipline : MCA

Rules and regulations, Course Scheme and Scheme of Examinations
(For those who join in June 2022 and after)

Course Scheme:

SEMESTER III

Semester	**Part**	**Course Title**	**Hours**	**Credit**	**Marks**			**Course Code**	**Courses having focus on employability/ entrepreneurship/ Skill development**	**Revised / New /No Change / Interchanged/ Percentage of revision**
					I	**E**	**Total**			
III	Core 12	Cryptography and Network Security	4	4	40	60	100	P20CAC31/ **P22CAC31**	Skill development	No Change
	Core 13	Principles of Compiler Design	4	4	40	60	100	P20CAC32/ **P22CAC32**	Skill development	No Change
	Core 14	Artificial Intelligence	4	4	40	60	100	**P22CAC33**	Employability/ Entrepreneurship/ Skill development	**7% Revised**
	Core 15	Internet of Things	4	4	40	60	100	P20CAC34/ **P22CAC34**	Employability/ Entrepreneurship/ Skill development	No Change
	Elective III	Software Project Management / Theory of Computation /Soft Computing	5	5	40	60	100	**P22CAE31** P20CAE32/ **P22CAE32** P20CAE33/ **P22CAE33**	Employability/ Entrepreneurship/ Skill development Employability Skill development	**5% Revised** No Change No Change
	Core 16 – Lab	**LAB**: Android Programming	5	3	40	60	100	**P22CAP31**	Employability/ Entrepreneurship/ Skill development	**2% Revised**
	Core 17 – Lab	**LAB**: Dot Net Programming	4	2	40	60	100	P20CAP32/ **P22CAP32**	Employability/ Entrepreneurship/ Skill development	No Change
	Total		**30**	**26**					P19CA4PV	

SEMESTER IV

Semester	**Part**	**Course Title**	**Hours**	**Credit**	**Marks**			**Course Code**	**Courses having focus on employability/ entrepreneurship/ Skill development**	**Revised / New /No Change / Interchanged/ Percentage of revision**
					I	**E**	**Total**			
IV	Core 18	R Programming	4	4	40	60	100	P20CAC41/ **P22CAC41**	Employability/ Skill development	No Change
	Core 19 - Lab	**LAB**: Angular	4	2	40	60	100	**P22CAP41**	Employability/ Entrepreneurship/ Skill development	**New**
		Project and Viva-Voce	-	6	50	50	100	**P22CA4PV**	Employability/ Entrepreneurship/ Skill development	**Mark Revised**
	Total		**8**	**12**						

Course Title: Cryptography and Network Security	**Total Hours: 60 Hours**
Course Code: P20CAC31/ **P22CAC31**	**Total Credits: 4**

Course Outcome:
Students, after successful completion of the course, will be able to:

COs	CO Statement
CO1:	To understand the fundamentals of Cryptography.
CO2:	To acquire Knowledge on standard algorithms used to provide confidentiality, Integrityaand authentication.
CO3:	To understand the various key distribution and management schemes.
CO4:	To understand how to deploy encryption techniques to secure data in transit across Data networks.
CO5:	To design security applications in the field of information technology.

Unit I **12 Hours**

Introduction – Security Goals; Cryptographic Attacks; Services and Mechanism; Techniques for security Goals Implementation.

Traditional Symmetric-Key Ciphers - Symmetric-Key Ciphers; Categories of Traditional Ciphers; Stream and Block Ciphers.

Introduction to Modern Symmetric-Key Ciphers - Modern Block Ciphers; Components of a Modern Block Cipher; Two classes of Product Ciphers; Attacks Designed for Block Ciphers; Modern Stream Ciphers.

Unit II **12 Hours**

Data Encryption Standard (DES) - History of Data Encryption Standard (DES); DES Structure; Security of DES.

Advanced Encryption Standard (AES) - History of Advanced Encryption Standard (AES); Transformations used by AES; Key Expansion; The AES Cipher; Analysis of AES.

Unit III **12 Hours**

Asymmetric-Key Cryptography - Difference between Symmetric-Key and Asymmetric; Key Cryptosystems; RSA Cryptosystem.

Message Integrity and Message Authentication – Message Integrity; Random Oracle Model; Message Authentication.

Unit IV **12 Hours**

Digital Signature – Comparison; Process; Services; Attacks on Digital Signature; Digital Signature Schemes.

Entity Authentication - Entity Authentication and Message Authentication; Passwords-based Authentication; Challenge-Response Protocols; Zero-Knowledge Protocols; Biometrics.

Unit V **12 Hours**

Key Management - Symmetric-Key Distribution; Kerberos; Symmetric-Key Agreement.

Security at the Application Layer - E-mail System; Secure/Multipurpose Internet Mail Extension(S/MIME).

System Security – Description of the System; Malicious Programs; Intrusion Detection Systems (IDS).

Text Book:

Behrouz A Forouzan, Debdeep Mukhopadhyay, "Cryptography and Network Security**".** Third Edition,Tata McGraw Hill Education Private Limited, Fifth re print 2017.

Unit I	Chapter 1 Chapter 3 Chapter 5
Unit II	Chapter 6 (Page No:137 -150, 156, 157) Chapter 7 (Page No: 169 -190, 192 -195)
Unit III	Chapter 10 (Page No: 259 – 276) Chapter 11
Unit IV	Chapter 13(Page No: 347 - 365) Chapter 14
Unit V	Chapter 15(Page No: 389 - 402) Chapter 16 (Page No:417– 419,438 - 447) Chapter 19 (Page No:525, 526, 535 - 547)

Reference Books:

1) Atul Kahate, "Cryptography and Network Security", Third Edition, McGraw Hill Education Private Limited, Eighth Reprint2017.

2) William Stallings, "Cryptography and Network Security", PHI,2008.

E – Resources

1. https://www.geeksforgeeks.org/cryptography-and-network-security-principles/
2. https://www.tutorialspoint.com/cryptography/index.htm
3. https://www.gatevidyalay.com/tag/cryptography-and-network-security-tutorial/

Course Title: Principles of Compiler Design	**Total Hours: 60 Hours**
Course Code: P20CAC32/ **P22CAC32**	**Total Credits: 4**

Course Outcomes:

CO1:	To understand the functionality of each phase involved in Compilation process.
CO2:	Implement the parsing techniques including Bottom-up and Top-down parsing for the given programming construct described in Context Free Grammar.
CO3:	To Constructing the different parsing table.
CO4:	To Generate the intermediate code and the implementation of symbol table

CO5:	To Apply the optimization techniques and generate the machine code.

Unit I **12 Hours**

Introduction to Compilers - Compilers and translators; Why do we need translators; The structure of a compiler; Lexical Analysis; Syntax Analysis; Intermediate Code Generation; Optimization; Code Generation; Bookkeeping; Error Handling; Compiler Writing Tools.

Finite Automata and Lexical Analysis - The role of the lexical analyzer; A simple approach to the design of lexical analyzers; Regular Expressions; Finite automata.

Unit II **12 Hours**

The Syntactic Specification of Programming Languages - Context free Grammars; Derivations and parse trees.

Basic Parsing Techniques - Parsers; Shift reduce parsing; Operator precedence parsing; Top down parsing; Predictive parsers.

Unit III **12 Hours**

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 IV **12 Hours**

Syntax-Directed Translation - Syntax directed translation schemes; Implementation of Syntax directed translators; Intermediate code; Postfix notation; Parse trees and syntax trees; Three address code, quadruples and triples.

Symbol Tables - The contents of a symbol table; Data structures for symbol tables.

Unit V **12 Hours**

Error Detection and Recovery – Errors; Lexical phase errors; Syntactic phase errors; Semantic errors.

Introduction to Code Optimization - The principal sources of optimization; Loop optimization; The DAG representation of basic blocks.

Code Generation - Object programs; Problems in code generation; A machine Model; A simple Code generator; Peephole optimization.

Text Book:

Alfred V.Aho, Jeffrey D.Ullman, ***Principles of Compiler Design,*** Narosa Publishing House, 2002.

Unit I	Chapter 1, Chapter 3 (3.1 to 3.4)
Unit II	Chapter 4 (4.1& 4.2), Chapter 5
Unit III	Chapter 6 (6.1 to 6.8)
Unit IV	Chapter 7 (7.1 to 7.6), Chapter 9 (9.1,9.2)

Unit V	Chapter 11, Chapter 12(12.1 - 12.3), Chapter 15(15.1 - 15.4, 15.7)

Reference Book:

1. Alfred V.Aho, Monica S.Lam, Ravi Sethi, Jeffrey D.Ullman, ***Compilers Principles, Techniques and Tools***, Second edition, Pearson Publications, 2007.

E – Resources:

1. https://archive.nptel.ac.in/courses/106/105/106105190/
2. https://archive.nptel.ac.in/courses/106/104/106104123/
3. https://archive.nptel.ac.in/courses/106/108/106108113/
4. https://archive.nptel.ac.in/content/storage2/courses/106104072/ui/Course_home-1.htm
5. https://www.tutorialspoint.com/compiler_design/index.htm
6. https://www.geeksforgeeks.org/compiler-design-tutorials/

Course Title: Artificial Intelligence	**Total Hours: 60 Hours**
Course Code: P22CAC33	**Total Credits: 4**

Course Outcome:
Students, after successful completion of the course, will be able to:

CO1:	Understand the basics of Artificial Intelligence
CO2:	Gain knowledge on Search Techniques.
CO3:	Understand how to encode knowledge.
CO4:	Learn problem solving by collecting evidence
CO5:	Learn advance concepts in Artificial Intelligence

Unit I **12 Hours**
What is Artificial Intelligence - The AI problems; The Underlying Assumption; What is an AI Technique; The Level of the Model; Criteria for Success.
Problems, Problem Spaces and Search - Defining the Problem as a State Space Search; Production Systems; Problem Characteristics; Production System Characteristics; Issues in the Design of Search Programs; Additional Problems.

Unit II **12 Hours**
Heuristic Search Techniques - Generate-and-Test; Hill Climbing; Best-First Search; Problem Reduction; Constraint Satisfaction; Means-Ends Analysis.
Knowledge Representation Issues - Representations and Mappings; Approaches to Knowledge Representation; Issues in Knowledge Representation; The Frame Problem.

Unit III **12 Hours**

Using Predicate Logic - Representing Simple Facts in Logic; Representing Instance and Isa Relationships; Computable Functions and Predicates; Resolution; Natural Deduction.

Representing Knowledge Using Rules - Procedural versus Declarative Knowledge; Forward versus Backward Reasoning; Matching.

Unit IV **12 Hours**

Statistical Reasoning - Probability and Bayes Theorem; Certainty Factors and Rule-based Systems; Bayesian Networks; Dempster-Shafer Theory; Fuzzy Logic.

Unit V **12 Hours**

Game Playing - Overview; The Minmax Search Procedure; Adding Alpha-Beta Cutoffs; Additional Refinements; Iterative Deepening.

Expert Systems - Representing and Using Domain Knowledge; Expert System Shells; Explanation; Knowledge Acquisition.

Text Book:

Elaine Rich and Kevin Knight, ***Artificial Intelligence***,Tata McGraw Hill Publishing Company Limited, New Delhi,Third Edition,2009.

Unit I	Chapter 1 (1.1 - 1.5), 2 (2.1 - 2.6)
Unit II	Chapter 3 (3.1 – 3.6), 4 (4.1 – 4.4)
Unit III	Chapter 5 (5.1 – 5.5), Chapter 6 (6.1, 6.3, 6.4)
Unit IV	Chapter 8 (8.1 – 8.5)
Unit V	Chapter 12 (12.1 – 12.5), 20 (20.1 – 20.4)

Reference Books:

1. Stuart Russell, Peter Norvig, ***Artificial Intelligence: A Modern Approach***, Pearson education, Third edition, 2014.
2. Richard E Neapolitan, ***Artificial Intelligence: With an Introduction to Machine Learning***, CRC Press, Second Edition, 2018.

E – Resources:

1. https://www.youtube.com/watch?v=qozouJ_Mu-o
2. https://techvidvan.com/tutorials/ai-heuristic-search/
3. https://www.tutorialride.com/artificial-intelligence/knowledge-representation-nlp-in-ai.htm
4. https://www.javatpoint.com/expert-systems-in-artificial-intelligence
5. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_expert_systems.htm
6. https://www.youtube.com/watch?v=kUcSgSzuRSs&t=287s
7. https://www.youtube.com/watch?v=XCPZBD9lbVo&t=8s

Course Title: Internet of Things	**Total Hours: 60 Hours**
Course Code: P20CAC34/ **P22CAC34**	**Total Credits: 4**

Course Outcomes:

COs	CO Statement
CO1:	Understand the concepts of Internet of Things.
CO2:	Determine the Market Perspective of IoT and Data Management in IoT.
CO3:	Design IoT applications in different domain and be able to analyze their performance.
CO4:	Implement basic IoT applications on embedded platform.
CO5:	Application of IoT in Industrial and Commercial Building Automation and Real World Design Constraints

Unit I **12 Hours**

Introduction to IoT: Introduction; Physical design of IoT; Logical design of IoT; IoT enabling Technologies; IoT Levels & Deployment templates.

Domain Specific IoTs: Introduction; Home Automation; Cities; Environment; Energy; Retail; Logistics; Agriculture; Industry; Health & Lifestyle.

Unit II **12 Hours**

IoT & M2M: Introduction; M2M; Difference between IoT and M2M; SDN and NFV for IoT.

IoT System Management with NETCONF-YANG: Need for IoT Systems Management; SNMP; Network Operator Requirements; NETCONF; YANG; IoT Systems Management with NETCONF-YANG.

Unit III **12 Hours**

IoT Platforms Design Methodology: Introduction; IoT Design Methodology; Case Study on IoT System for Weather Monitoring; Motivation for using Python.

Case studies illustrating IoT Design: Cities

Unit IV **12 Hours**

IoT Systems – Logical Design using Python: Introduction; Installing Python; Python Data types & Data Structures; Control Flow; Functions; Modules; Packages; File Handling; Date/ Time Operations; Classes; Python Packages of Interest for IoT.

Unit V **12 Hours**

IoT Physical Devices & Endpoints: What is an IoT Device; Exemplary Device: Raspberry Pi; About the Board; Linux on Raspberry Pi; Raspberry Pi Interfaces; Programming Raspberry Pi with Python; Other IoT Devices.

IoT Physical Servers & Cloud Offerings: Amazon Web Services for IoT.

Text Book:

Arshdeep Bahga, Vijay Madisetti, **Internet of Things: A Hands-On Approach**, Universities Press (India) Private Limited; First Edition, 2018.

Unit I	Chapter 1, 2
Unit II	Chapter 3, 4

Unit III	Chapter 5, 9 (9.3)
Unit IV	Chapter 6
Unit V	Chapter 7, 8 (8.6)

Reference Books:

1. Waltenegus Dargie,Christian Poellabauer. ***Fundamentals of Wireless Sensor Networks: Theory and Practice***, WILEY Publication; First Edition 2010.
2. Francis daCosta. ***Rethinking the Internet of Things: A Scalable Approach to Connecting Everything***, Apress Publications; First Edition,2013

E - Resources:

1. https://www.javatpoint.com/iot-internet-of-things
2. https://www.guru99.com/iot-tutorial.html
3. https://www.tutorialspoint.com/iot_internet_of_things/index.asp
4. https://w3cschoool.com/iot-tutorial
5. https://www.arduino.cc/

Elective – III (a)

Course Title : Software Project Management	**Total Hours : 75 Hours**
Course Code : P22CAE31	**Total Credits : 5**

Course Outcomes

COs	CO Statement
CO1:	Understand the practices and methods for successful software project management.
CO2:	Identify techniques for requirements, policies and decision making for effective resource management.
CO3:	Apply the evaluation techniques for estimating cost, benefits, schedule and risk.
CO4:	Devise a framework for software project management plan for activities, risk, monitoring and control.
CO5:	Devise a framework to manage people.

Unit I **15 Hours**

Introduction to Software Project Management - Introduction; Why is Software Project Management Important; What is a Project?; Software Projects versus Other Types of Project; Contract Management and Technical Project Management; Activities Covered by Software Project Management, Plans, Methods and Methodologies; Some Ways of Categorizing Software Projects; Stakeholders; Setting Objectives; The Business Case; Project Success and Failure; What is Management?; Management Control; Traditional versus Modern Project Management Practices.

An Overview of Project Planning - Introduction to Step Wise Project Planning; Step 0 : Select Project; Step 1: Identify Project Scope and Objectives; Step 2: Identify Project Infrastructure; Step 3: Analyse Project Characteristics; Step 4: Identify Project Products and Activities; Step 5: Estimate Effort for Each Activity; Step 6: Identify Activity Risks; Step 7:

Allocate Resources; Step 8: Review/Publicize Plan; Steps 9 and 10 : Execute Plan/ Lower Levels of Planning.

Unit II **15 Hours**

Project Evaluation and Programme Management – Introduction; A Business Case; Project Portfolio Management; Evaluation of Individual Projects; Cost Benefit Evaluation Techniques; Risk Evaluation;

Activity Planning – Introduction; Objectives of Activity Planning; When to Plan; Project Schedules; Projects and Activities; Sequencing and Scheduling Activities; Network Planning Models; Formulating a Network Model; Adding the Time Dimension; The Forward Pass; The Backward Pass; Identifying the Critical Path; Activity Float; Shortening the Project Duration; Identifying Critical Activities; Activity on Arrow Networks.

Unit III **15 Hours**

Risk Management - Introduction; Risk; Categories of Risk; A Framework for Dealing with Risk; Risk Identification; Risk Assessment; Risk Planning; Risk Management; Applying the PERT Technique; Monte Carlo Simulation.

Monitoring and Control – Introduction; Creating the Framework; Collecting the Data; Review; Project Termination Review; Visualizing Progress; Cost Monitoring; Earned Value Analysis; Prioritizing Monitoring; Getting the Project Back to Target; Change Control; Software Configuration Management.

Unit IV **15 Hours**

Managing Contracts – Introduction; Types of Contracts; Stages in Contract Placement; Typical Terms of a Contract; Contract Management; Acceptance.

Managing people in Software Environments – Introduction; Understanding Behaviour; Organizational Behaviour: A Background; Selecting the Right Person for the Job; Instruction in the Best Methods; Motivation; The Oldham; Hackman Job Characteristics Model; Stress; Health and Safety; Some Ethical and Professional Concerns.

Unit V **15 Hours**

Working in Teams – Introduction; Becoming a Team; Decision Making; Organization and Team Structures; Leadership.

Software Quality – Introduction; The Place of Software Quality in Project Planning; The Importance of Software Quality; Defining Software Quality; Product versus Process Quality Management; Quality Management Systems; Process Capability Models; Techniques to Help Enhance Software Quality; Testing.

Text Books

Bob Hughes, Mike Cotterell, Rajib Mall, "Software Project Management", Tata McGraw Hill Education, Fifth Edition, 2013.

Unit I	Chapter 1,3
Unit II	Chapter 2 (2.1 – 2.6), 6
Unit III	Chapter 7 (7.1 – 7.8, 7.10, 7.11), 9
Unit IV	Chapter 10, 11
Unit V	Chapter 12 (12.1 - 12.4, 12.9), 13 (13.1 – 13.4, 13.7 – 13.11)

Reference Books

1. Pankaj Jalote, "Software Project Management in Practise", Pearson Education, 2002.
2. Robert T. Futrell, Donald F. Shafer, Linda I. Safer, Quality Software Project Management, Pearson Education, Asia, 2002.

E- Resources

1. https://www.geeksforgeeks.org/software-engineering-software-project-management-spm/
2. https://www.castsoftware.com/research-labs/risk-management-in-software-development-and-software-engineering-projects
3. http://www.jmpcollege.org/Adminpanel/AdminUpload/Studymaterial/Ch02_project_evaluation.pdf
4. https://www.brainkart.com/article/Managing-People-and-Organizing-Teams_7961/
5. https://onlinecourses.nptel.ac.in/noc19_cs70/preview

Elective – III (b)

Course Title : Theory of Computation	**Total Hours :75 Hours**
Course Code : P20CAE32/ **P22CAE32**	**Total Credits : 5**

Course outcomes:

CO1:	To Design finite state automata and regular expression for a language specification and convert one form to another form
CO2:	To Learn to write Context Free Grammars and normalize.
CO3:	To Design push down automata (PDA) for languages and convert CFG to PDA and vice versa
CO4:	To Construct Turing machine by applying different techniques
CO5:	To Find undecidability in languages

Unit I: **15 Hours**

Finite Automata: Deterministic Finite Automata; Definition of a Deterministic Finite Automaton; How a DFA Processor Strings; Simpler Notations for DFA's; Extending the Transition Function to Strings; The Language of a DFA.

Nondeterministic Finite Automata: An Informal View of Nondeterministic Finite Automata;Definition of Nondeterministic Finite Automata; The Extended Transition Function;The Language of an NFA; Equivalence of Deterministic and Nondeterministic Finite Automata.

Finite Automata with Epsilon-Transitions: Uses of Epsilon; Transitions; The Formal Notation for an Epsilon-NFA; Epsilon-Closures; Extended Transitions and Languages for Epsilon-NFA's; Eliminating Epsilon-Transitions.

Regular Expressions: The Operators of Regular Expressions; Building Regular Expression; Precedence of Regular-Expression Operators.

Finite Automata and Regular Expressions: From DFA's to Regular Expressions; Converting DFA's to Regular Expressions by Eliminating States; Converting Regular Expressions to Automata.

Unit II: **15 Hours**

Proving Languages Not to Be Regular: The Pumping Lemma for Regular Languages; Applications of the Pumping Lemma.

Closure Properties of Regular Languages: Closure of Regular Languages Under Boolean Operations; Reversal; Homomorphisms; Inverse Homomorphisms.

Equivalence and Minimization of Automata: Testing Equivalence of States; Testing Equivalence of Regular Languages.

Context-Free Grammars: Definition of Context-Free Grammars; Derivation using a Grammar; Leftmost and Rightmost Derivations.

Parse Trees: Constructing Parse Trees; The Yield of a Parse Tree; Inference, Derivations, and Parse Trees.

Ambiguity in Grammars and Languages: Ambiguous Grammars; Removing Ambiguity from Grammars.

Unit III: **15 Hours**

Definition of the Pushdown Automaton: Informal Introduction; The Formal Definition of Pushdown Automata; A Graphical Notation for PDA's; Instantaneous Descriptions of a PDA.

Equivalence of PDA's and CFG's: From Grammars to Pushdown Automata; From PDA's to Grammars.

Deterministic Pushdown Automata: Definition of a Deterministic PDA; Regular Languages and Deterministic PDA's.

Normal Forms for Context-Free Grammars: Eliminating Useless Symbols; Eliminating Epsilon-Productions; Eliminating Unit Productions; Chomsky Normal Form.

The Pumping Lemma for Context-Free Languages: Statement of the Pumping Lemma; Applications of the Pumping Lemma for CFL's.

Unit IV: **15 Hours**

The Turing Machine: Notation for the Turing Machine; Instantaneous Descriptions for Turing Machines; Transition Diagrams for Turing Machines; The Language of a Turing Machine; Turing Machines and Halting.

Programming Techniques for Turing Machines: Storage in the State; Multiple Tracks; Subroutines.

Extensions to the Basic Turing Machine: Multitape Turing Machines; Equivalence of One-Tape and Multitape TM's; Nondeterministic Turing Machines.

Unit V: **15 Hours**

Undecidable Problems About Turing Machines: Reductions; Turing Machines That Accept the Empty Language; Rice's Theorem and Properties of the RE Languages.

Post's Correspondence Problem: Definition of Post's Correspondence Problem; The "Modified" PCP; Completion of the Proof of PCP Undecidability.

The Classes P and NP: Problems Solvable in Polynomial Time; Polynomial-Time Reductions; NP-Complete Problems.

Text Book:

John E.Hopcroft, Rajeev Motwani, Jeffery D.Ullman, "Introduction to Automata Theory, Languages, and Computation", Pearson Education., 3rd Edition, 2009.

Unit I	Chapter 2: **2.2:** 2.2.1 – 2.2.5, **2.3:**2.3.1 – 2.3.5, **2.5:** 2.5.1 – 2.5.5 Chapter 3: **3.1:** 3.1.1 – 3.1.3, **3.2:** 3.2.1 – 3.2.3
Unit II	Chapter 4: **4.1:**4.1.1 – 4.1.2, **4.2:**4.2.1 – 4.2.4, **4.4:**4.4.1 – 4.4.2 Chapter 5: **5.1:** 5.1.2 – 5.1.4,**5.2:** 5.2.1 – 5.2.3, **5.4:** 5.4.1 – 5.4.2
Unit III	Chapter 6: **6.1:** 6.1.1 – 6.1.4, **6.3:**6.3.1 – 6.3.2, **6.4:** 6.4.1 – 6.4 Chapter 7: **7.1:** 7.1.1, 7.1.3 – 7.1.5, **7.2:** 7.2.2, 7.2.3
Unit IV	Chapter 8: **8.2:**8.2.2– 8.2.6**, 8.3:** 8.3.1 – 8.3.3, **8.4:**8.4.1,8.4.2, 8.4.4
Unit V	Chapter 9: **9.3:**9.3.1 – 9.3.3, **9.4:**9.4.1 – 9.4.3 Chapter 10: **10.1:** 10.1.1,10.1.5,10.1.6

Reference Book:

Harry R.Lewis,Christos H.Papadimitriou, ***Elements Of The Theory Of Computation***, Dorling kindersley (India) Pvt. Ltd., 2nd Edition, 2008.

e-Resources:

1. https://www.geeksforgeeks.org/introduction-of-theory-of-computation/
2. https://www.javatpoint.com/automata-tutorial
3. https://www.tutorialspoint.com/what-is-the-theory-of-computation
4. https://archive.nptel.ac.in/courses/106/104/106104148/

Elective – III (c)

Course Title : Soft Computing	**Total Hours :75 Hours**
Course Code : P20CAE33/ **P22CAE33**	**Total Credits : 5**

Course Outcome:

CO1:	Learn the basic concepts of Soft Computing and gain knowledge of Artificial Neural Network.
CO2:	Learn associative memory and unsupervised learning
CO3:	Learn fuzzy sets and its representations
CO4:	Learn Classical Logic, Fuzzy Logic and decision making
CO5:	Learn Genetic algorithms and pattern recognition.

Unit I: **15 Hours**

Introduction: Introduction to Soft Computing; Evolutionary Computing; Hard Computing Vs Soft Computing; Soft Computing Methods; Recent Trends in Soft Computing. **Fundamentals of Artificial Neural Network:** Introduction; Model of Biological Neuron; Mathematical Model of Neuron; ANN Architecture; Learning Rules; Learning Paradigms; Perceptron Network; Adaline and Madaline Networks; Applications of Neural Network.

Unit II: **15 Hours**

Associative Memory: Introduction; Autoassociative Memory; Hetero-associative Memory; Bidirectional Associative Network; Applications of Associative Memory.

Unsupervised Learning: Introduction; Winner-Takes-All Network; Learning Vector Quantization; Self-organization Map; Adaptive Resonance Theory; Neocognitron; Applications of Unsupervised Learning.

Unit III: **15 Hours**

Associate Models: Hopfield Network; Boltzmann Network; Simulated Annealing; Application of Networks.

Classical Sets and Fuzzy Sets: Crisp Sets; Fuzzy Sets; History and Origin; Fuzzy Sets; Basic Concepts; Paradigm Shift; Representation of Fuzzy Sets.

Unit IV: **15 Hours**

Classical Logic and Fuzzy Logic: Logic; Interval Analysis; Fuzzy Numbers; Fuzzy Logic.

Fuzzy Decision Making: Introduction; Individual Fuzzy Decision Making; Multiperson Decision Making; Multicriteria Decision Making; Multistage Decision Making.

Unit V: **15 Hours**

Genetic Algorithms: History of Evolutionary Computing; Crossover and Mutation Properties; Genetic Algorithm Cycle; Fitness Function; Applications of Genetic Algorithm.

Application of Soft Computing Techniques: Pattern Recognition; Image Processing; Application of Soft Computing in Real Estate.

Text Book:

B.K. Tripathy, J. Anuradha , ***Soft Computing Advances and Applications,*** 2015.

Unit I	Chapter 1, Chapter 2
Unit II	Chapter 4, Chapter 5
Unit III	Chapter 6, Chapter 7 (7.1 – 7.5)
Unit IV	Chapter 9, Chapter 11
Unit V	Chapter 13, Chapter 17 (17.1 – 17.3)

Reference Book:

Dilip K. Pratihar, ***Soft Computing Fundamentals and Applications***, 2015.

e-Resources:

1. **https://www.javatpoint.com/what-is-soft-computing**
2. **https://archive.nptel.ac.in/courses/106/105/106105173/**

Core - 16

Course Title : LAB: Android Programming	**Total Hours :75 Hours**
Course Code : P22CAP31	**Total Credits : 3**

Course Outcomes

COs	CO Statement
CO1:	Develop simple android application using Eclipse IDE.
CO2:	Exploring and Designing Android UI using views and view groups.
CO3:	Implementing activities and intents in developing applications.
CO4:	Develop interactive applications using data persistence and Messaging.
CO5:	Deployment of Android Application as APK.

1. Develop a sample Android application using Eclipse.
2. Develop an Android application using Activities.
3. Develop an Android application using Intents. (Explicit)
4. Develop an Android application using Built-in Intents.
5. Develop an Android application using Fragments.
6. Develop an Android application using Action Bar.
7. Develop an Android application for registration form using Basic Views.
8. Develop an Android application using Picker Views.
9. Develop an Android application using List Views.
10. Develop an Android application using Spinner Views.
11. Develop an Android application for Gallery using Image Views.
12. Develop an Android application using Menus. (Option Menu and Context Menu)
13. Develop an Android application to save and retrieve user data using Preferences.
14. Develop an Android application using Databases.
15. Develop an Android application using Content Providers.
16. Develop an Android application using 'Messaging' and 'E-Mail'.

Core 17 – Lab

Course Title : LAB: Dot Net Programming	**Total Hours: 60 Hours**
Course code: P20CAP32/ **P22CAP32**	**Total Credits: 2**

Course Outcomes:

COs	CO Statement
CO1:	Understand the basic concepts VB.NET data structure
CO2:	Know the knowledge of GUI tools in VB.NET windows application
CO3:	Understand Classes and Objects, Inheritance in C#.NET
CO4:	Understand function, procedure in ASP.NET
CO5:	Work on Database Connectivity in VB.NET, C#.NET and ASP.NET

List of Programmes:

1. Write a program in VB.Net console application to perform Array List Operations.
2. Write a program in VB.Net console application to demonstrate Constructor Overloading.
3. Write a program in VB.Net windows application to design a Scientific Calculator.
4. Develop Departmental Store application in VB.Net windows application.
5. Write a program in VB.Net to perform Number checking (like Armstrong, Adam, Palindrome, Perfect)
6. Write a C#.net console application to implement Multi Level Inheritance
7. Write a C#.net windows application to implement Regular Expression.
8. Develop a simple animation using C#.net windows application
9. Develop C#.net windows application for library management system
10. Write an ASP .Net program to find the Factorial of a given number by using Function
11. Write an ASP .Net program to generate a Fibonacci series by using Subroutine
13. Write an ASP .Net program for form validation by using all validation controls
14. Create an Asp.net web application for employ payroll processing.

SEMESTER - IV

Course Title: R Programming	**Total Hours: 60 Hours**
Course Code: P20CAC41/ **P22CAC41**	**Total Credits: 4**

Course Outcomes:

COs	CO Statement
CO1:	To understand the basic concepts of R and gain knowledge on R programming.
CO2:	To gain knowledge on data storage in R Programming and to import and export other File formats to R Programming
CO3:	To gain knowledge on mathematical operations in R Programming.
CO4:	To understanding the graphical operations in R Programming.
CO5:	To gain knowledge on Customization of Plots.

Unit I: **12 Hours**

A Short Introduction to R: Introduction; Installing R; Getting Started; Some Information on R Commands; Special Values; Objects; Functions; Simple Manipulations - Numbers and Vectors; Matrices and Arrays; Factors; Lists; Data Frames.

Programming Using R: Introduction; Function Creation; Scripts; Logical Operators; Conditional Statements; Loops in R; Switch Statement.

Unit II: **12 Hours**

Lists and Data Frames: Introduction; Creating a List; Common List Operations; Recursive List; Creating a Data Frame; Common Data Frame Operations; Using lapply() and sapply() Functions.

Import and Export: Introduction; Saving and Loading R data; Import and Export to CSV Files; Importing Data from SAS; Import and Export via ODBC.

Unit III: **12 Hours**
Mathematical and Statistical Concepts: Introduction; Maximum and Minimum; Frequency Distribution; Frequency Distribution Types; Measure of Central Tendency; Measure of Dispersion; Correlation.

Unit IV: **12 Hours**
Graphics: Introduction; Basic Plots; Labeling and Documenting plots; Adjusting the Axes; Specifying Colors; Specifying Fonts; Specifying Sizes; Plotting Symbols.

Unit V: **12 Hours**
Customised Plotting: Introduction; Change of Plotting Line Style; Adding Items on a Plot; Higher Dimensional Data Display; Changing the Plot Settings Using par() Function.

Text Book:
Dr Sandip Rakshit. **R For Beginners.** McGraw Hill Education (India) Private Limited; First Edition, 2017.

Unit I	Chapter 1, Chapter 2
Unit II	Chapter 3, Chapter 6
Unit III	Chapter 7
Unit IV	Chapter 12
Unit V	Chapter 13

Reference Books:

1. Garrett Grolemund. **Hands on Programming with R**, O'Reilly Media Inc.
2. Andrie de Vries. **R for Dummies,** John Wiley & Sons Inc, Second Edition, 2015
3. Kun Ren. **Learning R Programming,** Packt Publishing Ltd, First Edition, 2016

e-Resources:

1. https://www.tutorialspoint.com/r/index.htm
2. https://www.w3schools.com/r/
3. https://www.javatpoint.com/r-tutorial
4. https://www.rstudio.com/
5. https://www.codecademy.com/learn/learn-r

Course Title : LAB: Angular	**Total Hours : 60 Hours**
Course Code : P22CAP41	**Total Credits : 2**

Course Outcomes

COs	CO Statement
CO1:	Create a basic sample application in Angular.
CO2:	Develop an application using Components and Directives.
CO3:	Implementing Services and Pipes in developing applications.
CO4:	Develop interactive applications using Reactive Forms and Custom Validators.

CO5:	Develop Angular applications using Routing.

1. Develop a sample Angular Application using AngularCLI.
2. Develop an Angular application using Interpolation.
3. Develop an Angular application using Property Binding.
4. Develop an Angular application using Two Way Binding.
5. Develop an Angular application using Structural Directives.
6. Develop an Angular application using Reactive Forms.
7. Develop a user authentication application using Firebase.
8. Develop an Angular application using Services.
9. Develop an Angular application using Pipes.
10. Develop an Angular application using Routing.

Course Title : Project and Viva-Voce	
Course Code : P22CA4PV	**Total Credits : 6**

(Industry/Institutional Based)

Course Outcomes:

CO1:	Students will have hands of experience of real life system development life cycle
CO2:	The students will learn to apply the technologies learnt during the course in real life projects
CO3:	Students will learn to work in real life project development environments involving deadlines and teamwork
CO4:	Students will learn to pick up and apply upcoming technologies in project development not covered during the course

Objectives:

- To solve real life problems in the Industry/Academic Institutions/Computer science research.
- The Project and Viva-voce is one that involves practical work for understanding and solving problems in the field of computing.
- Students will do individually Commercial or Technical Project based on their Industry /Academic Institutions needs.
- With the known/needed technologies they can develop the software.
