MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(AUTONOMOUS)

Affiliated to JNTUA, Anantapur & Approved by AICTE, New Delhi
Recognised Research Center
Accredited by NBA for CSE, ECE, EEE & ME
World Bank funded Institute
Recognised by UGC under the sections 2(f) and 12(B) of the UGC act 1956
Recognised as Scientific & Industrial Research Organization by DSIR of DST

ACADEMIC REGULATIONS

COURSE STRUCTURE

AND

DETAILED SYLLABUS

MASTER OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING

M.Tech(CSE) Regular Two Year P.G. Degree Course
(Applicable for the batches admitted from 2014-2015)

Course Structure and Syllabi

M.Tech. Computer Science and Engineering [CSE]

I YEAR - I Semester

S. No	Course code	Subject	Theory	Lab.	Credits	I.M	E.M	M.M
1.	14CSP11T01	Programming using python	4	0	4	40	60	100
2.	14CSP11T02	Advanced Data Structures and Algorithms	4	0	4	40	60	100
3.	14CSP11T03	Computer System Design	4	0	4	40	60	100
4.	14CSP11T04	Advance Databases	4	0	4	40	60	100
5.	14CSP11T05	Object Oriented Analysis Design and Design Pattern	4	0	4	40	60	100
6.	14CSP11E1a 14CSP11E1b 14CSP11E1c 14CSP11E1d	Elective-I Software Engineering Artificial Intelligence Formal Language and Automata Theory Grid Computing	4	0	4	40	60	100
7.	14CSP11P01	programming using python and Advanced data structures Lab	0	3	2	40	60	100
		Contact periods/week	24	3	26			
		Total	**27**		**26**	**280**	**420**	**700**

I YEAR - II Semester

S. No	Course code	Subject	Theory	Lab.	Credits	I.M	E.M	M.M
1.	14CSP12T06	Data ware Housing and Data mining	4	0	4	40	60	100
2.	14CSP12T07	Java and Web Technologies	4	0	4	40	60	100
3.	14CSP12T08	Mobile Computing	4	0	4	40	60	100
4.	14CSP12T09	Advance Computer netwoks	4	0	4	40	60	100
5.	14CSP12T10	Software Quality Assurance and Testing	4	0	4	40	60	100
6.	14CSP12E2a 14CSP12E2b 14CSP12E2c 14CSP12E2d	Elective-II Distributed Systems Cloud Computing Image Processing Compiler Construction	4	0	4	40	60	100
7.	14CSP12P02	Data Warehousing Data Mining lab and JWT Lab	0	3	2	40	60	100
		Contact periods/week	24	3	26			
		Total	**27**		**26**	**280**	**420**	**700**

II YEAR – (III & IV Semesters)

S. No	Course code	Subject	Credits	I.M	E.M	M.M
1	14CSP22S01	**Seminar**	2	100	-	100
2	14CSP22D01	**Project Work**	16	-	-	-

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

PROGRAMMING USING PYTHON

(14CSP11T01)

Course Objective:

- To make the student understand problem solving techniques and their applications
- *Students will be able to understand the syntax and semantics of python.*
- Get acquaintances with classes and objects, stacks and queues using python.

Course Outcomes:

After completion of the course the student will be able to

- Apply problem solving techniques to find solutions to problems.
- Able to use python effectively and implement solutions using it.
- Be capable to identity the stack and queues for a given problem or application. Improve logical and programming skills.

UNIT 1

Introduction to Unix:- Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, wc, lp, od, tar, gzip. Unix Utilities:- Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities, disk utilities, networking commands, unlink, du, df, mount, umount, find, unmask, ulimit, ps, w, finger, arp, ftp, telnet, rlogin.

UNIT II

INTRODUCTION TO PYTHON

Getting Started: Introduction to Python- an interpreted high level language, How Python Runs Programs, Program Execution, Execution Model Variations, The Python Interactive Prompt, System Command Lines and Files, Module Imports and Reloads, The IDLE User Interface, **Python Variables, expressions and statements**: Values and types, keywords, Operators Expressions, Interactive mode and script mode, String operations, Comments. **Functions & Modules:** Function calls, Type conversion functions, Math functions, Adding new functions, Definitions and uses, Flow of execution, Parameters and arguments, Random numbers, The time module, The math module. **Conditionals:** Conditional execution, Alternative execution, Chained conditionals, Nested conditionals. **Iteration:** Multiple assignment, Updating variables, the while statement, break, continue.

UNIT III

Strings: A string is a sequence, Traversal with for loop, String slices, Strings are immutable, Searching, Looping and counting, String methods, the in operator, String comparison. **Tuples:** Tuples are immutable, Tuple assignment, Tuples as return values, Lists and tuples, **Dictionaries** and tuples, Comparing tuples, Sequences of sequences, Debugging. **Lists:** Traversing a list, List operations, List slices, List methods. **Recursion:** Stack diagrams for recursive functions, Infinite. **Files:** Persistence, Reading and writing, Filenames and paths.

UNIT IV

Classes and Objects: User-defined types, Attributes, Instances as return values **Methods:** The init method, The str method, Operator overloading, Polymorphism. **Inheritance:** Importance, examples. **Event handling:** Key press events, Mouse events. **Exceptions:** Catching exceptions, Raising our own exceptions, the finally clause of the try statement.

UNIT-V

Definition and use of Stacks: Abstract data types, The Stack ADT, Implementing stacks with Python lists, pushing and popping, Using a stack to evaluate postfix, Parsing, Evaluating postfix. **Queues:** The Queue ADT, Linked Queue, Performance characteristics, Improved Linked Queue, Priority queue.

Text Books:

1. Unix and shell Programming Behrouz A. Forouzan, Richard F. Gilberg.Thomson.
2. Learning Python, 4th Edition Mark Lutz's, O'Reilly .
3. Think Python - How to Think Like a Computer Scientist, Green Tea Press, Needham, Massachusetts, Allen Downey, Version 2.0.13, June 2014.
4. How to Think Like a Computer Scientist: Learning with Python 3, Peter Wentworth, Jeffrey Elkner, Allen B. Downey and Chris Meyers, Documentation Release 3rd Edition.

Reference Books:

1.How to Solve it by Computer by R.G. Dromey, Pearson.

2.Introduction to Computation and Programming using Python by John V.Guttag, 3.Spring 2013 Edition, The MIT Press Cambridge, Massachusetts, London, England.

3.Python for Everyone by Cay Horstmann, San Jose State University, Rance D. Necaise, College of William and Mary, Wiley.

4.Head First Python,Paul Barry, O'Reilly .

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

ADVANCED DATA STRUCTURES AND ALGORITHMS

(14CSP11T02)

Course Objectives:

- To develop skills to design and analyze linear and nonlinear data structures.
- Develop algorithms for manipulating linked lists,stscks,queues,trees,and graphs.
- Develop recursive algorithms as they apply to trees and graphs.
- To develop a base for advanced computer study.

Course Outcomes:

After completion of the course the student will be able to

- Study variety of advanced abstract data type (ADT) and data structures and their Implementations.Identify and apply the suitable data structure for the given real world problem.

UNIT-I

C++ Programming: Objects, Class Definition, Class Members, Access Control, Class Scope, Constructors and destructors, parameter passing methods, Inline functions, , dynamic memory allocation and deal location (new and delete), exception handling. Function over Loading.

Object-based Programming: Generic Programming- Function and class templates, Inheritance basics, base and derived classes, inheritance types, base class access control, runtime polymorphism using virtual functions.

UNIT II

Data structures-Linear and nonlinear data structures, ADT concept, Linear List ADT, Array Representation, Linked representation, Representation of single, two dimensional arrays, sparse matrices and their representation. Singly linked lists -insertion, deletion, search operations, doubly linked lists-insertion, deletion operations, and circular lists.

UNIT III

Stack and Queue ADTs, array and linked list representations, infix to postfix conversion using stack. Implementation of recursion, Circular queue-insertion and deletion, Dequeue ADT, array and linked list representations, Priority queue ADT, implementation using Heaps, Insertion into a Max Heap, Deletion from a Max Heap.

UNIT IV

Searching–Linear and binary search methods, Hashing-Hash functions, Collision Resolution methods-Open Addressing, Chaining Sorting –Bubble sort, Insertion sort, Quick sort, Merge sort, Heap sort comparison of sorting methods.

UNIT IV

Trees- Ordinary and Binary trees terminology, Properties of Binary trees, Binary tree ADT, representations, recursive and non-recursive traversals Search trees- Binary search tree-Binary search tree ADT, insertion, deletion and searching operations, Balanced search trees, AVL trees- Red Black trees .

Graphs- Graphs terminology, Graph ADT, representations, graph traversals/search methods- dfs and bfs,

Text Books:

1. Data Structures and Algorithms using C++, Ananda Rao Akepogu and Radhoka Raju Palagiri, Pearson Education.
2. Data structures, Algorithms and Applications in C++, S.Sahni, University Press (India) Pvt.Ltd, 2nd edition, Universities Press Orient Longman Pvt. Ltd.

Reference Books :

1. Data structures and Algorithms in C++, Michael T.Goodrich, R.Tamassia and .Mount, Wiley student edition, John Wiley and Sons.
2. Data Structures and Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education. Ltd., Second Edition.
3. Data Structures and algorithms in C++, 3rd Edition, Adam Drozdek, Thomson.
4. Data Structures using C and C++, Langsam, Augenstein and Tanenbaum, PHI.
5. Problem Solving with C++, The OOP, Fourth Edition , W. Savitch, Pearson education.
6. Data Structures using C++, D.S. Malik, Cengage Learning. India Edition.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

COMPUTER SYSTEM DESIGN
(14CSP11T03)

Course Objectives:

- To understand the design of the various functional units of digital computers.
- To understand the basic processing unit and how they are connected and how it generates control signals (using hardwired and micro programmed approaches).
- To be familiar with multithreading and the concepts of memory management including virtual memory.
- To understand the issues related to file system interface and implementation, disk management with protection and security mechanisms.

Course Outcomes:

After completion of the course the student will be able to

- Students will learn about computer performance, computer design, and trade-offs between cost and performance as well as between hardware and software.
- Students will formulate and solve problems, understand the performance requirements of systems.
- At the end of the course the students knows the need and requirement of an interface between Man and Machine.
- Students will be able to relate the features of operating systems and the fundamental theory associated with process, and file management's components of different operating systems.
- Students will learn about and understand theoretical concepts and programming constructs used for the operation of modern operating systems.

UNIT 1
Basic structure of computer:
Computer types, functional units, Basic Operational units, Bus software, software performance,Multiprocessors and multicomputers,Von-neumann architecture.

UNIT II
Computer structure – IA -32 Pentium: registers and addressing, input/Output organizaton, interrupts, DMA, Buses, Interface circuits, I/O interfaces, device drivers in windows, interrupt handlers.

UNIT III
Processing Unit: Execution of a complete instruction, multiple bus organization, hardwired control, micro programmed control.**Pipelining**: data hazards, instruction hazards, influence on instruction sets, data path & control consideration,RISC architecture introduction.

UNIT – IV
Processes and Threads: processes, threads, inter process communication, classical IPC problems, Deadlocks.

UNIT – V

File system: Files, directories, Implementation, Unix file system**Security:** Threats, intruders, accident data loss, basics of cryptography, user authentication.

Text Books:

1.Computer Organization – Car Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition,McGraw Hill.(UNIT I,II,III)

2. Modern Operating Systems, Andrew S Tanenbaum 2nd edition Pearson/PHI(UNIT IV,V)

Reference Books:

1. Computer Organization and Architecture – William Stallings Sixth Edition,pearson/PHI.
2. Morris Mano -Computer System Architecture –3rd Edition-Pearson Education .
3. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7thEdition, John Wiley.
4. Operating Systems – Internals and Design Principles Stallings, Fifth Edition–2005,Pearson Education/PHI.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

ADVANCE DATABASES
(14CSP11T04)

***Course Objectives*:**

- To make the student understand basic database concepts, including the structure of the relational data model.
- Construct simple and moderately advanced database queries using Structured Query Language (SQL).
- Describe and discuss selected advanced database topics, such as distributed database systems and the data warehouse.

Course Outcomes:

After completion of the course the student will be able to

- Student can able to write SQL Queries to retrieve the data from Database.
- Student can able to design the Databases.
- Student can effectively use Recovery Techniques.
- Student can able to write Fragment queries.

UNIT-I

Databases and Database Users - Database System Concepts and Architecture, Data Models, Schemas, and Instances, DBMS Architecture and Data Independence, Database Languages and Interfaces – Data Modelling Using the Entity-Relationship Approach, ER Model Concepts, Entity-Relationship (ER) Diagrams - Relational Model Concepts, Relational Integrity Constraints -Functional Dependencies and Normalization for Relational Databases, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form (BCNF).

UNIT-II

Relational Database Design Algorithms and Further Dependencies, Algorithms for Relational Database Schema Design, Multivalued Dependencies and Fourth Normal Form, Join Dependencies and Fifth Normal Form - Relational Algebra, Relational Operations - SQL - A Relational Database Language, The Relational Calculus, QUEL, Domain Relational Calculus, and QBE.

UNIT-III

Query Processing and Optimization: Basic Algorithms for Executing Query Operations, Using Heuristics in Query Optimization, Using Cost Estimates in Query Optimization, Semantic Query Optimization - Transactions Processing Concepts, Transaction and System Concepts, Desirable Properties of Transactions, Schedules and Recoverability, Serializability of Schedules.

UNIT-IV

Concurrency Control Techniques : Locking Techniques for Concurrency Control, Concurrency Control Techniques Based on Timestamp Ordering, Multiversion Concurrency Control Techniques, Optimistic Concurrency Control Techniques, Granularity of Data Items,

Recovery Techniques - Recovery Concepts, Recovery Techniques Based on Deferred Update, Recovery Techniques Based on Immediate Update, Shadow Paging, Database Security and Authorization - Introduction to Database Security Issues - Discretionary Access Control Using Privileges, Mandatory Access Control for Multilevel Security.

UNIT-V

Advanced Data Modeling Concepts : Enhanced-ER(EER) Model Concepts, EER-to-Relational Mapping, Data Abstraction and Knowledge Representation Concepts, Integrity Constraints in Data Modeling, Conceptual Design of Transactions, Overview of Other Data Models. Design – database fragments, allocation, Translation of global queries to fragment queries Optimization of Access strategies – Framework, Join queries, - Management of Distributed transactions – Framework – Supporting atomicity for distributed transactions, Concurrency control for distributed transactions, Architectural aspects.

Text Books:

1.Ramez Elmasri & Shamkant B. Navethe, Fundamentals of Database Systems, fourth Edition, Pearson Education, 2004. (Unit I,II,III,IV,V)

2.Abraham Silberchatz, Henry F. Korth, S.Sudarsan, *Database System Concepts*, Fifth Edition, McGraw-Hill, 2006. (Unit I, II)

3.Stefano Ceri, Giuseppe Pelagatti, *Distributed Databases Principles and Systems*, McGraw-Hill International Editions, 1985. (Unit V)

Reference Books:

1.Thomas M. Connolly, Carolyn E. Begg, Database Systems – A Practical Approach to Design, Implementation and Management, Third edition, Pearson Education, 2003.

2Jefrey D. Ullman, Jenifer Widom, A First Course in Database Systems, Pearson E ducation Asia, 2001.

3 .Stefano Ceri, Giuseppe Pelagatti, Distributed Databases Principles and Systems, McGraw-Hill International Editions, 1985.

4.Rajesh Narang, Object Oriented Interfaces and Databases, Prentice Hall of India, 2002.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

OBJECT ORIENTED ANALYSIS AND DESIGN
&
DESIGN PATTERNS
(14CSP11T05)

***Course Objectives*:**

- To have clear idea on the fundamentals of uml, principles of modeling.
- To learn on graphical notations & to be able to understand on all the building blocks of uml.
- To make students to know about how to design the class models and their modeling techniques.
- To learn on how to develop UML diagrams for an application.
- To learn concepts of design patterns and document editor.

Course Outcomes:

After completion of the course the student will be able to

- Students will be strong in the fundamentals of uml, principles of modeling.
- Students will analyse and design the application using UML diagrams.
- Students will get the ability to develop and deploy applications.
- They will get clear idea on design patterns and designing a document editor.

UNIT I

Introduction to UML: The meaning of Object Orientation, object identity, Encapsulation, information hiding, polymorphism, generosity, importance of modeling, principles of modeling, object oriented modeling,conceptual model of the UML, Architecture.

UNIT II

Basic Structural Modeling: Classes, Relationships, common Mechanisms, and diagrams.
Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams.

UNIT III

Interaction Diagrams, Common Modeling Techniques, Basic Behavioral Modeling: Use cases, Use case Diagrams, Activity Diagrams.Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams.
Architectural Modeling: Component, Deployment, Component diagrams and Deployment diagrams.

UNIT IV

Introduction To Design Patterns-Design Pattern Definition, Design Patterns in Small Talk MVC, Describing Design Patterns, Catalog of Design Patterns, Organizing the Catalog, Solving of Design Problems using Design Patterns, Selection of a Design Pattern, use of Design Patterns.

UNIT V

Designing A Document Editor: A Case Study -Design problems, Document structure, Formatting, Embellishing the User Interface, Supporting Multiple Look and Feel standards, Supporting Multiple Window Systems, User Operations, Spelling Checking and Hyphenation

Text Books :

1. The Unified Modeling Language User Guide By Grady Booch, James Rumbaugh, Ivar Jacobson 2nd Edition, Pearson Education.
2.Gamma, Belm, Johnson, "Design Patterns: Elements of Reusable Object Oriented Software",1995, PEA

Reference Books :

1. Fundamentals of Object Oriented Design in UML By Meilir Page-Jones, Pearson Education
2. Object Oriented Analysis & Design By Atul Kahate, The McGraw-Hill.
3. Fowler, Martin. UML Distilled. 3rd ed. Pearson Education. 2004.
4. Michael Blaha and James Rumbaugh, "Object-oriented modeling and design with UML", Prentice-Hall of India, 2005.
5. Booch, Grady. Object Oriented Analysis and Design. 2nd ed. Pearson Education. 2000.
6. Cooper, "Java Design Paterns", Pearson.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM(CSE)

L	T	C
4	0	4

SOFTWARE ENGINEERING
(14CSP11E1a)
(Elective-I)

Course Objectives:

- To help students to develop skills that will enable them to construct software of high quality – software that is reliable, and that is reasonably easy to understand, modify and maintain.
- To foster an understanding of why these skills are important.
- To help students to develop skills that will enable them to construct software of high quality for real world application.

Course Outcomes:

After completion of the course the student will be able to

- The ability to analyze, design, verify, validate, implement, apply, and maintain software systems.
- The ability to appropriately apply discrete mathematics, probability and statistics, and relevant topics in computer science and supporting disciplines to complex software systems.
- The ability to work in one or more significant application domains.
- The ability to manage the development of software systems.

UNIT I

Software, Software Engineering, and Process: The nature of Software, The unique nature of WebApps, Software engineering- A layered technology, The essence and principles of software engineering practice, Generic process model (framework), Process patterns, Process assessment and improvement, CMMI, Software myths. **Process Models:** Prescriptive process models: The waterfall model, Incremental process models, Evolutionary process models. The Unified process, Aspect oriented software development, Agile development: Agile process, Extreme programming.

UNIT II

Software Requirements: Introduction to functional and non-functional requirements, Requirements engineering activities, Eliciting requirements, *Requirements modeling*, Requirements validation, Software requirements specification (SRS), Requirements management. ***Requirements modeling*:** Structured view: Data modeling (ERD), Functional modeling (DFD), Behavioral modeling. Object oriented view: Use cases, CRC Modeling, Analysis classes, Collaborations, Responsibilities, Object relationship model, Object behavior model. **Software Project Estimation:** Empirical estimation models.

UNIT-III:

Design Concepts: Software design quality guidelines and attributes, Design concepts.

Software Architecture: Architecture and its importance, Architectural Styles, Data design, Architectural design. **Design :** Structured view (Traditional view): Architectural mapping using data flow (Call and return architecture), Interface design, Function based component design.
Object oriented view: OO Architecture, Class hierarchies, Message design, Class based component design. **Performing User Interface Design:** Golden rules, User interface analysis and design, interface analysis, interface design steps.

UNIT-IV:
Pattern Based Design: Design patterns, Pattern based software design, Architectural patterns, Component level design patterns, User interface design patterns. **Testing :** Software testing strategies: A strategic approach to software testing, Test strategies (Unit testing and integration testing) for conventional and object oriented software, Validation testing, System testing, The art of debugging.**Testing Conventional Applications:** Software testing fundamentals, White-Box testing: basis path testing, condition (predicate) testing, data flow testing, loop testing, Black box testing: Equivalence partitioning, Boundary value analysis, Graph based testing methods.

UNIT-V:
Testing Object Oriented Applications: OO testing methods, Testing methods applicable at class level, Interclass test case design. **Umbrella Activities :** Risk management, Software quality assurance, Software configuration management, Measurement and metrics: Size oriented metrics, Function oriented metrics, Metrics for software quality, Product metrics: Metrics for the requirements model, Metrics for the design model, Metrics for source code, Metrics for testing, Metrics for maintenance. **Software Reengineering:** A software reengineering process model, Software reengineering activities.

Text Books:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 7th edition. McGrawHill International Edition. (UNIT-I, III, V)
2. Software Engineering- Sommerville , 7th edition, Pearson education.(UNIT-II,IV)

Reference Books
1. Software Engineering- K.K. Agarwal & Yogesh Singh,New Age International Publishers
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiely.
3. Systems Analysis and Design- Shely Cashman Rosenblatt,Thomson Publications.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

ARTIFICIAL INTELLIGENCE
(14CSP11TE1b)
(Elective-I)

***Course Objectives*:**

- To familiarize students with Artificial Intelligence techniques for building well-engineered and efficient intelligent systems.
- To familiarize students with Pattern-directed inference systems and different types of truth maintenance systems will be discussed in length from both theoretical and applied point of view.
- Some cutting edge applications of these AI systems will also be discussed to make student implement in real world.

Course Outcomes:

- After completion of the course the student will be able to Understand different types of AI agents.
- Know various AI search algorithms (uninformed, informed, heuristic, constraint satisfaction, genetic algorithms).
- Understand the fundamentals of knowledge representation (logic-based, frame-based, semantic nets), inference and theorem proving .
- Know how to build simple knowledge-based systems.
- Demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
- Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.
- Ability to carry out independent (or in a small group) research and communicate it effectively in a seminar setting.

UNIT I

Introduction – Agents – Problem formulation – uninformed search strategies – heuristics– informed search strategies – constraint satisfaction.

UNIT II

Logical agents – propositional logic – inferences – first-order logic – inferences in first order logic – forward chaining – backward chaining – unification – resolution.

UNIT III

Planning with state-space search – partial-order planning – planning graphs – planning and acting in the real world.

UNIT IV

Uncertainty – review of probability - probabilistic Reasoning – Bayesian networks – inferences in Bayesian networks – Temporal models – Hidden Markov models.

UNIT V

Learning from observation - Inductive learning – Decision trees – Explanation based learning – Statistical Learning methods - Reinforcement Learning.

Text Books:

1. S. Russel and P. Norvig, “Artificial Intelligence – A Modern Approach”, Second Edition, Pearson Education, 2003.(UNIT -I,II,III,IV.V)

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel, ”Computational Intelligence : a logicalapproach”, Oxford University Press, 2004.
2. G. Luger, “Artificial Intelligence: Structures and Strategies for complex problem solving”, Fourth Edition, Pearson Education, 2002.
3.]. Nilsson, “Artificial Intelligence: A new Synthesis”, Elsevier Publishers, 1998.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

FORMAL LANGUAGES & AUTOMATA THEORY
(14CSP11E1c)
(Elective-I)

Course Objectives:

- To make student learn the fundamental concepts of formal languages and automata.
- To make student to construct regular expressions, grammars, and automata for different levels of formal languages.
- To make student understand the relationship between Turing machines and modern computers.
- To make student understand the limitations and undecidable problems of modern computers.

Course Outcomes:

After completion of the course the student will be able to

- Understand different types of AI agents.
- Know various AI search algorithms (uninformed, informed, heuristic,constraint satisfaction , genetic algorithms).
- Understand the fundamentals of knowledge representation (logic-based, frame-based, semantic nets), inference and theorem proving.
- Know how to build simple knowledge-based systems.
- Demonstrate working knowledge of reasoning in the presence of incomplete and/or uncertain information.
- Ability to apply knowledge representation, reasoning, and machine learning techniques to real-world problems.
- Ability to carry out independent (or in a small group) research and communicate it effectively in a seminar setting.

UNIT I

Fundamentals : Strings, Alphabet, Language, Operations, Finite state machine, definitions, finite automaton model, acceptance of strings, and languages, deterministic finite automaton and non-deterministic finite automaton, transition diagrams and Language recognizers.

Finite Automata: NFA with Î transitions - Significance, acceptance of languages. Conversions and Equivalence: Equivalence between NFA with and without Î transitions, NFA to DFA conversion, minimization of FSM, equivalence between two FSM's, Finite Automata with output- Moore and Melay machines.

UNIT II

Regular Languages : Regular sets, regular expressions, identity rules, Constructing finite **Automata for a given regular expressions, Conversion of Finite Automata to Regular expressions. Pumping lemma of regular sets, closure properties of regular sets (proofs not**required).

Grammar Formalism : Regular grammars-right linear and left linear grammars, equivalence between regular linear grammar and FA, inter conversion, Context free grammar, derivation trees, sentential forms. Right most and leftmost derivation of strings.

UNIT III

Context Free Grammars : Ambiguity in context free grammars. Minimisation of Context Free Grammars. Chomsky normal form, Greiback normal form, Pumping Lemma for Context Free Languages. Enumeration of properties of CFL (proofs omitted).

Push Down Automata : Push down automata, definition, model, acceptance of CFL, Acceptance by final state and acceptance by empty state and its equivalence. Equivalence of CFL and PDA, interconversion. (Proofs not required). Introduction to DCFL and DPDA.

UNIT IV

Turing Machine : Turing Machine, definition, model, design of TM, Computable functions, recursively enumerable languages. Church's hypothesis, counter machine, types of Turing machines (proofs not required).

UNIT V

Computability Theory : Chomsky hierarchy of languages, linear bounded automata and context sensitive language, LR(0) grammar, decidability of, problems, Universal Turing Machine, undecidability of posts. Correspondence problem, Turing reducibility, Definition of P and NP problems, NP complete and NP hard problems.

Text Books:

1."Introduction to Automata Theory Languages and Computation". Hopcroft H.E. and Ullman J. D. Pearson Education. (UNIT-I, II, III)

2. Introduction to Theory of Computation –Sipser 2nd editionThomson(UNIT-IV, V)

Reference Books

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

2Introduction to languages and the Theory of Computation ,John C Martin, T

3."Elements of Theory of Computation", Lewis H.P. & Papadimition CPearson /PHI.

4 .Theory of Computer Science – Automata languages and computation -Mishra and Chandrashekaran, 2nd edition, PHI.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

GRID COMPUTING
(14CSP11E1d)
(Elective-I)

Course Objectives:

- To understand the genesis of grid computing.
- To understand the application of grid *computing.*

Course Outcomes:

After completion of the course the student will be able to

- Students will understand the concepts of Grid Computing.
- Students will know about the applications and technologies of grid computing.

UNIT 1
INTRODUCTION AND OVERVIEW OF GRID COMPUTING

Early Grid Activities-Current Grid Activities-An Overview of Grid Business Areas-Grid Applications-Grid Infrastructure.

UNIT II
WEB SERVICES AND RELATED TECHNOLOGIES

Service – Oriented Architecture-Web Service Architecture-XML, Related Technologies, and Their Relevance to Web services-XML Messages and Enveloping-Service Message Description Mechanisms-Relationship between Web Service and Grid Service – Web Service Interoperability and the Role of the WS-I Organization.

UNIT III
DISTRIBUTED OBJECT TECHNOLOGY FOR GRID COMPUTING (OGSA)

Introduction to Open Grid Services Architecture (OGSA)- Commercial Data Center- National Fusion Collaboratory- The OGSA Platform Components .

UNIT IV
OPEN GRID SERVICES INFRASTRUCTURE (OGSI)

Introduction-Grid Services-A High-Level Introduction to OGSI – Introduction to Service Data Concepts – Grid Service: Naming and Change Management Recommendations.

UNIT V
OGSA BASIC SERVICES AND THE GRID COMPUTING TOOLKITS

Common Management Model(CMM)-Security Architecture- GLOBUS GT3 Toolkit: Architecture- GLOBUS GT3 Toolkit: - Architecture, Programming model, High level services .

Text Books

1. Joshy Joseph & Craig Fellenstein, “Grid Computing”, Pearson/PHI PTR-2003.

Reference Books

1. Ahmar Abbas, “Grid Computing: A Practical Guide to technology and Applications”, Charles River media – 2003.

2. Fran Berman , Geoffrey Fox, Anthony J.G. Hey, Grid Computing: Making The Global Infrastructure a Reality,Wiley, 2003
3. Maozhen Li , Mark Baker , The Grid: Core Technologies, Wiley, 2005

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)	**(14CSP11P01)**	**L**	**T**	**C**
		4	**0**	**4**

Course Objectives:

- To develop skills to design and analyze the problems.
- To inculcate the programming skills of python language.
- To make the students to feel comfort with the object oriented programming.

Course Outcomes:

After completion of the course the student will be able to

- Students can learn python language constructs.
- Students will learn OOP concepts which will be helpful for them in further semesters.
- Students will have basic idea on data structures.

LIST OF EXPERIMENTS

Week 1

a) Implement Python script to read person's age from keyboard and display whether he is eligible for voting or not.
b) Implement Python script to find biggest number between two numbers.

Week 2

a) Implement Python Script to generate prime numbers series up to n.
b) Implement Python Script to check given number is palindrome or not.
c) Implement Python script to print factorial of a number.

Week 3

a) Implement Python Script to perform various operations on string using string libraries.
b) Implement Python Script to check given string is palindrome or not.

Week 4

a) Define a function max_of_three() that takes three numbers as arguments and returns the largest of them.
b) Write a program which makes use of function to display all such numbers which are divisible by 7 but are not a multiple of 5, between 1000 and 2000.

Week 5

a) Define a function which generates Fibonacci series up to n numbers.
b) Define a function that checks whether the given number is Armstrong.

Week 6

a) Write a program which accepts a sequence of comma-separated numbers from console and generate a list and a tuple which contains every number.

Suppose the following input is supplied to the program:34,67,55,33,12,98. Then, the output should be: ['34', '67', '55', '33', '12', '98'] ('34', '67', '55', '33', '12', '98').

b) With a given tuple (1,2,3,4,5,6,7,8,9,10), write a program to print the first half values in one line and the last half values in one line.

Week 7

a) Write a python script to perform basic dictionary operations like insert, delete and display.

b) Write a python script to find frequency of words in a file using dictionaries.

Week 8

a) Write Python script to display file contents.

b) Write Python script to copy file contents from one file to another.

Week 9

a) Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

b) Define a class named Circle which can constructed by radius. The derived classes Area, Circumference uses methods called calArea(), calCirc() respectively to calculate area, circumference of circle.

Week 10

a) Implement Python script to develop stack ADT and its operations.

b) Implement Python script to evaluate postfix expression.

Week 11

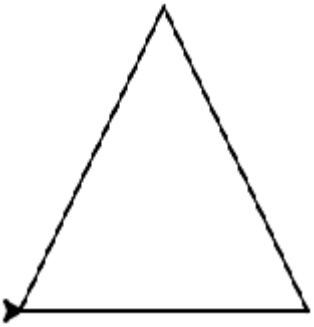
a) Implement Python script to develop queue ADT and its operations.

b) Implement Python script to perform tree traversals.

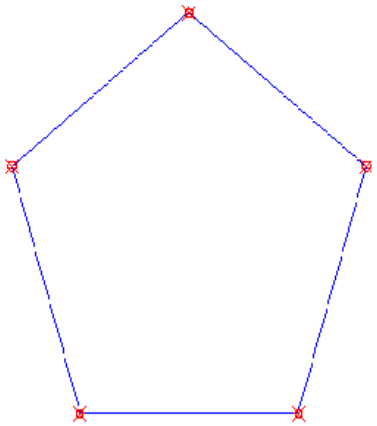
Week 12

Write a python script to display following shapes using turtle.

a)



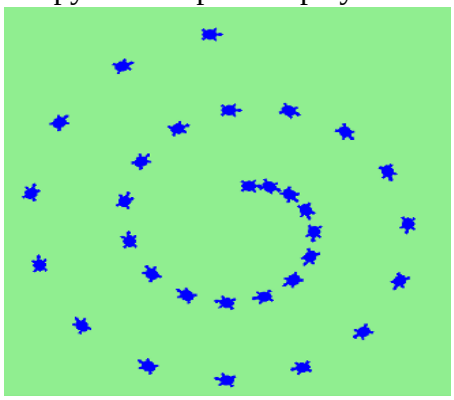
b)



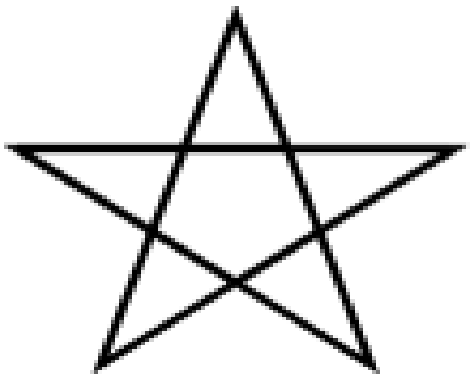
Week 13

Write a python script to display following shapes using turtle.

a)



b)



Advanced Data Structure Lab

1. Linked lists
2. Multistacks
3. Double Ended Queue (Deques) & Circular Queues
4. Min Heap
5. Deaps
6. Leftist Heap
7. AVL Tree
8. B:Tree
9. Quick Sort
10. Greedy algorithm
11. Knapsack using Dynamic Programming
12. Graph coloring using backtracking

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I SEM (CSE)

L	T	C
4	0	4

DATA WAREHOUSING AND DATAMINING
(14CSP12T06)

***Course Objectives*:**

- To differentiate OnLine Transaction Processing and OnLine Analytical processing
- To learn Multidimensional schemas suitable for data warehousing
- To understand various data mining functionalities
- To inculcate knowledge on data mining query languages.
- To know in detail about data mining algorithms
-

Course Outcomes:

After completion of the course the student will be able to

- Ability to apply Design a data mart or data warehouse for any organization
- Ability to apply Develop skills to write queries using DMQL
- Able to extract knowledge using data mining techniques
- Ability to adapt to new data mining tools.
- Ability to explore recent trends in data mining such as web mining, spatial-temporal mining.

UNIT I

Introduction : Data Mining, Kinds of Data, Data Mining Functionalities, Classification of Data Mining Systems, Primitives, Major Issues in Data Mining. **Data Preprocessing**: Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation.

UNIT II

Data Warehouse and OLAP Technology: What is Data Warehouse, A Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, From Data Warehouse to Data Mining. **Mining Frequent Patterns and Associations:** Basic Concepts, Efficient and Scalable Frequent Itemset Mining Methods, Mining Various Kinds of Association Rules.

UNIT III

Classification and Prediction: Issues regarding classification and prediction, classification by decision tree induction, Bayesian classification, Rule based classification, Prediction, Accuracy and Error Measures. **Cluster Analysis:** Types of Data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density based Methods, Grid based methods, model based clustering methods, Clustering high dimensional data, Outlier analysis.

UNIT IV

Mining Stream, Time-Series, and Sequence Data: Mining Data Streams, Mining Time-Series Data, Mining Sequence Patterns in Biological Data.

UNIT V

Applications and Trends in Data Mining: Data Mining Applications, : Data Mining for Financial Data Analysis, Data Mining for the Retail Industry, Data Mining for the Telecommunication Industry, Data Mining for Biological Data Analysis, Data Mining in Other Scientific Applications, Data Mining for Intrusion Detection, Social Impacts of Data Mining.

Text Books:

1.Jiawei Han and Micheline Kamber, Data Mining, Concepts and Techniques, Elsevier, II Edition, 2008.

(UNIT -I,II,III,IV.V)

Reference Books:

1.Margaret H Dunham, Data Mining Introductory and Advanced Topics, Pearson Education, 2e, 2006.

2.Amitesh Sinha, Data Warehousing, Thomson Learning, 2007.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - IISEM (CSE)

L	T	C
4	0	4

JAVA AND WEB TECHNOLOGIES
(14CSP12T07)

Course Objectives:

- To make the students better understanding of web concepts.
- To learn about Java and Applets.
- To know about Swing components and Multimedia programming.
- To study about the need and importance of Java Beans.
- To learn about Servelts and JSP.

Course Outcomes:

After completion of the course the student will be able to

- Students will get better understanding of web concepts and to design dynamic webpages.
- To implement the concepts in Java and also using Applets.
- To make use of Swing concepts in programming.
- To deploy Java Beans for applications.
- Students will get ideas to implement programs using Servelts and JSP.

UNIT I
Introduction – Network Concepts – Web Concepts – Internet Addresses – Retrieving Data with URL – HTML – DHTML Cascading Style Sheets – Scripting Languages Javascript – VBscript.

UNIT II
Java Fundamentals Classes – Inheritance – Packages – Interfaces – Exceptions Handling – Multithreading – Applets.

UNIT III
Introduction of Swing, Swing Components, Look and Feel for Swing Components, Introduction to Multimedia Programming.

UNIT IV
Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound Properties, Bean Info Interface, Constrained properties Persistence, Customizes, Java Beans API.

UNIT V
Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading initialization parameters. Introduction to JSP: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC architecture.

Text Books:

1. Deitel, Deitel and Neito, "Internet and World Wide Web, How to program", Pearson education Asia, 2001.
2. Java 2 Unleashed (Techmedia – SAMS) By Jamie Jaworski.
3. Web Programming, building internet applications, Chris Bates 2nd edition, WILEY Dreamtech
4. The complete Reference BooksJava 2 Fifth Edition ,Patrick Naughton and Herbert Schildt., TMH 5.Java Server Pages –Hans Bergsten, SPD O'Reilly (UNITs 3,4,5).

Reference Books:

1. Programming world wide web-Sebesta, Pearson Core SERVLETS ANDJAVASERVER. PAGES, VOLUME 1: CORE TECHNOLOGIES , Marty Hall and Larry Brown Pearson
2. Professional Java Server Programming (a Press) By Allamaraju.
3. Jaison Hunder & William Crawford, Java Servlet Programming, O'REILLY, 2002.
4. David Flanagan,Jim Parley, William Crawford & Kris Magnusson , Java Enterprise in a nutshell- A desktop Quick Reference Books– O'REILLY, 2003 .

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - IISEM (CSE)

L	T	C
4	0	4

MOBILE COMPUTING
(14CSP12T08)

Course Objectives:

- Introduction of an advanced element of learning in the field of wireless communication.
- Introduces the basic concepts and principles in mobile computing.
- Expose the students to the concepts of wireless devices and mobile computing.

Course Outcomes*:*

After completion of the course the student will be able to

- Students to understand the concept of mobile computing and architecture of mobile communication.
- Apply the concepts of mobile computing on conventional wired networks .

UNIT I
Introduction to Mobile Communication and Computing: Novel applications – A short history of wireless communication-A maket for mobile communications.
GSM: Mobile services, System architecture, Radio Interface, Protocols, Localization and calling, Handover, Security, New data Services

UNIT II
(Wireless) Medium Access Control: Motivation for a specialized MAC (Hidden and exposed terminals, Near and far terminals), SDMA, FDMA, TDMA, CDMA.

UNIT III
Mobile Network Layer: Mobile IP (Goals, assumptions, entities and terminology, IP packet delivery, agent advertisement and discovery, tunnelling and encapsulation), Dynamic Host Configuration Protocol (DHCP), Properties and Applications of a MANETs - Routing – DSDV – DSR.

UNIT IV
Mobile Transport and Application Layer: Traditional TCP - Indirect TCP - Snooping TCP - Mobile TCP – Fast retransmit/Fast recovery - Transmission/time-out freezing - Selective retransmission – Transaction Oriented TCP – Wireless Application Protocol.

UNIT-V
Protocols and Tools: Bluetooth(user scenarios-architecture-Radiolayer-Baseband layer-Link manager protocol-L2CAP-Security-SDA-Profiles). Wireless application protocol(architecture-wieless datagram protocol-wireless transport layer security-Wireless session protocol-wireless application environment-wireless markup language).

Text Books:

1. Jochen Schiller, "Mobile Communications", *Addison-Wesley*. (Chapters I,II,III,IV,V),second edition, 2004.
2. Stojmenovic and Cacute, "Handbook of Wireless Networks and Mobile Computing", *Wiley*, 2002.

Reference Books:

1. Reza Behravanfar, "Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML", ISBN: 0521817331, Cambridge University Press, October 2004.
2. Adelstein, Frank, Gupta, Sandeep KS, Richard III, Golden, Schwiebert, Loren, "Fundamentals of Mobile and Pervasive Computing", ISBN: 0071412379, McGraw-Hill Professional, 2005.
3. Hansmann, Merk, Nicklous, Stober, "Principles of Mobile Computing", *Springer*, second edition, 2003.
4. MartynMallick, "Mobile and Wireless Design Essentials", Wiley *DreamTech*, 2003.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

L	T	C
4	0	4

ADVANCED COMPUTER NETWORKS
(14CSP12T09)

Course Objectives:

- To make the students recall the basics of computer networks and to discuss different computer networking protocols.
- To refresh the knowledge of networking devices and learn data link layer and LAN protocols in detail.
- To make the students learn the core concepts of cellular wireless networks.
- To introduce the concepts of optical networks and protocols in optical networks.
- To make the students learn what is wireless sensor networks and different WSN protocols

Course Outcomes*:*

After completion of the course the student will be able to

- Students will refresh their knowledge in basics of computer networks and TCP/IP and OSI layers.
- Students will learn different link layer protocols and LAN protocols.
- Students will learn channel assignments, handoffs, allocation, channel interference and power control.
- They will be able to study what are optical networks and use of MPLS protocol in optical networking, and learn about optical switches.
- They will learn about Wireless Sensor Networks (WSN), and WSN energy efficient routing protocols.

UNIT I

Review of Computer Networks and Networking devices: What is the Internet, The Network edge, The Network core, Access Networks and Physical media, ISPs and Internet Backbones, Delay and Loss in Packet-Switched Networks, History of Computer Networking and the Internet - **Foundation of Networking Protocols:** 5-layer TCP/IP Model, 7-Layer OSI Model, Internet Protocols and Addressing, Equal-Sized Packets Model: ATM. **Networking Devices:** Multiplexers, Modems and Internet Access Devices, Switching and Routing Devices, Router Structure.

UNIT II

Routing and Internetworking: Network–Layer Routing, Least-Cost-Path algorithms, Non-Least-Cost-Path algorithms, Intradomain Routing Protocols, Interdomain Routing Protocols, Congestion Control at Network Layer. **Logical Addressing:** IPv4 Addresses, IPv6 Addresses - **Internet Protocol:** Internetworking, IPv4, IPv6, Transition from IPv4 to IPv6 – **Multicasting Techniques and Protocols:** Basic Definitions and Techniques, Intradomain Multicast Protocols, Interdomain Multicast Protocols, Node-Level Multicast algorithms

.

UNIT III

Transport and Application layer protocols: Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, TCP Congestion Control **Application Layer:** Principles of Network Applications, The Web and

HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name System (DNS), P2P File Sharing, Socket Programming with TCP and UDP, Building a Simple Web Server.

UNIT IV

Wireless Networks-Mobile IP and optical networks: Infrastructure of Wireless Networks, Wireless LAN Technologies, IEEE 802.11 Wireless Standard, Cellular Networks, Mobile IP, Wireless Mesh Networks (WMNs).**Optical Networks and WDM Systems:** Overview of Optical Networks, Basic Optical Networking Devices, Large-Scale Optical Switches, Optical Routers, Wavelength Allocation in Networks, Case Study: An All-Optical Switch.

UNIT V

VPNs, Overlay Networks and Sensor Networks: Virtual Private Networks (VPNs), Multiprotocol Label Switching (MPLS), Overlay Networks – **VoIP and Multimedia Networking:** Overview of IP Telephony, VoIP Signaling Protocols, Real-Time Media Transport Protocols, Distributed Multimedia Networking, Stream Control Transmission Protocol.**Mobile A-Hoc Networks:** Overview of Wireless Ad-Hoc Networks, Routing in Ad-Hoc Networks, Routing Protocols for Ad-Hoc Networks – **Wireless Sensor Networks:** Sensor Networks and Protocol Structures, Communication Energy Model, Clustering Protocols, Routing Protocols.

Text Books:

1.Computer Networking: A Top-Down Approach Featuring the Internet, *James F. Kurose*, *Keith W.Ross*, Third Edition, Pearson Education, 2007. (UNITS-I, II, III).

2.Computer and Communication Networks, *Nader F. Mir*, Pearson Education, 2007. (UNIT-IV, V).

Reference Books:

1.Data Communications and Networking, Behrouz A. Forouzan, Fourth Edition, Tata McgrHill,2007.

2.Guide to Networking Essentials, Greg Tomsho,Ed Tittel, David Johnson,Fifth Edition, Thomson.

3.An Engineering approach to computer Networking, K. Keshav, Pearson Education.

4.Campus Network Design Fundamentals, Diane Teare, Catherine paquet, Pearson Education (CISCO Press).

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

L	T	C
4	0	4

SOFTWARE QUALITY ASSURANCE AND TESTING
(14CSP12T10)

Course Objectives:

- Why do software testing
- The meaning of black-box testing and white-box testing; Software Testing throughout the Software Process; Software Testing and Extreme Programming.
- The Automation of Software Testing; Difficulties and Limitations of Software Testing
- The Business of Software Testing; Implementing and Automated Testing.

Course Outcomes:

After completion of the course the student will be able to

- Understand the effectively strategies of testing, the methods and technologies of software testing
- Design test plan and test cases.
- Do automatic testing.
- Establish a testing group and manage the whole testing project.
- Clearly and correctly report the software defectives.
- Distinguish relationship between the software testing.

UNIT I

Software Quality Assurance Framework and Standards SQA Framework: What is Quality? Software Quality Assurance, Components of Software Quality Assurance – **Software Quality Assurance Plan:** Steps to develop and implement a Software Quality Assurance Plan – **Quality Standards:** ISO 9000 and Companion ISO Standards, , Malcom Balridge, 3 Sigma, 6 Sigma .

UNIT II

Software Quality Assurance Metrics and Measurement Software Quality Metrics: Product Quality metrics, In-Process Quality Metrics, Metrics for Software Maintenance, Examples of Metric Programs.

UNIT III

Software Quality metrics methodology: Establish quality requirements, Identify Software quality metrics, Implement the software quality metrics, analyze software metrics results, validate the software quality metrics – **Software quality indicators – Fundamentals in Measurement theory .**

UNIT IV

Software Testing Techniques Black-Box, Boundary value, Bottom-up, Branch coverage, Cause-Effect graphing, CRUD, Database, Exception, Gray-Box, Histograms, Inspections, JADs, Pareto Analysis, Prototyping, Random Testing, Risk-based Testing, Regression Testing, Structured Walkthroughs, Thread Testing, Performance Testing, White-Box Testing .

UNIT VSoftware Testing Tools Taxonomy of Testing tools, Methodology to evaluate automated testing tools, Load Runner, Win runner and Rational Testing Tools, Java Testing Tools, JMetra, JUNIT and Cactus.

Text Books:

1. Effective Methods for Software Testing, 2nd Edition, William E. Perry , Second Edition,
   Wiley India, 2006.
2. Software Quality, Mordechai Ben-Menachem/Garry S. Marliss, Thomson Learning publication,1997.

Reference books:

1. Testing and Quality Assurance for Component-based Software, by Gao, Tsao and Wu, Artech House Publishers
2. Software Testing Techniques, by Bories Beizer, Second Edition, Dreamtech Press
3. Managing the Testing Process, by Rex Black, Wiley
4. Handbook of Software Quality Assurance, by G. Gordon Schulmeyer, James I.McManus, Second Edition, International Thomson Computer Press
5. Software Testing and continuous Quality Improvement, by William E.Lewis, Gunasekaran Veerapillai, Second Edition, Auerbach Publications
6. Metrics and Models for Software Quality Engineering, by Stephen H. Kan, by Pearson Education Publication
7. Software Testing Tools, K.V.K.K. Prasad, Dream tech press, 2008.
8. Practical Software Testing, Ilene Burnstein, Springer, 2003.
9 Software Testing, Srinivasan Desikan & Gopalaswamy Ramesh, Pearson Education,2006.
10. Software testing techniques, Scott Loveland & Geoffrey Miller, Shroff Publishers, 2005.
11. Software Quality, Martin Wieczorek & Dirk Meyerhoff, Springer, 2001.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

L	T	C
4	0	4

DISTRIBUTED SYSTEMS
(14CSP12E2a)
(Elective II)

Course Objectives:

- To explore distributed systems principles asociated with communication, naming.
- To explore synchronization, distributed file systems, system design, distributed scheduling, and several case studies.
- To cover both foundational concepts and as well as practical deployments.

Course Outcomes:

After completion of the course the student will be able to

- How distributed systems may be constructed using a variety of tools and approaches.
- Students will be able to design, and implement distributed software systems in Java using sockets remote procedure cal l mechanisms and java RMI.

UNIT I

Characterization of Distributed Systems-Introduction-Examples-Resource Sharing and the Web-Challenges.System Models-Architectural-Fundamental.Interprocess Communication-Introduction-API for Internet protocols-External data representation and marshalling--Client-server communication-Group communication-Case study: Interprocess Communication in UNIX.

UNIT II

Distributed Objects and Remote Invocation-Introduction-Communication between distributed objects-Remote procedure calls-Events and notifications-Case study: JavaRMI. Operating System Support-Introduction-OS layer-Protection-Processes and threads-Communication and invocation OS architecture.

UNIT III

Distributed File Systems-Introduction-File service architecture-Case Study:Sun Network File System-Enhancements and further developments. Name Services-Introduction-Name Services and the Domain Name System-DirectoryServices-Case Study: Global Name Service.

UNIT IV

Time and Global States-Introduction-Clocks, events and process states-Synchronizing physical clocks-Logical time and logical clocks-Global states-Distributed debugging. Coordination and Agreement-Introduction-Distributed mutual exclusion-Elections-Multicast communication-Consensus and related problems.

UNIT V

Distributed Shared Memory-Introduction-Design and implementation issues-Sequential consistency and Ivy case study Release consistency and Munin case study-Other consistency models.CORBA Case Study- Introduction-CORBA RMI-CORBA services.

Text Books

1. George Coulouris, Jean Dollimore, Tim Kindberg, , "Distributed Systems: Concepts and Design", 4th Edition, Pearson Education, 2005.(UNIT I,II,III,IV,V).

Reference Books

1. A.tS. Tanenbaum and M. V. Steen, "Distributed Systems: Principles and Paradigms", Second Edition, Prentice Hall, 2006.
2. M.L.Liu, “Distributed Computing Principles and Applications”, Pearson Addison Wesley, 2004.
3. Mukesh Singhal, “Advanced Concepts In Operating Systems”, McGrawHill Series in Computer Science, 1994.
4. Nancy A. Lynch, "Distributed Algorithms", The Morgan Kaufmann Series in Data Management System, Morgan Kaufmann Publishers, 2000.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

L	T	C
4	0	4

CLOUD COMPUTING

(14CSP12E2b)
(Elective II)

Course Objectives:

- The objective focuses on technologies specific to the networked, distributed dimension of software and access to services and data.
- It will support long-term research on new principles, methods, tools and techniques enabling software developers in the EU to easily create interoperable services based on open standards, with sufficient flexibility and at a reasonable cost

Course outcomes*:*

After completion of the course the student will be able to

- Intelligent and autonomic management of cloud resources, ensuring agile elastic scalability. Scalable data management strategies, addressing the issues of heterogeneity, consistency, availability, privacy and supporting security.
- Technologies for infrastructure virtualization, cross platforms execution as needed for service composition across multiple, heterogeneous environments, autonomous.
- Management of hardware and software resources.
- Interoperability amongst different clouds, portability, protection of data in cloud environments, control of data distribution and latency.Seamless support of mobile, context-aware applications.

UNIT 1

Examining the Value Proposition: Defining Cloud Computing [CC for short]:

Definition; Cloud types; characteristics of CC; Role of open standards. Assessing the Value Proposition: Measuring the could value, Avoiding capital expenditure, Total cost of ownership, Specifying SLAs

UNIT 2

Architecture and types: Cloud Architecture: CC stack; Connecting to the cloud.

Services and Applications by type: IaaS, PaaS, SasS, IDasS, CaaS.

UNIT 3

Platforms: Abstractions and Virtualization: Using virtualization technologies; Load balancing and virtualization, Hypervisor, Machine image, porting applications.

Capacity planning: defining baseline and metrics; Network capacity.

Platform as a Service: Defining service;, PaaS application frameworks.

UNIT 4

Cloud computing with Titans: Google and Amazon web services, Microsoft cloud services.

UNIT 5

Exploring Cloud Infrastructures: Managing the cloud: Administrating the cloud; Cloud management products; Emerging cloud management standards.

Cloud Security: Securing the cloud; securing data; Establishing identity and presence.

Services & Applications: Service Oriented Architecture: Introduction; Defining SOA communications; Managing and monitoring SOA; Relating SOA and CC. Moving applications to cloud. Cloud based storage. Webmail services.

Text Books:

1. Cloud Computing – Bible, Barrie Sosinsky, Wiley-India edn.,2012
2. Cloud Computing – A Practical Approach, Anthony T. Velte, Toby J. Velte and Robert Elsenpeter, Tata McGraw-Hill edn. 2011.
3.

Reference Books:

1. Kai Hwang, Geoffrey C Fox, Jack G Dongarra, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2012.
2. John W.Rittinghouse and James F.Ransome, "Cloud Computing: Implementation, Management, and Security", CRC Press, 2010.
3. George Reese, "Cloud Application Architectures: Building Applications and Infrastructure in the Cloud" O'Reilly, 2009.
4 James E. Smith, Ravi Nair, "Virtual Machines: Versatile Platforms for Systems and Processes", Elsevier/Morgan Kaufmann, 2005.
5. Katarina Stanoevska-Slabeva, Thomas Wozniak, Santi Ristol, "Grid and Cloud ComputinA Business Perspective on Technology and Applications", Springer, 2010.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - I I SEM (CSE)

L	T	C
4	0	4

IMAGE PROCESSING
(14CSP12E2c)
(Elective II)

Course Objectives:

- To learn the fundamentals of Image Processing.
- To learn sampling and reconstruction procedures.
- To learn the various transforms used in image Processing.
- To study various concepts of image enhancement, reconstruction and image compression.
- To design image processing systems

Course *Outcomes:*

After completion of the course the student will be able to

- Develops ability to identify, formulate &solve problems involving images.
- Develops ability to design &conduct experiments, analyze &interpret image data.
- To design a software, Component or process as per needs &specifications.
- It will demonstrate the skills to use modern engineering tools, software's &equipment to analyze problems.
- Develop confidence for self-education & ability for life-long.

UNIT I
DIGITAL IMAGE FUNDAMENTALS
Introduction, Image sensing & acquisition, Concept of gray levels. Gray level to binary image conversion. Sampling and quantization. Relationship between pixels. Imaging Geometry, operations on digital image: array Vs matrix, linear Vs non-linear, arithmetic operations, set and logical operations, spatial operations, vector and matrix operations, probabilistic methods.

UNIT II
IMAGE TRANSFORMS2-D FFT, Properties. Walsh transform, Hadamard Transform, Discrete cosine Transform, Haar transform, wavelet transform, Hotelling transform, comparison of different image transforms.

UNIT III
IMAGE ENHANCEMENT
Enhancement in Spatial Domain: Point processing. Histogram processing. Spatial filtering(Smoothing and sharpening), Enhancement in frequency domain: Basics of filtering in frequency domain, Image smoothing, Image sharpening, Homomorphic filtering, basics of colour image processing.

UNIT IV
IMAGE RESTORATION
Noise models, Degradation model, Restoration in the presence of noise only, Spatial filtering, Inverse filtering, Least mean square filters, Constrained Least Squares Restoration, Interactive Restoration.

UNIT V
IMAGE SEGMENTATION
Introduction, Detection of discontinuities. Edge linking and boundary detection, Thresholding, Region based segmentation. Use of motion in segmentation
.

Text Books :

1. Digital Image processing – R.C. Gonzalez & R.E. Woods, Addison Wesley/ Pearson education,2nd Education, 2002.
2. Digital image processing by S.Jayaraman, S.Esakkirajan & T.Veera Kumar, Tata McGraw Hill, 2010.

Reference Books :

1.Fundamentals of Digital Image processing – A.K.Jain , PHI.
2.Digital Image processing using MAT LAB – Rafael C. Gonzalez, Richard E Woods and Steven 3.L. Edition, PEA, 2004.
3.Digital Image Processing – William K. Pratt, John Wilely, 3rd Edition, 2004.
4.Fundamentals of Electronic Image Processing – Weeks Jr., SPIC/IEEE Series, PHI.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

L	T	C
4	0	4

COMPILER CONSTRUCTION
(14CSP12E2d)
(Elective-II)

Course Objectives:

- Thorough understanding of the overall architecture of a modern compiler.
- Being familiar with both top-down and bottom-up parsing paradigms.
- Fluent with syntax-directed translation scheme and different compiler-compilers.
- Knowledgeable with assembly language and code-block based code generation scheme.
- Knowing the inner details of compilers, libraries, operating systems/platforms, and how they interact with each other to form modern computing environments.

Course Outcomes:

After completion of the course the student will be able to

- Generally applying the top down ad bottom approaches of parsing, and the lexical analysis.
- Student will be able to generate code generation and optimization phases of compilation, and design a compiler for a concise programming language.

UNIT-I

Introduction: Language processors, structure of a compiler: Lexical analysis, syntax analysis, semantic analysis, intermediate code generation, code optimization, code generation, symbol table management, the grouping of phases into passes. Definition of Assembler, pre-processor, compiler, Natural language processor and interpreter.

UNIT-II

Syntax analysis- introduction: the role of the parser. CFG: definition, notation, derivation, parse tree, ambiguity. Writing a grammar: Eliminating ambiguity, elimination of left recursion, left factoring Top down Parsing: recursive descent Parsing, first and follow, LL (1) grammar.

UNIT-III

Bottom up Parsing: reductions, handle pruning, shift reduce parsing, and conflict during shift reduce parsing. Introduction to LR parsing: SLR, Why LR parsers, Items and LR (0) automaton, LR Parsing Algorithm, constructing SLR parsing tables. More powerful LR Parsers: CLR (1) Items, CLR (1) Parsing tables.

UNIT-IV

Intermediate code generation-variants of syntax trees, DAG for expressions. Three address code: addresses and instructions, quadruples, triples, indirect triples. Types and declarations: type expressions, type equivalence. Type checking: rules for type checking, type conversions.

UNIT-V

Symbol Tables: Symbol table format, organization for block structures languages, hashing, tree structures representation of space information. Block structures and non block structure storage allocation: static, Runtime stack and heap storage allocation, storage allocation for arrays, strings and records. Storage organization: static versus dynamic storage allocation, Activation trees, Activation Records, calling sequences.

Text Books:

1. Compilers principles, techniques, &tools- A.V.Aho. J.D.Ullman; pearson Education. Second edition.

Reference Books:

1. Compiler design: Theory, Tools and examples by seth D.Bergmann.

MADANAPALLE INSTITUTE OF TECHNOLOGY & SCIENCE
(UGC - AUTONOMOUS)

M. Tech I Year - II SEM (CSE)

P	T	C
3	0	2

DATA WAREHOUSING DATA MINING & JAVA AND WEB TECHNOLOGIES LAB (14CSP12P02)

Data Warehousing Data Mining Lab

Course Objectives:

- Compare and contrast different uses of data mining as evidenced in both research and application.
- Explain the value of finding associations in market basket data.
- Characterize the kinds of patterns that can be discovered by association rule mining.
- Describe how to extend a relational system to find patterns using association rules.

Course Outcomes:

After completion of the course the student will be able to

- Evaluate different methodologies for effective application of data mining.
- Identify and characterize sources of noise, redundancy, and outliers in presented data.
- Identify mechanisms (on-line aggregation, anytime behavior, interactive visualization) to close the loop in the data mining process.
- Describe why the various close-the-loop processes improve the effectiveness of data mining.

Description: The business of banks is making loans. Assessing the credit worthiness of an applicant is of crucial importance. You have to develop a system to help a loan officer decide whether the credit of a customer is good, or bad. A bank's business rules regarding loans must consider two opposing factors. On the one hand, a bank wants to make as many loans as possible. Interest on these loans is the banks profit source. On the other hand, a bank cannot afford to make too many bad loans. Too many bad loans could lead to the collapse of the bank. The bank's loan policy must involve a compromise: not too strict, and not too lenient.

To do the assignment, you first and foremost need some knowledge about the world of credit. You can acquire such knowledge in a number of ways.
1. Knowledge Engineering. Find a loan officer who is willing to talk. Interview her and try to represent her knowledge in the form of production rules.
2. Books. Find some training manuals for loan officers or perhaps a suitable textbook on finance. Translate this knowledge from text form to production rule form.
3. Common sense. Imagine yourself as a loan officer and make up reasonable rules which can be used to judge the credit worthiness of a loan applicant.
4. Case histories. Find records of actual cases where competent loan officers correctly judged when, and when not to, approve a loan application.

The German Credit Data:
Actual historical credit data is not always easy to come by because of confidentiality rules. Here is one such dataset, consisting of 1000 actual cases collected in Germany. credit dataset (original) Excel spreadsheet version of the German credit data (Down load from web).
In spite of the fact that the data is German, you should probably make use of it for this assignment. (Unless you really can consult a real loan officer !)

A few notes on the German dataset
• DM stands for Deutsche Mark, the unit of currency, worth about 90 cents Canadian (but looks and acts like a quarter).
• owns_telephone. German phone rates are much higher than in Canada so fewer people own telephones.
• foreign_worker. There are millions of these in Germany (many from Turrkey). It is very hard to get German citizenship if you were not born of German parents.
• There are 20 attributes used in judging a loan applicant. The goal is the classify the applicant into one of two categories, good or bad.

Subtasks : (Turn in your answers to the following tasks)

1. List all the categorical (or nominal) attributes and the real-valued attributes seperately.

2. What attributes do you think might be crucial in making the credit assessement ? Come up with some simple rules in plain English using your selected attributes.

3. One type of model that you can create is a Decision Tree - train a Decision Tree using the complete dataset as the training data. Report the model obtained after training.

4. Suppose you use your above model trained on the complete dataset, and classify credit good/bad for each of the examples in the dataset. What % of examples can you classify correctly ? (This is also called testing on the training set) Why do you think you cannot get 100 % training accuracy ?

5. Is testing on the training set as you did above a good idea ? Why orWhy not ?

6. One approach for solving the problem encountered in the previous question is using cross-validation ? Describe what is cross-validation briefly. Train a Decistion Tree again using cross-validation and report your results. Does your accuracy increase/decrease ? Why ? (10 marks)

7. Check to see if the data shows a bias against "foreign workers" (attribute 20),or "personal-status" (attribute 9). One way to do this (perhaps rather simple minded) is to remove these attributes from the dataset and see if the decision tree created in those cases is significantly different from the full dataset case which you have already done. To remove an attribute you can use the preprocess tab in Weka's GUI Explorer. Did removing these attributes have any significant effect? Discuss.

8. Another question might be, do you really need to input so many attributes to get good results? Maybe only a few would do. For example, you could try just having attributes 2, 3, 5, 7, 10, 17 (and 21, the class attribute (naturally)). Try out some combinations. (You had removed two attributes in problem 7. Remember to reload the arff data file to get all the attributes initially before you start selecting the ones you want.)

9. Sometimes, the cost of rejecting an applicant who actually has a good credit (case 1) might be higher than accepting an applicant who has bad credit (case 2). Instead of counting the misclassifcations equally in both cases, give a higher cost to the first case (say cost 5) and lower cost to the second case. You can do this by using a cost matrix in Weka. Train your Decision Tree again and report the Decision Tree and cross-validation results. Are they significantly different from results obtained in problem 6 (using equal cost)?

10. Do you think it is a good idea to prefer simple decision trees instead of having long complex decision trees ? How does the complexity of a Decision Tree relate to the bias of the model ?

11. You can make your Decision Trees simpler by pruning the nodes. One approach is to use Reduced Error Pruning - Explain this idea briefly. Try reduced error pruning for training your Decision Trees using cross-validation (you can do this in Weka) and report the Decision Tree you obtain ? Also, report your accuracy using the pruned model. Does your accuracy increase ?

12. (Extra Credit): How can you convert a Decision Trees into "if-then-else rules". Make up your own small Decision Tree consisting of 2-3 levels and convert it into a set of rules. There also exist different classifiers that output the model in the form of rules - one such classifier in Weka is rules.PART, train this model and report the set of rules obtained. Sometimes just one attribute can be good enough in making the decision, yes, just one ! Can you predict what attribute that might be in this dataset ? OneR classifier uses a single attribute to make decisions (it chooses the attribute based on minimum error). Report the rule obtained by training a one R classifier. Rank the performance of j48, PART and oneR.

Task Resources:

Andrew Moore's Data Mining Tutorials (See tutorials on Decision Trees and Cross Validation)

- Decision Trees (Source: Tan, MSU)
- Tom Mitchell's book slides (See slides on Concept Learning and Decision Trees)
- Weka resources:
    - Introduction to Weka (html version) (download ppt version)
    - Download Weka
    - Weka Tutorial
    - ARFF format
    - Using Weka from command line

Java and Web Technologies Lab

S.No.	Name of Exercise
1.	Implementation of Stack Using Arrays & Creation of college website
2.	Implementation of Queue Using Arrays & Working on CSS
3.	Implementation of List Using Arrays & Client side validation using Javascript
4.	Towers of Hanoi & Displaying Book Information using XML
5.	Infix to Postfix Conversion & Developing College Website using Java Applets
6.	Binary Tree Search Operations & Developing Online Shopping Program
7.	Binary Tree Traversal & Converting INR to Equivalent Dollar Value
8.	AVL Tree Operations & Creation of Java Bean
9.	B-Tree Operations & Database Access
10.	Implementation of Merge Sort Technique & Installation of Tomcat
11.	Implementation of Quick Sort Technique & Servlets
12.	Java Server Pages