L	T	P	C
4	0	0	4

(17F00401) OBJECT ORIENTED ANALYSIS & DESIGN

Course Objectives

- To understand how to solve complex problems
- Analyze and design solutions to problems using object oriented approach
- Study the notations of Unified Modeling Language

Course Outcomes:

- Ability to find solutions to the complex problems using object oriented approach
- Represent classes, responsibilities and states using UML notation
- Identify classes and responsibilities of the problem domain

Unit-I

Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems, Evolution of Object Model, Foundation of Object Model, Elements of Object Model, Applying the Object Model.

Unit-II

Classes and Objects: Nature of object, Relationships among objects, Nature of a Class, Relationship among Classes, Interplay of Classes and Objects, Identifying Classes and Objects, Importance of Proper Classification, Identifying Classes and Objects, Key abstractions and Mechanisms.

Unit-III

Introduction to UML: Why model, Conceptual model of UML, Architecture, Classes, Relationships, Common Mechanisms, Class diagrams, Object diagrams.

Unit-IV

Structural Modeling: Package Diagram, Composite Structure Diagram, Component Diagram, Deployment Diagram, Profile Diagram.

Unit-V

Behavioral Modeling: Use Case Diagram, Activity Diagrams, State Machine Diagrams, Sequence Diagram, Communication Diagram, Timing Diagram, Interaction Overview Diagram.

Text Books:

1. "Object- Oriented Analysis And Design with Applications", Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston, PEARSON, 3rd edition, 2013.
2. "The Unified Modeling Language User Guide", Grady Booch, James Rumbaugh, Ivar Jacobson, PEARSON 12th Impression, 2012.
3. http://www.omg.org/

References:

1. "Object-oriented analysis and design using UML", Mahesh P. Matha, PHI
2. "Head first object-oriented analysis and design", Brett D. McLaughlin, Gary Pollice, Dave West, O'Reilly
3. "Object-oriented analysis and design with the Unified process", John W. Satzinger, Robert B. Jackson, Stephen D. Burd, Cengage Learning
4. "The Unified modeling language Reference manual", James Rumbaugh, Ivar Jacobson, Grady Booch, Addison-Wesley

L	T	P	C
4	0	0	4

(17F00402) DESIGN AND ANALYSIS OF ALGORITHMS

Course Objectives:

- To know the importance of the complexity of a given algorithm.
- To study various algorithm design techniques.
- To utilize data structures and/or algorithmic design techniques in solving new problems.
- To know and understand basic computability concepts and the complexity classes P, NP, and NP-Complete.
- To study some techniques for solving hard problems.

Course Outcomes:

- Analyze the complexity of the algorithms
- Use techniques divide and conquer, greedy, dynamic programming, backtracking, branch and bound to solve the problems.
- Identify and analyze criteria and specifications appropriate to new problems, and choose the appropriate algorithmic design technique for their solution.
- Able to prove that a certain problem is NP-Complete.

UNIT I

Introduction: What is an Algorithm, Algorithm specification, Performance analysis.
Divide and Conquer: General method, Binary Search, Finding the maximum and minimum, Merge sort, Quick Sort, Selection sort, Stressen's matrix multiplication.

UNIT II

Greedy Method: General method, Knapsack problem, Job Scheduling with Deadlines, Minimum cost Spanning Trees, Optimal storage on tapes, Single-source shortest paths.
Dynamic programming: General Method, Multistage graphs, All-pairs shortest paths, Optimal binary search trees, 0/1 knapsack, The traveling sales person problem.

UNIT III

Basic Traversal and Search Techniques: Techniques for binary trees, Techniques for Graphs, Connected components and Spanning trees, Bi-connected components and DFS
Back tracking: General Method, 8 – queens problem, Sum of subsets problem, Graph coloring and Hamiltonian cycles, Knapsack Problem.

UNIT IV

Branch and Bound: The method, Travelling salesperson, 0/1 Knapsack problem, Efficiency Considerations.
Lower Bound Theory: Comparison trees, Lower bounds through reductions – Multiplying triangular matrices, inverting a lower triangular matrix, computing the transitive closure.

UNIT V

NP – Hard and NP – Complete Problems: NP Hardness, NP Completeness, Consequences of beingin P, Cook's Theorem, Reduction Source Problems, Reductions: Reductions for some known problems

Text Books:

1. "Fundamentals of Computer Algorithms", Ellis Horowitz, S. Satraj Sahani and Rajasekhran, 2nd edition, University Press.2014,
2. "Design and Analysis of Algorithms", Parag Himanshu Dave, Himanshu Bhalchandra Dave, Pearson Education, Second Edition, 2009.

Reference Books:

1. "Introduction to Algorithms", second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd./ Pearson Education.
2. "Introduction to Design and Analysis of Algorithms A strategic approach", R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc Graw Hill.
3. "Data structures and Algorithm Analysis in C++", Allen Weiss, Second edition, Pearson education.
4. "Design and Analysis of algorithms", Aho, Ullman and Hopcroft,Pearson education.
5. "Algorithms" – Richard Johnson baugh and Marcus Schaefer, Pearson Education

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00403) LINUX PROGRAMMING

Course Objectives:

- To understand the UNIX system structure.
- To understand and use command line shell.
- To make effective use of UNIX utilities and Shell scripting language such as bash.
- To produce programs similar to standard UNIX utilities such as ls,mv,cp etc.using Unix system calls.
- To develop the skills necessary for Unix systems programming including file system programming,process and signal management, and interprocess communication.
- To develop the basic skills required to write network programs using Sockets.
-

Course Outcomes:

- Able to describe and use the UNIX operating system.
- Able to describe and use the fundamental UNIX system tools and utilities.
- Able to describe and write shell scripts in order to perform basic shell programming.
- Able to describe and understand the UNIX file system.

Prerequisites

Familiarity with using Unix Programming environment and having a good working knowledge of the C programming language.

UNIT I

Linux Utilities-File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Sed-Scripts,Operation,Addresses,Commands,Applications, awk-Execution,Fields and Records, Scripts,Operation,Patterns,Actions,Associative Arrays,String and Mathematical functions,System commands in awk,Applications..

Shell programming with Bourne again shell(bash)- Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

Review of C programming concepts-arrays, strings (library functions),pointers,function pointers,structures,unions,libraries in C.

UNIT II

Files and Directories- File Concept, File types, File System Structure,file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, creat, read, write, close, lseek, dup2,file status information-stat family, file and record locking-lockf and fcntl functions,file permissions - chmod, fchmod,file ownership-chown,lchown,fchown, links-soft links and hard links – symlink, link, unlink.

Directories-Creating,removing and changing Directories-mkdir, rmdir, chdir, obtaining current working directory-getcwd, Directory contents,Scanning Directories-opendir, readdir, closedir,rewinddir, seekdir, telldir functions.

UNIT III

Process – Process concept, Layout of a C program image in main memory,Process environment-environment list,environment variables,getenv,setenv,Kernel support for process, process identification, process hierarchy,process states, process control - process creation, replacing a process image,waiting for a process, process termination, zombie process,orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, system,I/O redirection,Process Groups,Sessions and Controlling Terminal,Differences between threads and processes.

Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise , alarm, pause, abort, sleep functions.

UNIT IV

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems,pipes-creation,IPC between related processes using unnamed pipes, FIFOs-creation,IPC between unrelated processes using FIFOs(Named pipes), differences between unnamed and named pipes,popen and pclose library functions.

Message Queues- Kernel support for messages, APIs for message queues, client/server example. Semaphores-Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

UNIT V

Shared Memory- Kernel support for shared memory, APIs for shared memory, shared memory example.

Sockets- Introduction to Berkeley Sockets, IPC over a network,Client-Server model,Socket address structures (Unix domain and Internet domain),Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection,Multiple simultaneous clients, Comparison of IPC mechanisms.

TEXT BOOKS:

1. Unix System Programming using C++, T.Chan, PHI.
2. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH,2006.
3. Beginning Linux Programming, 4th Edition, N.Matthew, R.Stones,Wrox, Wiley India Edition,rp-2008.
4. Unix Network Programming ,W.R.Stevens,PHI.
5. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.

REFERENCE BOOKS:

1. Linux System Programming, Robert Love, O'Reilly, SPD, rp-2007.
2. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson Education, 2003.
3. Advanced Programming in the Unix environment, 2nd Edition, W.R.Stevens, Pearson Education.
4. System Programming with C and Unix,A.Hoover,Pearson.
5. Unix System Programming,Communication,Concurrency and Threads,K.A.Robbins and S.Robbins,Pearson Education.
6. Unix shell Programming,S.G.Kochan and P.Wood,3rd edition,Pearson Education.
7. Shell Scripting,S.Parker,Wiley India Pvt. Ltd.
8. C Programming Language,Kernighan and Ritchie,PHI

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00404) COMPUTER GRAPHICS AND MULTIMEDIA
Elective-I

OBJECTIVES:
- To develop, design and implement two and three dimensional graphical structures
- To enable students to acquire knowledge Multimedia compression and animations
- To learn Creation, Management and Transmission of Multimedia objects.

OUTCOMES:
- Ability to develop programs to control the content, structure and appearance of objects.
- Ability to design, organize and produce multimedia projects of all kinds

UNIT I

2D PRIMITIVES:

Elements of pictures created in computer graphics – Graphics input primitives and devices Drawing primitives in open GL and Basic open GL programming - open GL basic Graphics primitives – Output primitives – Line, Circle and Ellipse drawing algorithms – Attributes of output primitives.

UNIT II

2D GEOMETRIC TRANSFORMATIONS:

2D Viewing – Window-Viewport Transformation - Two dimensional Geometric transformations – Line, Polygon, Curve and Text clipping algorithms.

UNIT III

MULTIMEDIA BASICS

Introduction and definitions – applications – elements – Animations – Compression – Types of Compressions: Lossless – Lossy – Video compression – Image Compression – Audio compression – Data and file format standards – Multimedia data structures: KD Trees –R trees.

UNIT IV

MULTIMEDIA:

Where to use multimedia, Text: The power of meaning, About fonts and faces, Images: Before you start to create, Making still images, colour, Sound: The power of sound, Digital audio, MIDI Audio, MIDI Vs Digital audio, Multimedia system sounds, Audio File formats, Animation, Video: Using video, How video works and is displayed, Digital video containers

UNIT V

MULTIMEDIA AUTHORING AND APPLICATIONS Creating interactive multimedia – Multimedia Authoring Systems – Multimedia Authoring Software Applications – Video On demand – Virtual Reality – Augmented Reality – Content based retrieval in digital libraries.

TEXT BOOKS:

1. Donald D. Hearn, M. Pauline Baker and Warren Carithers, “Computer Graphics with OpenGL”, Fourth Edition, Pearson Education, 2010.
2. Ze-Nian Li and Mark S.Drew, “Fundamentals of Multimedia”, First Edition, Pearson Education, 2007
3. Multimedia: Making It Work”, , Tay Vaughan, 8th Edition, 2011, Tata McGrawHill Edition

REFERENCE BOOKS:

1. F.S.Hill, “Computer Graphics using OPENGL”, Second edition, Pearson Education, 2003.
2. Prabhat K Andleigh, Kiran Thakrar, “Multimedia systems design”, First Edition, PHI, 2007.

L	T	P	C
4	0	0	4

(17F00405) INTERNET OF THINGS

Elective-I

Objectives

- Makes clear view over physical computing, ubiquitous computing, or the Internet of Things, it's a hot topic in technology.
- It discusses design concepts that will make IOT products eye-catching and appealing.

Outcomes

- Ability to combine sensors, servos, robotics, Arduino chips, and more with various or the Internet, to create interactive, cutting-edge devices.
- Better idea of the overview of necessary steps to take the idea of IOT concept through production

UNIT 1

Introduction - Internet of Things – **Design Principles for Connected Devices** – Web Thinking for Connected Devices – **Internet Principles** – IP – TCP – IP Protocol Suite – UDP – IP Address – MAC Address – TCP and UDP Ports – Application Layer Protocols.

UNIT 2

Prototyping – Prototypes and Production – Cloud – Open Source vs Closed Source – Tapping into the Community **– Prototyping Embedded Devices** – Electronics – Embedded Computing Basics – Ardunio – Raspberry Pi – Beagle Bone Black – Electronic Imp.

UNIT 3

Prototyping thePhysicalDesign – Laser Cutting – 3D Printing – CNC Milling – Repurposing and Recycling – **Prototyping Online Components** – New API – Real Time Reactions – Other Protocols.

UNIT 4

Techniques for writing Embedded Code – Memory Management – Performance and Battery life – Libraries – Debugging – **Business Models** – Models – Funding an Internet of Things Startup.

UNIT 5

Moving to Manufacture – Designing Kits – Designing Printed Circuit Boards – Manufacturing Printed Circuit Boards – Mass Producing the case and other Fixtures – Scaling up Software – **Ethics** – Characterizing the Internet of Things – Control – Environment – Solutions.

Text Books:

1. Adrian Mcewen and HakinCassimally, “Designing The Internet of Things” Wiley Publications , 2015

Reference Books:

1. Vijay Madisetti and ArshdeepBahga, “Internet of Things (A Hands-on-Approach)”, 1stEdition, VPT, 2014.
2. Francis daCosta, “Rethinking the Internet of Things: A Scalable Approach to Connecting Everything”, 1st Edition, Apress Publications, 2013
3. CunoPfister, “Getting Started with the Internet of Things”, O“Reilly Media, 2011

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00406) ARTIFICIAL INTELLIGENCE
Elective-I

Course Objective:

- To learn the difference between optimal reasoning Vs human like reasoning
- To understand the notions of state space representation, exhaustive search, heuristic search along with the time and space complexities
- To learn different knowledge representation techniques
- To understand the applications of AI namely, Game Playing, Theorem Proving, Expert Systems, Machine Learning and Natural Language Processing

Learning Outcome:

- Possess the ability to formulate an efficient problem space for a problem expressed in English
- Possess the ability to select a search algorithm for a problem and characterize its time and space complexities.
- Possess the skill for representing knowledge using the appropriate technique
- Possess the ability to apply AI techniques to solve problems of Game Playing, Expert Systems and Machine Learning.

Unit – I

Foundations of AI: What is AI, History of AI, Strong and weak AI, The State of the Art.
Intelligent Agents: Agents and Environments, Good Behavior: The Concept of Rationality, The Nature of Environments, The Structure of Agents.

Unit – II

Solving Problems by Searching: Problem – Solving Agents, Example Problems, Searching for Solutions, uniformed search Strategies, Informed (Heuristic) Search Strategies, Heuristic Functions.

Unit – III

Knowledge Representation: Ontological Engineering, Categories and Objects, Events, Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information, The Internet Shopping World.

Unit – IV

Learning from Examples: Forms of Learning, Supervised Learning, Learning Decision Trees, Evaluating and Choosing the Best Hypothesis, The Theory of Learning, Regression and Classification with Learner Models, Nonparametric Models, Support Vector Machines, Ensemble Learning, Practical Machine Learning.

Unit – V

Learning Probabilistic Models: Statistical Learning, Learning with Complete data, Learning with Hidden variables: The EM Algorithm.

Text Books :

1. "Artificial Intelligence A Modern Approach", Stuart J. Russell & Peter Norvig – Pearson.
2. "Artificial Intelligence", Elaine Rich, Kevin Knight & Shivashankar B Nair – McGraw Hill Education.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00407) BIG DATA ANALYTICS
Elective-II

Objectives:

- To learn to analyze the big data using intelligent techniques.
- To understand the various search methods and visualization techniques.
- To learn to various techniques for mining data stream.
- To understand the applications using Map Reduce Concepts.

Outcomes:

On completion of this course the student will able to

- Analyze the big data analytics techniques for useful business application.
- Design efficient algorithms for mining the data from large volumes.
- Analyze the HADOOP and Map Reduce technologies associated with big data analytics.
- Explore on big data applications using Pig and Hive.

UNIT-I

Introduction to Big Data

Introduction to Big Data Platform – Challenges of Conventional System – Intelligent data analysis – Nature of Data – Analytic Processes and Tool – Analysis vs Reporting – Modern Data Analytic Tool – Statistical Concepts: Sampling Distributions – Re-Sampling – Statistical Inference – Prediction Error.

UNIT- II

Mining Data Streams

Introduction To Stream Concepts – Stream Data Model and Architecture - Stream Computing – Sampling Data in a Stream – Filtering Stream – Counting Distinct Elements in a Stream – Estimating Moments – Counting Oneness in a Window – Decaying Window – Real time Analytics Platform(RTAP) Applications – Case Studies – Real Time Sentiment Analysis, Stock Market Predictions.

UNIT – III

Hadoop

History of Hadoop- The Hadoop Distributed File System – Components of Hadoop – Analyzing the Data with Hadoop – Scaling Out – Hadoop Streaming – Design of HDFS- Java interfaces to HDFSBasics- Developing a Map Reduce Application – How Map Reduce Works – Anatomy of a Map Reduce Job run – Failures – Job Scheduling – Shuffle and Sort – Task Execution – Map Reduce Types and Formats – Map Reduce Features.

UNIT – IV

Hadoop Environment

Setting up a Hadoop Cluster – Cluster specification – Cluster Setup and Installation –Hadoop Configuration – Security in Hadoop – Administering Hadoop – HDFS – Monitoring – Maintence – Hadoop Benchmarks – Hadoop in the Cloud.

UNIT –V

Frameworks

Applications on Big Data Using Pig and Hive – Data Processing operators in Pig – Hive Services – HiveQL – Querying Data in Hive – fundamentals of HBase and Zookeeper – IBM Info Sphere Big Insights and Streams. Visualization - Visual data analysis techniques, interaction techniques; Systems and applications.

Text Books:

1. Michael Berthold, David J.Hand, Intelligent Data Analysis, Spingers, 2007.
2. Tom White, Hadoop: The Definitive Guide Third Edition, O'reilly Media, 2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, Uderstanding Big Data : Analytics for Enterprise Class Hadoop and Streaming Data, McGrawHill Publishing, 2012.
4. AnandRajaraman and Jeffrey David UIIman, Mining of Massive Datasets Cambridge University Press, 2012.

Reference Books:

1. Bill Franks, Taming the big Data tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & sons, 2012.
2. Glenn J. Myatt, Making Sense of Data , John Wiley & Sons, 2007 Pete Warden, Big Data Glossary, O'Reilly, 2011.
3. Jiawei Han, MichelineKamber, Data Mining Concepts and Techniques, Second Edition.
4. Elsevier, Reprinted 2008. Da Ruan, Guoquing Chen, Etienne E.Kerre, Geert Wets, Intelligent Data Mining, Springer, 2007.
5. Paul Zikopoulos, Dirk deRoos, Krishnan Parasuraman, Thomas Deutsch, James Giles, David Corrigan, Harness the Power of Big Data the IBM Big Data Platform, Tata McGraw Hill Publications, 2012.
6. Michael Minelli (Author), Michele Chambers (Author), AmbigaDhirraj (Author), Big Data, BigSnalytics.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00408) SCRIPTING LANGUAGES
Elective-II

Course Objectives:
The primary objective of the course is to learn web programming by designing and developing a web based project and also learn basic User Interface Principles.

Course Outcomes:
At the end of the course, the student should be able to:
- Ability to build interactive, data-driven sites and solve real time problems using PHP
- Ability to create dynamic web pages using the PHP scripting language and MySQL database.
- Understand the concepts of object-oriented programming as used in Python: classes, subclasses, inheritance, and overriding. Understand the basics of OO design.

Unit-1
Essential PHP: Enter PHP, Getting PHP, Creating your Development Environment, Creating and running first PHP page, Mixing HTML and PHP, Printing some Text, Using PHP "here" documents, command-line PHP, comments, variables, storing data, Interpolating Strings, creating variables and constants and data types.
Operators and flow controls: PHP's Math Operators, working with the assignment operator, Increment and decrement values, String operators, Bitwise, Execution operator, Operator precedence, If statement, comparison operator, logical operators, else, elif, ternary, switch, for, while, do-while loops, foreach, terminating, skipping iterations.
Strings and Arrays: String Functions, Converting to and from Strings, Formatting Text Strings, Arrays, Array PHP Functions, Sorting Arrays, handling Multi dimensional arrays, splitting and merging arrays.

Unit-2
Creating Functions: Creating in Functions in PHP, Passing Functions some data, Passing Arrays to functions, Passing by reference, default arguments, returning data from functions, returning arrays and lists, variable scope in php, accessing global data, static variables, conditional functions, nesting functions, creating include files, returning errors from functions.
Reading data in web Pages: Setting up web pages to communicate with PHP, Handling of Text fields, Text areas, Check boxes, Radio buttons, List boxes, Password controls, Hidden controls, Image Maps, File Uploads, buttons.
PHP browser Handling Power: Using PHP's Server Variables, HTTP Headers, User's browser type, Redirecting browsers with HTTP Headers, Dumping a form's data all at once, Handling form data with custom arrays, putting all it into the page, performing data validation, checking entered data, requiring numbers and text, Persisting user data, client side data validation, handling html tags in user input.

Unit-3
Object Oriented Programming: Creation of class and Objects, setting access to Properties and methods, Using constructors to Initialize Objects, using Destructors to clean up after objects, Inheritance, Method Overriding, overloading, autoloading classes .
Advanced Object Oriented Programming: Creating Static Methods, static members and

Inheritance, abstract classes, interfaces, Supporting Object Iteration, Comparing Objects, creating class constants, final key word, cloning objects.

Working with Database: what is database?, Essential sql, creating MySql database, creating table, Accessing the database in php, Updating, Inserting, deleting, sorting data.

Sessions, Cookies, and FTP: Setting cookie and reading cookie, setting cookie expiration, deleting cookies, Working with FTP, Downloading and Uploading and deleting files with FTP, creating and removing directories with FTP, Sending E-mail, sending advanced email, adding attachments to the email, sorting data.

Unit-4

Introduction to PYHON: Getting Started with Python Programming, Detecting and correcting Syntax errors.

Software Development, data types, and Expressions: Software Development Process, case study: Income tax calculator, Strings, Assignments, and comments, Numeric data types and character sets, Expressions, Using functions and modules, Control statements.

Design with Functions: Functions as abstraction Mechanism, Problem solving with top-down design, recursive function, File system, Managing a program's namespace, Higher Order Functions.

Unit-5

String and Text Files: Accessing characters and substrings in strings, data encryption, strings and number Systems, String Methods, Text Files, Case Study : Text Analysis.

List and Dictionaries: List, Defining simple Functions, case study: generating sentences, dictionaries, case study: nondirective psychology.

Design with classes: Getting inside objects and classes, case study: Playing the game of craps, data modelling examples, Structuring classes with Inheritance and Polymorphism.

Text Books:

1. The Complete Reference PHP by Steven Holzner, M H HILL Education, Indian Edition, 2008.
2. Fundamentals of PYTHON By Kenneth A. and Lambert and B.L Juneja, Cengage Learning, 2012.

Reference Books:

1. Core Python application and programming by Wesley J.Chun, Pearson, 3rd edition.
2. Introduction to computing and programming in python by Mark. J. Guzdial and Barbara Ericson, Pearson, 4th edition.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR
MCA II Year II semester

L	T	P	C
4	0	0	4

(17F00409) DISTRIBUTED SYSTEMS
Elective-II

Course Objectives:

The student should be made to:
- Understand the issues involved in studying process and resource management.
- Understand in detail the system level and support required for distributed system.
- Introduce the idea of peer to peer services and file system.
- Understand foundations of Distributed Systems.

Course Outcomes:
Student should be able to:
- Design process and resource management systems.
- Apply remote method invocation and objects.
- Apply network virtualization.
- Discuss trends in Distributed Systems.

UNIT I
INTRODUCTION
Examples of Distributed Systems – Trends in Distributed Systems – Focus on resource sharing – Challenges. Case study: World Wide Web.

UNIT II
COMMUNICATION IN DISTRIBUTED SYSTEM
System Model – Inter process Communication - the API for internet protocols – External data representation and Multicast communication. Network virtualization: Overlay networks. Case study: MPI Remote Method Invocation And Objects: Remote Invocation – Introduction - Request-reply protocols - Remote procedure call - Remote method invocation. Case study: Java RMI - Group communication - Publish-subscribe systems - Message queues - Shared memory approaches - Distributed objects - Case study: Enterprise Java Beans -from objects to components.

UNIT III
PEER TO PEER SERVICES AND FILE SYSTEM
Peer-to-peer Systems – Introduction - Napster and its legacy - Peer-to-peer – Middleware - Routing overlays. Overlay case studies: Pastry, Tapestry- Distributed File Systems – Introduction - File service architecture – Andrew File system. File System: Features-File model -File accessing models - File sharing semantics Naming: Identifiers, Addresses, Name Resolution – Name Space Implementation – Name Caches – LDAP.

UNIT IV
SYNCHRONIZATION AND REPLICATION

Introduction - Clocks, events and process states - Synchronizing physical clocks- Logical

time and logical clocks – Global states – Coordination and Agreement – Introduction - Distributed mutual exclusion – Elections – Transactions and Concurrency Control– Transactions -Nested transactions – Locks – Optimistic concurrency control - Timestamp ordering – Atomic Commit protocols -Distributed deadlocks – Replication – Case study – Coda.

UNIT V
PROCESS & RESOURCE MANAGEMENT

Process Management: Process Migration: Features, Mechanism - Threads: Models, Issues, Implementation. Resource Management: Introduction- Features of Scheduling Algorithms – Task Assignment Approach – Load Balancing Approach – Load Sharing Approach.

TEXT BOOK:

1. George Coulouris, Jean Dollimore and Tim Kindberg, “Distributed Systems Concepts and Design”, Fifth Edition, Pearson Education, 2012.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
0	0	4	2

(17F00410) OBJECT ORIENTED ANALYSIS AND DESIGN LAB

Course Objectives:

- Practice the notation for representing various UML diagrams
- Analyze and design the problem by representing using UML diagrams
- Become familiar with all phases of OOAD

Course Outcomes:

- Find solutions to the problems using object oriented approach
- Represent using UML notation and interact with the customer to refine the UML diagrams

UML diagrams to be developed are:

1. Use Case Diagram.
2. Class Diagram.
3. Sequence Diagram.
4. Collaboration Diagram.
5. State Diagram
6. Activity Diagram.
7. Component Diagram
8. Deployment Diagram.
9. Test Design.

Description for an ATM System

The software to be designed will control a simulated automated teller machine (ATM) having a magnetic stripe reader for reading an ATM card, a customer console (keyboard and display) for interaction with the customer, a slot for depositing envelopes, a dispenser for cash (in multiples of Rs. 100, Rs. 500 and Rs. 1000), a printer for printing customer receipts, and a key-operated switch to allow an operator to start or stop the machine. The ATM will communicate with the bank's computer over an appropriate communication link. (The software on the latter is not part of the requirements for this problem.)

The ATM will service one customer at a time. A customer will be required to insert an ATM card and enter a personal identification number (PIN) - both of which will be sent to the bank for validation as part of each transaction. The customer will then be able to perform one or more transactions. The card will be retained in the machine until the customer indicates that he/she desires no further transactions, at which point it will be returned - except as noted below.

The ATM must be able to provide the following services to the customer:

1. A customer must be able to make a cash withdrawal from any suitable account linked to the card, in multiples of Rs. 100 or Rs. 500 or Rs. 1000. Approval must be obtained from the bank before cash is dispensed.
2. A customer must be able to make a deposit to any account linked to the card, consisting of cash and/or checks in an envelope. The customer will enter the amount of the deposit into the ATM, subject to manual verification when the envelope is removed from the machine by an operator. Approval must be obtained from the bank before physically accepting the envelope.
3. A customer must be able to make a transfer of money between any two accounts linked to the card.

4. A customer must be able to make a balance inquiry of any account linked to the card.

5. A customer must be able to abort a transaction in progress by pressing the Cancel key instead of responding to a request from the machine.

The ATM will communicate each transaction to the bank and obtain verification that it was allowed by the bank. Ordinarily, a transaction will be considered complete by the bank once it has been approved. In the case of a deposit, a second message will be sent to the bank indicating that the customer has deposited the envelope. (If the customer fails to deposit the envelope within the timeout period, or presses cancel instead, no second message will be sent to the bank and the deposit will not be credited to the customer.)

If the bank determines that the customer's PIN is invalid, the customer will be required to re-enter the PIN before a transaction can proceed. If the customer is unable to successfully enter the PIN after three tries, the card will be permanently retained by the machine, and the customer will have to contact the bank to get it back.

If a transaction fails for any reason other than an invalid PIN, the ATM will display an explanation of the problem, and will then ask the customer whether he/she wants to do another transaction.

The ATM will provide the customer with a printed receipt for each successful transaction

The ATM will have a key-operated switch that will allow an operator to start and stop the servicing of customers. After turning the switch to the "on" position, the operator will be required to verify and enter the total cash on hand. The machine can only be turned off when it is not servicing a customer. When the switch is moved to the "off" position, the machine will shut down, so that the operator may remove deposit envelopes and reload the machine with cash, blank receipts, etc.

List of Tasks for which students have to design all UML diagrams:

1. Banking system
2. Online bookshop system
3. University Systems
4. Library management system
5. Hospital management system
6. Result processing system

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
0	0	4	2

(17F00411) DESIGN AND ANALYSIS OF ALGORITHMS LAB

Course Objectives

- Implement the various algorithms that are being studied in Design and Analysis of Algorithms subject in C++/Java.

Note:You may develop programs using java or C++

1. Write a program that implements Prim's algorithm to generate minimum cost spanning tree.
2. Write a program that implements Kruskal's algorithm to generate minimum cost spanning tree.
3. Write a program to implement Huffman's algorithm for text compression.
4. Write a program to implement Dijkstra's algorithm for Single source shortest path problem.
5. Write a program to implement Floyd's algorithm for the All pairs shortest path problem.
6. Write a program to implement greedy algorithm for job sequencing with deadlines.
7. Write programs for the implementation of bfs and dfs for a given graph.
8. Write a program to find Minimum Cost Binary Search Tree.
9. Write a program to implement Dynamic Programming algorithm for 0/1 Knapsack problem.
10. Write a program to implement the Backtracking algorithm for the sum of subsets problem.
11. Write programs to implement backtracking algorithms for
    a) N-queens problem
    b) The Hamiltaonian cycles problem
    c) The m-colourings graph problem

TEXT BOOKS

1. Data structures and Algorithms in java,3rd edition,A.Drozdek,Cengage Learning.
2. Data structures with Java,J.R.Hubbard,2nd edition,Schaum's Outlines,TMH.
3. Data structures and algorithms in Java, 2nd Edition, R.Lafore, Pearson Education.
4. Data Structures using Java, D.S.Malik and P.S. Nair, Cengage Learning.
5. Data structures, Algorithms and Applications in java, 2nd Edition, S.Sahani, Universities Press.
6. Data structures, Algorithms and Applications in C++, 2nd Edition, S.Sahani, Universities Press.
7. Data structures and Algorithm Analysis in C++,2nd Edition,M.A.Weiss,Pearson education.
8. Design and Analysis of Algorithms, P.H.Dave and H.B.Dave, Pearson education.
9. Data structures and java collections frame work,W.J.Collins,Mc Graw Hill.

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

MCA II Year II semester

L	T	P	C
0	0	4	2

(17F00412) LINUX PROGRAMMING LAB.

Course Objectives:

- To implement some standard Unix utilities using system calls.
- To develop shell scripts to solve problems.
- To produce programs in C for network-based applications.
- To implement CPU scheduling algorithms,file allocation methods and page replacement algorithms in C.

Course Outcomes:

After completion of the course students will be able to

• Work confidently in Unix/Linux environment
• Write shell scripts to automate various tasks
• Master the basics of linux administration

Note: Use Bash for Shell scripts.

1. Write a shell script that accepts a file name, starting and ending line numbers as arguments and displays all the lines between the given line numbers.
2. Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.
3. Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
4. Write a shell script that receives any number of file names as arguments checks if every argument supplied is a file or a directory and reports accordingly. Whenever the argument is a file, the number of lines on it is also reported.
5. Write a shell script that accepts a list of file names as its arguments, counts and reports the occurrence of each word that is present in the first argument file on other argument files.
6. Write a shell script to list all of the directory files in a directory.
7. Write a shell script to find factorial of a given integer.
8. Write an awk script to count the number of lines in a file that do not contain vowels.
9. Write an awk script to find the number of characters, words and lines in a file.
10. Write a C program that makes a copy of a file using standard I/O and system calls.
11. Implement in C the following Unix commands using System calls
    a). cat b) mv
12. Write a C program to list files in a directory.
13. Write a C program to emulate the Unix ls –l command.
14. Write a C program to list for every file in a directory, its inode number and file name.
15. Write a C program that redirects standard output to a file.Ex: ls > f1.
16. Write a C program to create a child process and allow the parent to display “parent” and the child to display “child” on the screen.
17. Write a C program to create a Zombie process.

18. Write a C program that illustrates how an orphan is created.
19. Write a C program that illustrates how to execute two commands concurrently with a command pipe. Ex:- ls –l | sort
20. Write C programs that illustrate communication between two unrelated processes using named pipe.
21. Write a C program in which a parent writes a message to a pipe and the child reads the message.
22. Write a C program (sender.c) to create a message queue with read and write permissions to write 3 messages to it with different priority numbers.

23. Write a C program (receiver.c) that receives the messages (from the above message queue as specified in (22)) and displays them.

24. Write a C programs to transfer a large amount of data between processes,using
a) a pipe b)a FIFO c)a message queue.

25. Write a C program to allow cooperating processes to lock a resource for exclusive use, using: a)Semaphores b)flock or lockf system calls.

26. Write a C program that illustrates suspending and resuming processes using signals.

27. Write a C program that implements a producer-consumer system with two processes. (using Semaphores).

28. Write client and server programs(using c) for interaction between server and client processes using
Unix Domain sockets.

29. Write client and server programs(using c) for interaction between server and client processes using
Internet Domain sockets.

30. Write C programs that illustrate two processes communicating using shared memory.

TEXT BOOKS:

1. Advanced Unix Programming, N.B.Venkateswarulu, BS Publications.
2. Unix and Shell programming, B.A.Forouzan and R.F.Gilberg, Cengage Learning.
3. Unix and Shell Programming, M.G. Venkatesh Murthy, Pearson Education, 2005.
4. Unix Shells by Example, 4th Edition, Elllie Quigley, Pearson Education.
5. Sed and Awk, O.Dougherty&A.Robbins,2nd edition,SPD.