(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Programming & Data Structures

SEMESTER - I **L-T-P-C: 3-0-0-3**

Course Objectives:

The objective of this course is to explore basic data structures such as stacks and queues, introduce a variety of data structures such as hash tables, search trees, tries, heaps, graphs, sorting and pattern matching algorithms

Course Outcomes (CO): *At the end of the course, student will be able to*

Course Outcomes		Knowledge Level (K)#
CO1	Implement basic programs by using C concepts.	**K1**
CO2	Select the data structures that efficiently model the information in a problem	**K3**
CO3	Assess efficiency trade-offs among different data structure implementations or combinations	**K5**
CO4	Implement and know the application of algorithms for sorting and pattern matching.	**K2**
CO5	Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and AVL-trees	**K6**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	1	1			1	3	1	1	1
CO2	2	3						2		
CO3	1	3					1	1		
CO4	1	3	2			1	2	2		
CO5	3	3	1	1			2	2	1	2

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT - I :

Introduction to C: Constants and variables, Operators and Expressions, Managing Input and Output operators, Decision making-branching and looping, Arrays,

UNIT-II:

Functions, Structures and Unions, Pointers, File handling in C.

UNIT – III :

Data structure: Definition, types of data structures Recursion Definition, Design Methodology and Implementation of recursive algorithms, Linear and binary recursion. Preliminaries of algorithms, analysis and complexity

.**Linear list** – singly linked list, Double linked list and circular linked list - implementation, insertion, deletion and searching operations on linear list.

UNIT - IV:

Stacks-Operations, array and linked representations of stacks, stack applications, **Queues**-operations, array and linked representations. **Hash Table Representation:** hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing and rehashing, extendible hashing.

UNIT - V:

Sorting Techniques: Insertion sort, selection sort, exchange-bubble sort, quick sort and merge sort Algorithms. **Trees:** Binary Trees, terminology, representation and traversals- pre, post & in order traversals. **Search Trees:** Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion

Text Books:

1. Programming in ANSI C, 5e, E. Balaguruswamy, TMH
2. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and
Susan Anderson Freed, Universities Press.
3. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M.J. Augenstein,
PHI/Pearson Education.

Reference Books:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A. Forouzan, Cengage Learning.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Computer Organization

SEMESTER - I L-T-P-C: 3-0-0-3

Course Objectives:

The objectives of this course are to

- Conceptualize the basics of organizational and architectural issues of a digital computer.
- Learn the function of each element of a memory hierarchy.
- Study various data transfer techniques in digital computer.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Understand the basic organization of computer and different instruction formats and addressing modes	**K2**
CO2	Analyze the concept of pipelining, segment registers and pin diagram of CPU.	**K4**
CO3	Understand and analyze various issues related to memory hierarchy	**K2**
CO4	Evaluate various modes of data transfer between CPU and I/O devices	**K5**
CO5	Examine various inter connection structures of multi processors	**K4**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1						2	3		
CO2		1	1				2	2		
CO3	1	2								
CO4	2	2	1					2		
CO5		1					1	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Structure Of Computers: Computer Types, Functional units, Basic Operational concepts, Bus structures, Software, Performance, multiprocessor and multi computers, Historical perspective.

UNIT II:

Machine Instructions and Programs: Numbers, Arithmetic Operations, and c Characters, Memory locations and addresses, Memory operations, Instructions and Instruction sequencing, Addressing Modes, Assembly Languages, stacks and Queues Basic Input/output Operations, role of Stacks and Queues Additional Instructions

UNIT III:

Input/ Output Organization: Accessing I/O Devices, Interrupts, Processor examples, Direct Memory Access, Buses, Interface Circuits, and Standard I/O Interfaces

UNIT IV:

The Memory Systems: Some Basic concepts, Semi conductor RAM memories, Memory System Consideration, Read-Only Memories**,** Speed, Size, and cost, Cache Memories, Performance considerations, Virtual Memories, Memory Management Requirements, Secondary Storage

UNIT V:

Parallel Processing: Basic concepts, Pipeline Processors, Multiprocessors

Text Books:

1. Computer Organization, Carl Hamacher, Zvonks Vranesic, Safea Zaky, 5th Edition, McGraw Hill.
2. Computer Architecture and Organization , John P. Hayes ,3rd Edition, McGraw Hill

Reference Books:

1. Computer Organization and Architecture, William Stallings Sixth Edition, Pearson/PHI
2. Structured Computer Organization, Andrew S. Tanenbaum, 4th Edition PHI/Pearson
3. Fundamentals or Computer Organization and Design, Sivarama Dandamudi Springer Int. Edition.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Database Management Systems

SEMESTER - I **L-T-P-C: 3-0-0-3**

Course Objectives:

This Course will enable students to

- Explain the concept of databases, database management systems, database structures and how they work.
- Make use of Entity-Relationship Modeling and Relational Modeling for creating simple databases from the real world scenarios.
- Write relational algebra and structured query language (SQL) statements.
- Normalize a database using Normalization Rules.
- Discuss the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing

Course Outcomes(COs): At the end of the course the student will be able to:

CO	Course Outcomes	Knowledge Level (K)#
CO1	Illustrate the concept of databases, database management systems, database languages, database structures and their work	K2
CO2	Apply ER modeling and Relational modeling for designing simple databases.	K3
CO3	Summarize the concepts related to relational model and SQL and Write database queries using relational algebra and structured query language.	K2
CO4	Design and develop databases from the real world by applying the concepts of Normalization.	K6
CO5	Outline the issues associated with Transaction Management and Recovery, Tree Structured and Hash-Based Indexing	K2

#Based on suggested revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	1	1		1	2			
CO2	1	2	3				2	3		
CO3	2	3	3	2		2	2	2		
CO4	2	3	3	2		2	3	3	3	
CO5	2	2	3	2		2	3	3		

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit-I:

Introduction to Databases: Introduction, An Example, Characteristics of the Database Approach, Actors on Scene, Workers behind the scene,

Advantages of Using the DBMS Approach, A Brief History of Database Applications, When Not to Use a DBMS **[TB-3]**
Overview of Database Languages and Architectures: Data Models, Schemas and Instances, Three-Schema Architecture and Data Independence, Database Languages and Interfaces, The Database System Environment, Centralized and Client/Server Architecture for DBMSs, Classification of Database Management Systems **[TB-3]**

Unit-II:
Introduction to Database Design: Database Design and ER Diagrams, Entities, Attributes and Entity Sets, Relationships and Relationship Sets, Additional Features of the ER Model, Conceptual Design with the ER Model, Conceptual Design for Large Enterprises **[TB-1]**
Relational Model: Introduction to the Relational Model, Integrity Constraints over Relations, Enforcing Integrity Constraints, Querying Relational Data, Logical Database Design: ER to Relational, Introduction to Views, Destroying/Altering Tables and Views **[TB-1]**

Unit-III:
Relational Algebra: Selection and Projection, Set Operations, Renaming, Joins, Division, More Examples of Algebra Queries **[TB-1]**
SQL: Queries, Constraints, Triggers: The Form of a Basic SQL Query, UNION, INTERSECT and EXCEPT, Nested Queries, Aggregate Operators, Null Values, Complex Integrity Constraints in SQL, Triggers and Active Databases, Designing Active Databases **[TB-1]**

Unit-IV:
Introduction to Normalization Using Functional and Multivalued Dependencies:
Informal Design Guidelines for Relation Schema, Functional Dependencies, Normal Forms Based on Primary Keys, General Definitions of Second and Third Normal Forms, Boyce-Codd Normal Form, Multivalued Dependency and Fourth Normal Form, Join Dependencies and Fifth Normal Form **[TB-3]**

Unit-V:
Transaction Management and Concurrency Control: Transaction Concept, A Simple Transaction Model, Storage Structure, ACID Properties, Serializability, Transaction Isolation Levels, Concurrency Control, Lock-Based Protocols, Validation-Based Protocols **[TB-2]**
Note: For Practical Examples Please Go Through Reference 1

Text Books:
1. Data base Management Systems, 3/e, Raghurama Krishnan, Johannes Gehrke, Mc Graw-Hill
2. Data base System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S.
   Sudarshan, Mc Graw-Hill
3. Database Systems, 6/e Ramez Elmasri, Shamkant B. Navathe, Pearson

Reference Books:
1. Database Systems, 9/e, Carlos Coronel, Steven Morris, Peter Rob, Cengage
2. Introduction to Database Systems, 8/e, C J Date, Pearson

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Operating Systems

SEMESTER - I **L-T-P-C: 3-0-0-3**

Course Objectives:

This course enables the student to

- Introduce different types of operating systems.
- Learn process management techniques.
- Learn various memory management techniques.
- Introduce the architecture of Linux operating system.
- Learn multiple operating system like Unix and Windows.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Understand the basics of operating systems like kernel, shell, types and views of operating systems	**K2**
CO2	Understands CPU scheduling algorithms and compare the results using Gantt chart.	**K5**
CO3	Explain various memory management techniques and concept of thrashing	**K2**
CO4	Apply disk scheduling algorithms for better utilization of external memory	**K3**
CO5	Understand the architecture of UNIX operating system	**K1**
CO6	Write and execute shell programs	**K1**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	1								
CO2	1	2	3		2		2			
CO3	2	2	3			1		1		
CO4	1	2	3					2	2	
CO5	1		2							
CO6	2	3	2				2	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Introduction to Operating System Concept: Types of Operating Systems, Operating Systems Concepts, Operating System Operations. Operating Systems Structures- Operating System Services, User Operating-System Interface, Introduction to System calls, Types of System Calls.

UNIT–II:

Process Management: Process concept, Process State Diagram, Process control block, Process Scheduling, Inter process Communication, Threads-Threading Issues, Scheduling- Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

UNIT-III:

Process Synchronization: The Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, **Principles of deadlock:** System Model, Deadlock characterization, Deadlock handling, Deadlock Prevention, Detection and Avoidance, Recovery Starvation, Critical Regions form Deadlock

UNIT–IV:

Memory Management: Swapping, Contiguous Memory Allocation, Paging, structure of the Page Table, Segmentation Virtual Memory Management-Demand Paging, Page-Replacement Algorithms, Thrashing. **File-System Interface:** File Concept, Access Methods, Directory structure, File-System mounting, Files Sharing, Protection. File-System implementation- File-System Structure, Allocation Methods, Free-Space Management, Disk Structure, Disk Scheduling

UNIT-V:

Case Studies: Linux System: Design Principles, kernel Modules, Process Management, File Systems, Input and Output, Interprocess Communication, Network Structure, Security. **Windows7:** Design Principles, System Components, Terminal Services and Fast User, File System, Networking, Programmer Interface.

Text Books:

1. Operating system concepts, Abraham Siliberschatz, Peter Baer Galvin, Greg Gagne, John Wiley & Sons, Inc., Edition 9, 2011
2. Introduction to UNIX and Shell Programming, M. G. Venkateshmurhty, Pearson, 2005
3. UNIX & Shell Programming by B.M. Harwani, OXFORD University Press, 2013

Reference Books:

1. Advanced Programming in the UNIX Environment by W. Richard Stevens, Stephen Rago, Wesley Professional, 2013
2. UNIX Network Programming by W. Richard Stevens, 1990
3. Operating systems, William stallings, PHI/Pearson, 6/E, 2009
4. Operating systems, Dietal, Dietal, Pearson, 3/e, 2007
5. Operating systems, Dhamdhere, TMH, 2/e, 2009

Web Reference:

https://onlinecourses.swayam2.ac.in/cec20_cs06/preview

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Mathematical and Statistical Foundations

SEMESTER - I L-T-P-C: 3-1-0-4

Course Objectives: This course is aimed at enabling the students to

- To understand the mathematical fundamentals that is prerequisites for variety of courses like Data mining, Network protocols, analysis of Web traffic, Computer security, Software engineering, Computer architecture, operating systems, distributed systems bioinformatics, Machine learning.
- To develop the understanding of the mathematical and logical basis to many modern techniques in computer science technology like machine learning, programming language design, and concurrency.
- To study various sampling and classification problems.

Course Outcomes:

After the completion of the course, student will be able to

CO	Course Outcomes	Knowledge Level (K)#
CO1	To apply the basic rules and theorems of probability theory such as Baye's Theorem, to determine probabilities that help to solve engineering problems and to determine the expectation and variance of a random variable from its distribution.	**K3**
CO2	Able to perform and analyze of sampling, means, proportions, variances and estimates the maximum likelihood based on population parameters.	**K4**
CO3	To learn how to formulate and test hypotheses about sample means, variances and proportions and to draw conclusions based on the results of statistical tests.	**K6**
CO4	Design various ciphers using number theory.	**K6**
CO5	Apply graph theory for real time problems like network routing problem.	**K3**

# based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	2	1		2		2	2	3	
CO2	3	2	3		1		1	2	1	
CO3	3	2	1	2	3					
CO4	2	3	1				1			
CO5	3	2	2							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Probability and Random Variables: Random Experiments, Sample Spaces Events, the Concept of Probability the Axioms of Probability, Some Important Theorems on Probability Assignment of Probabilities, Conditional Probability Theorems on Conditional Probability, Independent Events, Bayes Theorem or Rule. Random Variables, Discrete Probability Distributions, Distribution Functions for Random Variables, Distribution Functions for Discrete Random Variables, Continuous Random Variables

UNIT II:

Sampling and Estimation Theory: Population and Sample, Statistical Inference Sampling With and Without Replacement Random Samples, Random Numbers Population Parameters Sample Statistics Sampling Distributions, Frequency Distributions, Relative Frequency Distributions, Computation of Mean, Variance, and Moments for Grouped Data. Unbiased Estimates and Efficient Estimates Point Estimates and Interval Estimates. Reliability Confidence Interval Estimates of Population Parameters, Maximum Likelihood Estimates

UNIT III:

Tests of Hypothesis and Significance: Statistical Decisions Statistical Hypotheses. Null Hypotheses Tests of Hypotheses and Significance Type I and Type II Errors Level of Significance Tests Involving the Normal Distribution One-Tailed and Two-Tailed Tests P Value Special Tests of Significance for Large Samples Special Tests of Significance for Small Samples Relationship between Estimation Theory and Hypothesis Testing Operating Characteristic Curves. Power of a Test Quality Control Charts Fitting Theoretical Distributions to Sample Frequency Distributions, The Chi-Square Test for Goodness of Fit Contingency Tables Yates' Correction for Continuity Coefficient of Contingency.

UNIT IV:

Algebraic Structures and Number Theory: Algebraic Systems, Examples, General Properties, Semi Groups and Monoids, Homomorphism of Semi Groups and Monoids, Group, Subgroup, Abelian Group, Homomorphism, Isomorphism. Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic, Modular Arithmetic (Fermat's Theorem and Euler's Theorem)

UNIT V:

Graph Theory: Basic Concepts of Graphs, Sub graphs, Matrix Representation of Graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian and Hamiltonian Graphs, Multigraphs, Planar Graphs, Euler's Formula, Graph Colouring and Covering, Chromatic Number, Spanning Trees, Algorithms for Spanning Trees (Problems Only and Theorems withou*t* Proofs).

Text Books:

1. Foundation Mathematics for Computer Science, 1st Edition, John Vince, Springer,2015
2. Probability & Statistics, 3rd Edition, Murray R. Spiegel, John J. Schiller and
   R. Alu Srinivasan, Schaum's Outline Series, Tata McGraw-Hill Publishers, 2018
3. Probability and Statistics with Reliability,2nd Edition, K. Trivedi, Wiley, 2011
4. Discrete Mathematics and its Applications with Combinatorics and Graph Theory, 7th Edition, H. Rosen, Tata McGraw Hill, 2003

Reference Books:

1. Probability and Computing: Randomized Algorithms and Probabilistic Analysis,1st Edition, M. Mitzenmacher and E. Upfal,2005
2. Applied Combinatorics,6th Edition, Alan Tucker, Wiley,2012

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Data Base Management Systems Lab

SEMESTER - I **L-T-P-C: 0-0-3-1.5**

Course Objectives:

This Course will enable students to

- Populate and query a database using SQL DDL/DML Commands.
- Declare and enforce integrity constraints on a database
- Writing Queries using advanced concepts of SQL
- Programming PL/SQL including procedures, functions, cursors and triggers

Course Outcomes:

At the end of the course the student will be able to:

CO	Course Outcomes	Knowledge Level (K)#
CO1	Utilize SQL to execute queries for creating database and performing data manipulation operations	K3
CO2	Examine integrity constraints to build efficient databases	K4
CO3	Apply Queries using Advanced Concepts of SQL	K3
CO4	Build PL/SQL programs including stored procedures, functions, cursors and triggers.	K6

#Based on suggested revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	2	3				1	1		
CO2	1	2	3	1			2			
CO3	1	2	3				2	3		
CO4		3	3				3	3		

(Levels of Correlation: 1-low, 2-medium 3-high)

1. Execute all DDL, DML and DCL commands on sample tables.
2. Implementation of different types of operators and built-in functions with suitable examples
3. Implementation of different types of joins with suitable examples
4. Create views, partitions, Sequence, Indexes and locks for a particular DB
5. Implement different types of constraints on relations.
6. Implementation of sub queries and nested queries.
7. Implement Queries on Group By & Having Clauses, ALIAS, Sequence By, Order By
8. Control Structure
   a) Write a PL/SQL block for Addition of Two Numbers
   b) Write a PL/SQL block for IF, IF and else condition
   c) Write a PL/SQL block for implementation of loops
   d) Write a PL/SQL block for greatest of three numbers using IF ANDELSEIF

9. Exception Handling- Implement the following with respect to exception handling.

Raising Exceptions, User Defined Exceptions, Pre-Defined Exceptions

10. Write PL/SQL block for an application using exception handling

10. *Procedures*

   a) Write a PL/SQL Procedure using Positional Parameters
   b) Write a PL/SQL Procedure using notational parameters
   c) Write a PL/SQL Procedure for GCD Numbers
   d) Write a PL/SQL Procedures for cursor implementation (explicit and implicit cursors)

11. Functions:

   a) Write a PL/SQL block to implement factorial using functions
   b) Write a PL/SQL function to search an address from the given database

12. Write a DBMS program to prepare Pl/SQL reports for an application using functions.

13. Triggers:

   a) Write a Trigger to pop-up the DML operations
   b) Write a Trigger to check the age valid or not Using Message Alert.
   c) Create a Trigger to Raise appropriate error code and error message.
   d) Create a Trigger on a table so that it will update another table while inserting values

14. Write PL/SQL block for an application using cursors and all types of triggers.

15. Write a PL/SQL block for transaction operations of a typical application using package

Text Books / Suggested Readings:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, “Introduction to SQL”, Fourth Edition, Pearson Education, 2007

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Programming & Data Structures Lab

SEMESTER - I **L-T-P-C: 0-0-4-2**

Course Objectives: This Course will enable students to

- Design and implement various data structures.
- Implement operations like searching, insertion, and deletion, traversing mechanism
- Develop applications using data structure algorithms.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement various basic data structures and its operations.	**K2**
CO2	Apply sorting and searching algorithms to given numbers	**K3**
CO3	Implement various tree operations.	**K2**
CO4	Implement various graphs algorithms.	**K2**
CO5	Develop applications using various data structures.	**K6**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2	3	1				1	1		
CO2	2	3	2				2	2		
CO3	2	3	3			2	1	2	2	
CO4	2	2	3			2	1	2	2	
CO5	2	3	3	1		2	3	2	1	

(Levels of Correlation: 1-low, 2-medium 3-high)

Experiment 1:

a) Write a program in C to display the n terms of even natural number and their sum.
b) Write a program in C to display the n terms of harmonic series and their sum. 1 + 1/2 + 1/3 + 1/4 + 1/5 … 1/n terms.
c) Write a C program to check whether a given number is an Armstrong number or not.
d) Write a C program to calculate the factorial of a given number.

Experiment 2:

a) Write a program in C for multiplication of two square Matrices.
b) Write a program in C to find transpose of a given matrix.

Experiment 3:

a) Write a program in C to check whether a number is a prime number or not using the function.
b) Write recursive program which computes the n^{th} Fibonacci number, for appropriate values of n.

c) Write a program in C to add numbers using call by reference.

Experiment 4:

a) Write a program in C to append multiple lines at the end of a text file.
b) Write a program in C to copy a file in another name.

Experiment 5:

Write recursive program for the following

a) Write recursive and non recursive C program for calculation of Factorial of an integer.
b) Write recursive and non recursive C program for calculation of GCD (n, m)
c) Write recursive and non recursive C program for Towers of Hanoi: N disks are to be transferred from peg S to peg D with Peg I as the intermediate peg.

Experiment 6:

a) Write C program that use both recursive and non recursive functions to perform Linear search for a Key value in a given list.
b) Write C program that use both recursive and non recursive functions to perform Binary search for a Key value in a given list.

Experiment 7:

a) Write C program that implement stack (its operations) using arrays.
b) Write C program that implement stack (its operations) using Linked list.

Experiment 8:

a) Write a C program that uses Stack operations to convert infix expression into postfix expression.
a) Write C program that implement Queue (its operations) using arrays.
b) Write C program that implement Queue (its operations) using linked lists.

Experiment 9:

Write a C program that uses functions to create a singly linked list and perform various operations on it.

Experiment 10:

Write a C program to store a polynomial expression in memory using linked list and perform polynomial addition.

Experiment 11:

a) Write a recursive C program for traversing a binary tree in preorder, inorder and postorder.
b) Write a non recursive C program for traversing a binary tree in preorder, inorder and postorder.

Experiment 12:

a) Write a C program to implement Prims' algorithm.
b) Write a C program to implement Kruskal's algorithm.

Experiment 13:

Implementation of Hash table using double hashing as collision resolution function.

Experiment 14:

Implementation of Binary Search trees- Insertion and deletion.

Experiment 15:

Implementation of AVL Tree – Insertion and Deletion

Experiment 16:

a) Write C program that implement Bubble sort, to sort a given list of integers in ascending order.

b) Write C program that implement Quick sort, to sort a given list of integers in ascending order.

c) Write C program that implement merge sort, to sort a given list of integers in ascending order

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Operating Systems Lab

SEMESTER - I L-T-P-C: 0-0-3-1.5

Course Objectives:

This Course will enable students to implement CPU scheduling algorithms, Disk scheduling algorithms, Execute different types of Linux commands and Write shell scripts

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement various CPU scheduling algorithms and compare results	**K5**
CO2	Implement various disk scheduling algorithms and compare results	**K5**
CO3	Implement page replace algorithms	**K2**
CO4	Implement various memory management techniques.	**K2**
CO5	Execute basic Linux commands	**K1**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	2	3		2		2	3	2	
CO2	3	2	3					2	2	
CO3	2	2	3			1		2	1	
CO4	2	2	3			1		1		
CO5	2	3	2				2	2		

(Levels of Correlation: 1-low, 2-medium 3-high)

List of Experiments:

UNIX Lab- Introduction to Unix

1. Study of Unix/Linux general purpose utility commands
2. Study of Bash shell, Bourne shell and C shell in Unix/Linux operating system .
3. Study of UNIX/LINUX File System(tree structure).
4. C program to emulate the UNIX ls –l command
5. C program that illustrates how to execute two commands concurrently with a command pipe. Ex: - ls –l | sort
6. Multiprogramming-Memory management-Implementation of fork (), wait (), exec() and exit (), System calls

Operating Systems Lab

1. Simulate the Following CPU Scheduling Algorithms
   A) FCFS B) SJF C) Priority D) Round Robin
2. Multiprogramming-Memory Management- Implementation of fork(), wait(), exec() and exit()

3. Simulate The Following
   a. Multiprogramming with A Fixed Number Of Tasks (MFT
   b. Multiprogramming with A Variable Number Of Tasks (MVT)
4. Write a program to implement first fit, best fit and worst fit algorithm for memory management.
5. Simulate Bankers Algorithm for Dead Lock Avoidance
6. Simulate Bankers Algorithm for Dead Lock Prevention.
7. Simulate The Following Page Replacement Algorithms.
   a) FIFO b) LRU c) LFU
8. Simulate the Following File Allocation Strategies
   a) Sequenced b) Indexed c) Linked

Linux Lab

1. Write a Shell program to check whether given number is prime or not.
2. Write a shell script which will display Fibonacci series up to the given range.
3. Write a shell script to check whether the given number is Armstrong or not.
4. Write a shell script to the calculate the value of
5. Write a shell script to accept student number, name, marks in 5 subjects.
6. Find total, average and grade using the following rules:
   Avg>=80 then grade A
   Avg<80&&Avg>=70 then grade B
   Avg<70&&Avg>=60 then grade C
   Avg<60&&Avg>=50 then grade D
   Avg<50&&Avg>=40 then grade E
7. Write a shell script to find minimum and maximum elements in the given list of elements.
8. Write a shell program to check whether the given string is palindrome or not.
9. Write an awk program to print sum, avg of students marks list
10. Write a shell script to compute no. of characters and words in each line of given file
11. Write a shell script to check whether the given input is a number or a string

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Computer Networks

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:
At the end of the course, the students will be able to:

- To Understands the fundamental concepts of computer networking and OSI Reference model.
- To Familiarize the student with the basic taxonomy and terminology of the computer networking area.
- To learn and understand the advanced networking concepts, preparing the student for entry advanced courses in computer networking.
- To develop and gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Explain the network architecture, TCP/IP and OSI reference models	**K2**
CO2	Identify and understand various techniques and modes of transmission	**K3**
CO3	Demonstrate the data link protocols, multi-channel access protocols and IEEE 802 standards for LAN	**K2**
CO4	Describe routing and congestion in network layer with routing algorithms and classify IPV4 addressing scheme	**K5**
CO5	Discuss the elements and protocols of transport layer	**K6**
CO6	Develop network security and define various protocols such as FTP, HTTP, Telnet, DNS	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**P2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1		1			1			1		
CO2	1							2		1
CO3		1	1						2	
CO4	1				2					
CO5										
CO6		2						1		

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I

Introduction: Network Topologies WAN, LAN, MAN. Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models.**Physical Layer –**Introduction to physical layer-Data and Signals, Periodic analog signals, digital signals, transmission impairment, ,Data rate limits, performance - Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and Unguided media: Wireless-Radio waves, microwaves, infrared.

Unit-II

The Data Link Layer - Services Provided to the Network Layer – Framing – Error Control – Flow Control, Error Detection and Correction – Error-Correcting Codes – Error Detecting Codes. **Elementary Data Link Protocols-** A Utopian Simplex Protocol-A Simplex Stop and

Wait Protocol for an Error free channel-A Simplex Stop and Wait Protocol for a Noisy Channel, Sliding Window Protocols-A One Bit Sliding Window Protocol-A Protocol Using Go-Back-N- A Protocol Using Selective Repeat.

UNIT-III

The Medium Access Control Sub layer-The Channel Allocation Problem-Static Channel Allocation-Assumptions for Dynamic Channel Allocation, Multiple Access Protocols-Aloha-Pure aloha- slotted aloha-Carrier Sense Multiple Access Protocols- Collision-Free Protocols-Limited Contention Protocols. **Wireless LAN Protocols-** Ethernet-Classic Ethernet Physical Layer-Classic Ethernet MAC Sub-layer Protocol-Ethernet Performance-Fast Ethernet-Wireless LANs-The 802.11 Architecture and Protocol Stack-The 802.11 Physical Layer-The802.11 MAC Sub-layer Protocol- The 805.11 Frame Structure-Services.

Unit-IV

The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks, Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical. **Congestion Control algorithms-**General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing- Admission Control-Traffic Throttling-Load Shedding. **Internet Working:** How networks differ- How networks can be connected- Tunneling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-, IP addresses-, Subnets-IP Version 6-The main IPV6 header- Internet control protocols- ICMP-ARP-DHCP.

UNIT-V

The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection- windows in TCP- flow control-Error control. **Application Layer –-** World Wide Web: HTTP , FTP-Two connections-control connection-Data connection-security of FTP-Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging. **Domain Name System:** Name Space, DNS in Internet, - Resolution-Caching- Resource Records- DNS messages-Registrars-security of DNS Name Servers.

Text Books:

1. Computer Networks: Andrew S Tanenbaum David J. Wetherall, 5/e, Pearson
2. Data communications and networking: Behrouz Forouzan, 5/e, McGraw Hill

Reference Books

1. Computer Networks – A System Approach, Peterson, Bruce Davie,2/e , Harcourt Asia
2. Compute communications and networking technologies, Gallo, Hancock, Cengage

An Engineering approach to compute networking, Kesha, Pearson

Business English and Soft Skills

SEMESTER - II **L-T-P-C: 2-0-0-2**

INTRODUCTION:

English is an international language as well as a living and vibrant one. People have found that knowledge of English is a passport for better career, better pay, advanced knowledge and for communication with the entire world. As it is a language of opportunities in this global age, English is bound to expand its domain of use everywhere. The syllabus has been designed to enhance communication skills of the students of professional courses. The prescribed books serve the purpose of preparing them for everyday communication and to face the global competitions in future.

The prescribed books for detailed study focus on soft skills and Business English Communication. The teachers should encourage the students to use the target language in business scenario. The classes should be interactive and student-centered. They should be encouraged to participate in the classroom activities keenly. The teacher can bring variety by using authentic materials such as newspaper articles, advertisements, promotional material etc.

OBJECTIVES:

1. To enable the students to communicate in English for business purpose.
2. To enhance presentation skills of the students required to persuade the target customers.
3. To develop the business writing skills of the students
4. To enhance the soft skills of the students in business setting.
5. To improve employability and negotiation skills necessary in business world.

Unit I

Introduction to Essentials of Business English -Importance of Business English -Definition– Types &Channels of Business English – Cultural barriers in Business English. Business etiquettes& Manners.

Unit II

Delivering Effective speech- How to make first impression -Managing questions and answering. Speaking at meetings. Business Presentations– Body language-Developing self confidence.

Unit III

Business report writing - Business Letter Writing– Writing Effective Resume and covering letters

Unit IV

Goal setting-Time Management- Leadership Skills– Team Work - Team building- Art of Negotiation– Emotional Intelligence- Logical Thinking.

Unit V

Employability Skills- Interview Skills- Group Discussions–Video Conferencing

EXPECTED OUTCOME:

The students will be able to

- Define types of Business English and its barriers
- Discuss the different aspects of business communication
- Apply the Soft Skills in business scenario

- Analyze different ideas in Group Discussions
- Appraise ones strengths and weaknesses to project a good image in an interview
- Develop a resume suitable to the requirements of the job

Prescribed Books:

1. Soft Skills - Key to Success in Workplace and Life by Meenakshi Raman and Shalini Upadhyay, Cengage, 2018
2. Business English: A Complete Guide for All Business and Professional Communications by PREM P.BHALLA, V&S Publishers, 2018 edition

SUGGESTED READING:

1. Personality Develpment and Soft Skills by Barun K. Mitra, Oxford, 2016.
2. English Conversation Practice –Grant Taylor, Tata McGraw Hill,2009.
3. A Course in Communication Skills- Kiranmai Dutt & co. Foundation Books, 2012.
4. Writing with a Purpose, Tickoo and Sasi Kumar, OUP, 2011
5. Strengthen Your Writing, Orient Blackswan

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Object Oriented Programming

SEMESTER - II L-T-P-C: 3-0-0-3

Course Objectives:

- To understand the basic concepts of object oriented programming concepts.
- To introduce the principles of inheritance and polymorphism and demonstrate how they are related to the design of abstract classes
- To understand the implementation of packages and interfaces
- To introduce the concept of multithreading and exception handling
- To learn and understand the design of Graphical User Interface using applets and swing controls

Course Outcomes (COs): At the end of the course, student will be able to

	Course Outcomes	**Knowledge Level (K)#**
CO1	Describe the uses OOP concepts	**K2**
CO2	Apply OOP concepts to solve real world problems	**K3**
CO3	Distinguish the concept of packages and interfaces	**K4**
CO4	Demonstrate the exception handing, multithread applications with synchronization	**K2**
CO5	Design the GUI based applications using AWT and Swings	**K6**
CO6	Discuss the Collection Framework	**K6**

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	1				1					
CO2		2						1	1	
CO3	1									
CO4		2	1						1	
CO5	1					1				

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Basics of Object Oriented Programming (OOP): Need for OO paradigm , A way of viewing world- Agents, responsibility, messages, methods, classes and instances, class hierarchies (Inheritance), method binding, overriding and exceptions, summary of OOP concepts, coping with complexity, abstraction mechanisms. **Java Basics:** Data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and costing, simple java program, classes and objects- concepts of classes, objects, constructors methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, string handling.

UNIT–II:

Inheritance: Hierarchical abstractions, Base class object, subclass, subtype, substitutability, forms of inheritance- specialization, specification, construction, extension, limitation, combination, benefits of inheritance costs of inheritance. Member access rules, super uses, using final with inheritance, polymorphism, abstract classes. **Packages and Interfaces:** Defining, Creating and Accessing a package, Understanding CLASSPATH, Importing packages, differences between classes and interfaces, defining an interface, Implementing interface, applying interfaces variables in interface and extending interfaces.

UNIT–III:

Exception handling and Multithreading: Concepts of exception handling, benefits of exception handling, Termination or presumptive models, exception hierarchy, usage of try, catch, throws and finally, built in exceptions, creating own exception sub classes. Differences between multi threading and multitasking, thread life cycle, creating threads, synchronizing threads, daemon threads, thread groups.

UNIT–IV:

Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes, inner classes. The AWT class hierarchy , user-interface components- labels, button, canvas, scrollbars, text components, check box, check box groups, choices, list panes- scroll pane, dialogs, menu bar, graphics, layout manager- layout manager types- boarder, grid, flow, card and grid bag.

UNIT-V:

Applets: Concepts of Applets, differences between applets and applications, lifecycle of an applet, types of applets, creating applets, passing parameters to applets, **Swings:** Introduction, limitations of AWT, MVC architecture, components, containers, exploring swing- JApplet, JFrame and JComponent, Icons and Labels, text fields, buttons-The JButton class, Check boxes, Radio Buttons, Combo boxes, Tabbed panes, Scroll panes, Trees and Tables.

Text Books:

1. Java-The complete reference,7/e, Herbert schildt, TMH
2. JAVA: How to program, 8/e, Dietal , Dietal,PHI
3. Introduction of programming with JAVA,S.Dean,TMH
4. Introduction to Java programming, 6/e, Y.Daniel Liang, Pearson

Reference Books:

1. Core Java 2, Vol 1(Vol 2) Fundamentals(Advanced), 7/e, Cay.S.Horstmann, Gary Cornell, Pearson

Big Java2,3/e, Cay.S. Horstmann, Wiley

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Software Engineering

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To understand the nature of software development and software life cycle models.
- To understand methods of capturing, specifying, visualizing and analyzing software requirements.
- To know basics of testing and understanding concept of software quality assurance and software configuration management process.
- To learn to provide correctness proofs for algorithms.

Course Outcomes (COs): At the end of the course, student will be able to

	Course Outcomes	Knowledge Level (K)#
CO1	Analyze software application domains and process models used in software development.	**K4**
CO2	Explain the software requirements collection and develop specifications and evaluate them.	**K2**
CO3	Convert the requirements model into the design model and evaluate the complexity metrics.	**K5**
CO4	Compare various testing strategies and tactics and their applications with the supporting tools.	**K4**
CO5	Adopt the activities of Software Project Development principles in project development.	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		2								
CO2	1	1			1					
CO3		1					1			
CO4			3						1	1
CO5				2						1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Introduction: Software Engineering and its history, Software crisis, Evolving of a Programming System Product, Characteristics of Software, Brooks' No Silver Bullet, and Software Myths. **Software Development Life Cycles :** Software Development Process, The Code-and-Fix model, The Waterfall model, The Evolutionary Model, The Incremental Implementation, Prototyping, The Spiral Model, Software Reuse, Critical Comparisons of SDLC models. **An Introduction to Non-Traditional Software Development Process:** Rational Unified Process, Rapid Application Development, Agile Development Process.

UNIT–II:

Requirements: Importance of Requirement Analysis, User Needs, Software Features and Software Requirements. **Classes of User Requirements :** Enduring and Volatile, Sub phases of Requirement Analysis, Functional and Non-functional requirements, Barriers to Eliciting User requirements, The software requirements document and SRS standards, Requirements Engineering, Case Study of SRS for a Real Time System. **Tools for Requirements Gathering:** Document Flow Chart, Decision Table, Decision Tree, Introduction to non-traditional Requirements.

UNIT- III:

Software Design: Goals of good software design, Design strategies and methodologies, Data oriented software design. **Structured Design:** Structure chart, Coupling, Cohesion, Modular structure, Packaging, Object oriented design, Top-down and bottom-up approach, Design patterns. **Structured Analysis:** DFD, Data Dictionary, Software Measurement and Metrics : Various Size Oriented Measures : Halstead's software science, Function Point (FP) based measures, Cyclomatic Complexity Measures : Control flow graphs Development : Selecting a language, Coding guidelines, Writing code, Code documentation.

UNIT– IV:

Software Testing : Testing process, Design of test cases, Functional Testing : Boundary value analysis, Equivalence class testing, Decision table testing, Cause effect graphing, Structural testing, Path testing, Data flow and mutation testing, Unit testing, Integration and system testing, Debugging, Alpha & beta testing, testing tools & standards.

UNIT-V:

Software Maintenance: Management of maintenance, Maintenance process, Maintenance models, Regression testing, Reverse engineering, Software reengineering, Configuration management, documentation.

Text Books:

1. Software Engineering: A Practitioner's Approach by R. S. Pressman, McGraw Hill, 9th Edition, Sept 2019

Reference Books:

1. Zero Defect Software, G. G. Schulmeyer, Published by McGraw Hill, 1992
2. Object Oriented Modeling and Design, J. Rumbaugh, Published by Prentice Hall, 1991
3. Software Engineering K.K. Aggarwal, Yogesh Singh, Published by New Age International Publishers, Third Edition, 2007
4. Software Engineering , Ian Sommerville, Published by Addison Welsley, 9th Edition, 2010.

1. An Integrated Approach to Software Engineering, Pankaj Jalote, Published by Narosa Publishing House, 3rd Edition, 2007Data base System Concepts, 6/e, Abraham Silberschatz, Henry F. Korth, S.
   Sudarshan, Mc Graw-Hill
2. Database Systems, 6/e Ramez Elmasri, Shamkant B. Navathe, Pearson

Reference Books:

1. Database Systems, 9/e, Carlos Coronel, Steven Morris, Peter Rob, Cengage
2. Introduction to Database Systems, 8/e, C J Date, Pearson

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Design and Analysis of Algorithms

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To analyze the asymptotic performance of algorithms.
- To understand the write rigorous correctness proofs for algorithms.
- To familiarity with major algorithms and data structures.
- Apply important algorithmic design paradigms and methods of analysis.
- Synthesize efficient algorithms in common engineering design situations.

Course Outcomes (COs): At the end of the course, student will be able to

CO	Course Outcomes	Knowledge Level (K)#
CO1	Explain the basic concepts of time and space complexity,	**K2**
CO2	Explain the basic concepts of divide-and-conquer Strategy, dynamic programming, greedy and approximate algorithm	**K3**
CO3	Describe the methodologies of how to analyze the following applications by Dynamic Programming algorithm	**K5**
CO4	Discuss the concept of graph coloring and back tracking	**K6**
CO5	Analyze the performance of algorithms	**K4**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3				1			1		
CO2				1						
CO3							1			
CO4	1								1	
CO5							2			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction: Algorithm, Pseudo code for expressing algorithms, performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, probabilistic analysis, Amortized analysis. Disjoint Sets- disjoint set operations, union and find algorithms, spanning trees, connected components and bi- connected components.

UNIT-II:

Divide and conquer: General method, applications-Binary search, Quick sort, Merge sort, Stassen's matrix multiplication. Greedy method: General method, applications-Job sequencing with deadlines, 0/1 knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT-III:

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

UNIT-IV:

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph coloring, Hamiltonian cycles.

UNIT-V:

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution. NP-Hard and NP-Complete problems: Basic concepts, non deterministic algorithms, NP - Hard and NP Complete classes, Cook's theorem.

Text Books:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press
2. The Algorithm Design Manual, 2nd edition, Steven S. Skiena, Springer
3. Introduction to Algorithms, second edition, T.H.Cormen, C.E.Leiserson, R.L.Rivest and C.Stein, PHI Pvt. Ltd

Reference Books:

1. Introduction to the Design and Analysis of Algorithms, Anany Levitin, PEA
2. Design and Analysis of Algorithms, Pearson Education, Parag Himanshu Dave, Himansu Balachandra Dave
3. Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T. Lee, S.S.Tseng, R.C.Chang and T.Tsai, Mc GrawHill.
4. Design and Analysis of algorithms, Pearson education, Aho, Ullman and Hopcroft.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 Artificial Intelligence

SEMESTER - II **L-T-P-C: 3-0-0-3**

Course Objectives:

- To learn the basic State space representation. Intelligent Systems Categorization of Intelligent concepts and techniques of AI and machine learning
- To explore the various mechanism of Knowledge and Reasoning used for building expert system.
- To become familiar with supervised and unsupervised learning models
- To design and develop AI and machine learning solution using modern tools.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Demonstrate knowledge of the building blocks of AI as presented in terms of intelligent agents	**K6**
CO2	Analyze and formalize the problem as a state space, graph, design heuristics and select amongst different search or game based techniques to solve them.	**K5**
CO3	Develop intelligent algorithms for constraint satisfaction problems and also design intelligent systems for Game Playing	**K6**
CO4	Attain the capability to represent various real life problem domains using logic based techniques and use this to perform inference or planning.	**K1**
CO5	Solve problems with uncertain information using Bayesian approaches.	**K3**

- *#Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	2							1		
CO2					1	1		2		
CO3		2			1				1	1
CO4	2							1		
CO5					1		1			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT– I

Introduction to AI- Definition, Problem, System, Components of AI Program, Foundations of AI, Applications of AI, Current trends in AI, Intelligent Agents: Anatomy, structure, Types.

UNIT– II

Problem solving-Solving problem by Searching: Problem Solving Agent, Formulating Problems. Uninformed Search Methods: Breadth First Search (BFS), Depth First Search (DFS), Depth Limited Search, Depth First Iterative Deepening (DFID), Informed Search Methods- Greedy best first Search, A* Search, Memory bounded heuristic Search. Local Search Algorithms and Optimization Problems- Hill climbing search Simulated annealing and local beam search.

UNIT – III

Knowledge and Reasoning-Knowledge based Agents, The Wumpus World, and Propositional logic**. First Order Logic**- Syntax and Semantic, Inference in FOL, Forward chaining, backward Chaining, Knowledge Engineering in First-Order Logic, Unification and Resolution.

UNIT –IV

Agents: Definition of agents, Agent architectures (e.g., reactive, layered, cognitive), Multi-agent systems- Collaborating agents, Competitive agents, Swarm systems and biologically inspired models. Expert Systems: Representing and Using Domain Knowledge, Expert System Shells, Explanation, Knowledge Acquisition.

UNIT –V

Expert Systems: Architecture of expert systems, Roles of expert systems, Knowledge Acquisition, Meta knowledge, Heuristics. Typical expert systems- MYCIN, DART, XOON, Expert systems shells.

Text Books:

1. Artificial Intelligence, Saroj kaushik Published by Cengage Learning India, 2011
2. Artificial Intelligence and Machine Learning By Vinod Chandra S.S., Anand Hareendran S
3. Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach “Second Edition" Pearson Education

Reference Books:

1. Ivan Bratko "PROLOG Programming for Artificial Intelligence", Pearson Education, Third Edition.
2. Elaine Rich and Kevin Knight "Artificial Intelligence "Third Edition
3. Han Kamber, “Data Mining Concepts and Techniques”, Morgann Kaufmann Publishers.

G. Luger, W. A. Stubblefield, "Artificial Intelligence", Third Edition, AddisoWesley Longman, 1998.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 **Advanced Unix Programming**

SEMESTER - II L-T-P-C: 3-0-0-3

Course Objectives:

- To understand the fundamental design of the unix Programming
- To become fluent with the systems calls provided in the unix environment
- To be able to design and build an application/service over the unix operating system

Course Outcomes: At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Explain Networking Commands, File handling Utilities and shell script examples	**K3**
CO2	Discuss about Unix file structure ,directories and system calls	**K6**
CO3	Compare process and threads concepts	**K3**
CO4	How to Implementing client server program using pipes and FIFOs	**K1**
CO5	Demonstrate socket structure, socket system calls for connection oriented protocol and connectionless protocol.	**K5**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1									
CO2		1						1		
CO3						1				1
CO4			1				1			
CO5	1				2				1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I

Review of Unix Utilities and Shell Programming: -File handling utilities, security by file permissions, process utilities, disk utilities, networking commands, backup utilities, text processing utilities. **Shell Programming:** shell, shell responsibilities, pipes and input redirection, output redirection, here documents, the shell as a programming language, shell meta characters, shell variables, shell commands, the environment, control structures, shell script examples.

UNIT-II

Unix Files: Unix file structure, directories, files and devices, System calls, library functions, low level file access, usage of open, create, read, write, close, lseek, stat, fstat, octl, umask, dup, dup2, Differences between system call and library functions. File and directory maintenance: chmod, chown, unlink, link, symlink, mkdir, rmdir, chdir, getcwd. **Directory handling system calls:** opendir, readdir, closedir, rewinddir, seekdir,telldir

UNIT-III

Unix Process: Threads and Signals: process, process structure, starting new process, waiting for a process, zombie process, orphan process, process control, process identifiers, system call interface for process management, -fork, vfork, exit, wait, waitpid, exec, system.

Signals: Signal functions, unreliable signals, interrupted system calls, kill and raise functions, alarm, pause functions, abort, sleep functions.

UNIT-IV

Inter process Communication: Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes, FIFOs, message queues, semaphores and shared memory. Differences between pipes and FIFOs. Implementing client server program using pipes and FIFOs. **Message Queues-**: IPC, permission issues, Access permission modes, message structure, working with message queues, client/server example. **Semaphores:** Creating semaphore sets, Unix kernel support for semaphores, Unix APIs for semaphores, file locking using semaphores.

UNIT-V

Shared Memory: Working with a shared memory segments, Unix kernel support for shared memory, client/server example.**Sockets:** Berkeley sockets, socket structure, socket system calls for connection oriented protocol and connectionless protocol, implementing client server programs using TCP and UDP sockets.

Text books:

1. Advanced programming in the unix environment, w- Richard Stevens 2nd Edition Pearson education
2. Unix Concepts and Applications, 3/e, Sumitabha Das, TMH

Reference books:

1. Unix and shell Programming, Sumitabha Das,TMH
2. A Beginner's Guide to Unix, N.P.Gopalan, B.Sivaselva,PHI
3. Unix Shell Programming, Stephen G.Kochan, Patrick Wood, 3/e,Pearson

Unix Shell Programming, Lowell Jay Arthus & Ted Burns,3/e,GalGotia

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-1 **Data Warehousing and Data mining**
SEMESTER - II L-T-P-C: 3-0-0-3

Course Objectives:

- Be familiar with mathematical foundations of data mining tools..
- Understand and implement classical models and algorithms in data warehouses and data mining
- Characterize the kinds of patterns that can be discovered by association rule mining, classification and clustering.
- Develop skill in selecting the appropriate data mining algorithm for solving practical problems.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	An ability to understand the basics of types of data, quality of data, suitable techniques required for preprocessing and measures required to perform data analysis	**K2**
CO2	Describe the need of classification, identify suitable technique(s) to perform classification, model building and evaluation	**K3**
CO3	Identify the requirements and usage of association rule mining on categorical and continuous data.	**K3**
CO4	Compare and Identify suitable clustering algorithm(s) (apply with open source tools), interpret, evaluate and report the result	**K4**
CO5	Describe the requirements and the need of web mining	**K2**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1				1					1
CO2	1		1							
CO3								3		
CO4			3							
CO5	1						2			

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-1:
Introduction to Data mining, types of Data, Data Quality, Data Processing, Measures of Similarity and Dissimilarity, Exploring Data: Data Set, Summary Statistics, Visualization, Data Warehouse, OLAP and multi dimensional data analysis.

UNIT-II:
Classification: Basic Concepts, Decision Trees and model evaluation: General approach for solving a classification problem, Decision Tree induction, Model over fitting: due to presence of noise, due to lack of representation samples, Evaluating the performance of classifier. Nearest Neighborhood classifier, Bayesian Classifier, Support vector Machines: Linear SVM, Separable and Non Separable case.

UNIT-III:
Association Analysis: Problem Definition, Frequent Item-set generation, rule generation, compact representation of frequent item sets, FP-Growth Algorithms. Handling Categorical, Continuous attributes, Concept hierarchy, Sequential, Sub graph patterns

UNIT-IV:

Clustering: Over view, K-means, Agglomerative Hierarchical clustering, DBSCAN, Cluster evaluation: overview, Unsupervised Cluster Evaluation using cohesion and separation, using proximity matrix, Scalable Clustering algorithm

UNIT-V:

Web data mining: Introduction, Web terminology and characteristics, Web content mining, Web usage mining, web structure mining, Search Engines: Characteristics, Functionality, Architecture, Ranking of WebPages, Enterprise search

Text Books:

1. Introduction to Data Mining, Tan, Steinbach and Vipin Kumar, Pearson Education, 2016
2. Data Mining:Concepts and Techniques, 2nd Edition, Jiawei Han and MichelineKamber, ELSEVIER

Reference Books:

1. Data Mining: The Textbook, Springer, May 2015, Charu C. Aggarwal.

Web resources:

1. NPTEL: https://nptel.ac.in/courses/106/105/106105174/

https://www.saedsayad.com/data_mining.htm

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Object Oriented Programming Lab

SEMESTER - II **L-T-P-C: 0-0-2-1**

Course Objectives:

- To understand how to design, implement, test, debug, and document programs that use basic data types and computation, simple I/O, conditional and control structures, string handling and functions.
- To understand the importance of Classes & objects along with constructors, Arrays and Vectors.
- Discuss the principles of inheritance, interface and packages and demonstrate though problem analysis assignments how they relate to the design of methods, abstract classes and interfaces and packages.
- To understand importance of Multi-threading & different exception handling mechanisms.
- To learn experience of designing, implementing, testing, and debugging graphical user interfaces in Java using applet and AWT that respond to different user events.
- To understand Java Swings for designing GUI applications based on MVC architecture

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Apply OOP concepts to solve real world problems	**K2**
CO2	Implement different forms of inheritance	**K3**
CO3	Create packages and to reuse them.	**K6**
CO4	Implement multi threaded programs using synchronization concepts	**K3**
CO5	Create user defined exceptions	**K6**
CO6	Design GUI applications using AWT and SWINGS.	**K6**

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3	1							1	
CO2			1		1					
CO3							1			
CO4			1		1					
CO5					1					
CO6		2							1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

List of Experiments:

1. The Fibonacci sequence is defined by the following rule. The first 2 values in the sequence are 1, 1. Every subsequent value is the sum of the 2 values preceding it. Write a Java Program that uses both recursive and non recursive functions to print the nth value of the Fibonacci sequence.
2. Write a Java Program that prompts the user for an integer and then prints out all the prime numbers up to that Integer.

3. Write a Java Program that checks whether a given string is a palindrome or not. Ex. MALAYALAM is a palindrome.
4. Write a Java Program for sorting a given list of names in ascending order.
5. Write a Java Program that illustrates how runtime polymorphism is achieved.
6. Write a Java Program to create and demonstrate packages.
7. Write a Java Program, using StringTokenizer class, which reads a line of integers and then displays each integer and the sum of all integers.
8. Write a Java Program that reads on file name form the user then displays information about whether the file exists, whether the file is readable/ writable, the type of file and the length of the file in bytes and display the content of the using File Input Stream class.
9. Write a Java Program that displays the number of characters, lines and words in a text/text file.
10. Write an Applet that displays the content of a file.
11. Write a Java Program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the +-*?% operations. Add a text field to display the result.
12. Write a Java Program for handling mouse events.
13. Write a Java Program demonstrating the life cycle of a thread.
14. Write a Java Program that lets users create Pie charts. Design your own user interface (with Swings & AWT).

Write a Java Program to implement a Queue, using user defined Exception Handling (also make use of throw, throws).

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
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.
10. A Practical guide to Data structures and Algorithms using Java,Goldman&Goldman, Chapman and Hall/CRC,Taylor and Francis Group.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Software Engineering Lab

1. Write down the problem statement for a suggested system of relevance.
2. Do requirement analysis and develop Software Requirement Specification Sheet (SRS) for suggested system.
3. To perform the function oriented diagram: Data Flow Diagram (DFD) and Structured chart.
4. To perform the user‘s view analysis for the suggested system: Use case diagram.
5. To draw the structural view diagram for the system: Class diagram, object diagram.
6. To draw the behavioral view diagram : State-chart diagram, Activity diagram
7. To perform the behavioral view diagram for the suggested system : Sequence diagram, Collaboration diagram
8. To perform the implementation view diagram: Component diagram for the system.
9. To perform the environmental view diagram: Deployment diagram for the system.
10. To perform various testing using the testing tool unit testing, integration testing for a sample code of the suggested system.
11. Perform Estimation of effort using FP Estimation for chosen system.
12. To Prepare time line chart/Gantt Chart/PERT Chart for selected software project.

Text Books: 1. K.K. Aggarwal & Yogesh Singh, ―Software Engineering‖, New Age International, 2005 2. Pankaj Jalote, ―An Integrated Approach to Software Engineering‖, Second Edition, Springer

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Machine Learning

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

From the course the student will learn

- To learn patterns and concepts from data without being explicitly programmed in various IOT nodes.
- To design and analyze various machine learning algorithms and techniques with a modern outlook focusing on recent advances.
- Explore supervised and unsupervised learning paradigms of machine learning.
- To explore Deep learning technique and various feature extraction strategies.

Course Outcomes(CO's): At the end of the course, student will be able to

	Course Outcomes	**Knowledge Level (K)#**
CO1	Illustrate and comprehend the basics of Machine Learning with Python	**K2**
CO2	Demonstrate the algorithms of Supervised Learning and be able to differentiate linear and logistic regressions	**K2**
CO3	Demonstrate the algorithms of Unsupervised Learning and be able to understand the clustering algorithms	**K2**
CO4	Evaluate the concepts of binning, pipeline Interfaces with examples	**K5**
CO5	Apply the sentiment analysis for various case studies	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	1		2							
CO2					1	1				
CO3							2			
CO4							1			
CO5	1				1				2	

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction to Machine Learning with Python: Introduction to Machine Learning, basic terminology, Types of Machine Learning and Applications, Using Python for Machine Learning: Installing Python and packages from the Python Package Index, Introduction to NumPy, SciPy, matplotlib and scikit-learn, Tiny application of Machine Learning.

UNIT-II:

Supervised Learning: Types of Supervised Learning, Supervised Machine Learning Algorithms: k-Nearest Neighbors, Linear Models, Naive Bayes Classifiers, Decision Trees, Ensembles of Decision Trees, Kernelized Support Vector Machines, Uncertainty Estimates from Classifiers.

UNIT-III:

Unsupervised Learning: Types of Unsupervised Learning, challenges, Preprocessing and scaling, Dimensionality Reduction, Feature Extraction, Manifold Learning, Clustering: K-Means Clustering, Agglomerative Clustering, DBSCAN, Comparing and Evaluating Clustering Algorithms.

UNIT-IV:

Representing Data and Engineering Features: Categorical Variables, Binning, Discretization, Linear Models, Trees, Interactions and Polynomials, Univariate Nonlinear Transformations, Automatic Feature Selection. Parameter Selection with Preprocessing, Building Pipelines, The General Pipeline Interface

UNIT-V:

Working with Text Data (Data Visualization) : Types of Data Represented as Strings, Example Application: Sentiment Analysis of Movie Reviews, Representing Text Data as a Bag of Words, Stop Words, Rescaling the Data with tf-idf, Investigating Model Coefficients, Approaching a Machine Learning Problem, Testing Production Systems, Ranking, Recommender Systems and Other kinds of Learning.

Text Books:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists, Andreas C. Muller & Sarah Guido, Orielly Publications, 2019.
2. Python Machine Learning, Sebastian Raschka & Vahid Mirjalili, 3rd Edition, 2019.
3. Building Machine Learning Systems with Python, Luis Pedro Coelho, Willi Richert, 2nd Edition, 2015.

Reference Books:

1. Machine Learning, Tom M. Mitchell, Mc Graw-Hill Publication, 2017

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Web Technologies

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To Learn PHP language for server side scripting
- To introduce XML and processing of XML Data with Java
- To introduce Server side programming with Java Servlets and JSP
- To introduce Client side scripting with JavaScript.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Analyze a web page and identify its elements and attributes.	**K4**
CO2	To acquire knowledge of xml fundamentals and usage of xml technology in electronic data interchange	**K2**
CO3	Build dynamic web pages using JavaScript (client side programming).	**K3**
CO4	To design and develop web based enterprise systems for the enterprises using technologies like jsp, servlet.	**K6**
CO5	Build web applications using PHP	**K3**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1			1				2			
CO2	2		1				1	1		
CO3							3			
CO4	1	2						1		
CO5			3							

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit I:

Web Basics- Introduction, Concept of Internet- History of Internet, Protocols of Internet, World Wide Web, URL, Web Server, Web Browser. **HTML- Introduction**, History of HTML, Structure of HTML Document: Text Basics, Structure of HTML Document: Images and Multimedia, Links and webs, Document Layout, Creating Forms, Frames and Tables, Cascading style sheets.

Unit II:

XML Introduction- Introduction of XMLXML: Introduction to XML, Defining XML tags, their attributes and values, Document Type Definition, XML Schemes, Document Object Model, XHTML Parsing XML Data – DOM and SAX Parsers in java.

Unit III:

Introduction to Servlets: Common Gateway Interface (CGI), Life cycle of a Servlet, deploying a Servlet, The Servlet API, Reading Servlet parameters, Reading Initialization parameters, Handling Http Request & Responses, Using Cookies and Sessions, connecting to a database using JDBC.

Unit IV:

Introduction to JSP: The Anatomy of a JSP Page, JSP Processing, Declarations, Directives, Expressions, Code Snippets, implicit objects, Using Beans in JSP Pages, Using Cookies and session for session tracking, connecting to database in JSP. Client-side Scripting: Introduction to JavaScript, JavaScript language – declaring variables, scope of variables, functions. event handlers (onClick, onSubmit etc.), Document Object Model, Form validation.

Unit V:

Introduction to PHP: Declaring variables, data types, arrays, strings, operators, expressions, control structures, functions, reading data from web form controls like text boxes, radio buttons, lists etc., Handling File Uploads. Connecting to database (MySQL as reference), executing simple queries, handling results, Handling sessions and cookies File Handling in PHP: File operations like opening, closing, reading, writing, appending, deleting etc. on text and binary files, listing directories.

Text Books:

1. Web Technologies, Uttam K Roy, Oxford University Press.
2. The Complete Reference PHP — Steven Holzner, Tata McGraw-Hill.

Reference Books:

1. Web Programming, building internet applications, Chris Bates 2″ edition, Wiley Dreamtech.
2. Java Server Pages —Hans Bergsten, SPD O’Reilly.
3. Java Script, D.Flanagan

Beginning Web Programming-Jon Duckett WROX.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Internet of Things

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

- Introduce the fundamental concepts of IoT and physical computing
- Expose the student to a variety of embedded boards and IoT Platforms
- Create a basic understanding of the communication protocols in IoT communications.
- Familiarize the student with application program interfaces for IoT.
- Enable students to create simple IoT applications.

UNIT I
Overview of IoT:

The Internet of Things: An Overview- The Flavor of the Internet of Things, The “Internet” of “Things”, The Technology of the Internet of Things, Enchanted Objects, Who is Making the Internet of Things?

Design Principles for Connected Devices: Calm and Ambient Technology, Privacy, Web Thinking for Connected Devices, Affordances.

Prototyping: Sketching, Familiarity, Costs Vs Ease of Prototyping, Prototypes and Production, Open source Vs Close source, Tapping into the community.

Learning Outcomes:

After completing this Unit, students will be able to

- Explain IoT architecture. [L2]
- Interpret the design principles that govern connected devices [L2]
- Understand the significance of Prototyping [L2]

UNIT II
Embedded Devices:

Electronics, Embedded Computing Basics, Arduino, Raspberry Pi, Mobile phones and tablets, Plug Computing: Always-on Internet of Things

Learning Outcomes:

After completing this Unit, students will be able to

- Explain the basics of microcontrollers [L2]
- Outline the architecture of Arduino [L2]
- Develop simple applications using Arduino [L3]
- Outline the architecture of Raspberry Pi [L2]
- Develop simple applications using Raspberry Pi [L3]
- Select a platform for a particular embedded computing application [L3]

UNIT III

Communication in the IoT:
Internet Communications: An Overview, IP Addresses, MAC Addresses, TCP and UDP Ports, Application Layer Protocols
Prototyping Online Components:
Getting Started with an API, Writing a New API, Real-Time Reactions, Other Protocols Protocol

Learning Outcomes:
After completing this Unit, students will be able to

- Interpret different protocols and compare them [L2]
- Select protocol to be used for a specific application [L3]
- Utilize the Internet communication protocols for IoT applications [L3]
- Select IoT APIs for an application [L3]
- Design and develop a solution for a given application using APIs [L6]
- Test for errors in the application [L4]

UNIT IV

Business Models: A short history of business models, The business model canvas, Who is the business model for, Models, Funding an Internet of Things startup, Lean Startups.
Manufacturing: What are you producing, Designing kits, Designing printed circuit boards.

Learning Outcomes:
After completing this Unit, students will be able to

- Plan the business model [L6]
- Predict the market value [L5]
- Assemble the product [L6]

UNIT V

Manufacturing continued: Manufacturing printed circuit boards, Mass-producing the case and other fixtures, Certification, Costs, Scaling up software.
Ethics: Characterizing the Internet of Things, Privacy, Control, Environment, Solutions.

Learning Outcomes:
After completing this Unit, students will be able to

- Employ the manufacturing techniques [L4]
- Adapt the Ethics [L6]

Text Book:

1. Adrian McEwen, Hakim Cassimally - Designing the Internet of Things, Wiley Publications, 2012

Reference Books:

1. Arshdeep Bahga, Vijay Madisetti - Internet of Things: A Hands-On Approach, Universities Press, 2014.
2. The Internet of Things, Enabling technologies and use cases – Pethuru Raj, Anupama C. Raman, CRC Press.

Reference sites:

1. https://www.arduino.cc/
2. https://www.raspberrypi.org/

Course outcomes:

At the end of the course, students will be able to

- Choose the sensors and actuators for an IoT application (L1)
- Select protocols for a specific IoT application (L2)
- Utilize the cloud platform and APIs for IoT applications (L3)
- Experiment with embedded boards for creating IoT prototypes (L3)
- Design a solution for a given IoT application (L6)

Establish a startup [L4]

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Cloud Computing**

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To explain the evolving computer model caned cloud computing.
- To introduce the various levels of services that can be achieved by cloud.
- To describe the security aspects in cloud.
- To motivate students to do programming and experiment with the various cloud computing environments.

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Illustrate the key dimensions of the challenge of Cloud Computing	**K2**
CO2	Classify the Levels of Virtualization and mechanism of tools.	**K2**
CO3	Analyze Cloud infrastructure including Google Cloud and Amazon Cloud.	**K4**
CO4	Explain Cloud Programming and Software Environments.	**K5**
CO5	Apply authentication, confidentiality and privacy issues in Cloud resource management.	**K3**

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1		1		1					
CO2						2				
CO3				1				1		
CO4					1		1			
CO5	2								1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT–I:

Systems modeling, Clustering and virtualization: Scalable Computing over the Internet, Technologies for Network based systems, System models for Distributed and Cloud Computing, Software environments for distributed systems and clouds, Performance, Security And Energy Efficiency.

UNIT–II:

Virtual Machines and Virtualization of Clusters and Data Centers: Implementation Levels of Virtualization, Virtualization Structures/ Tools and mechanisms, Virtualization of CPU, Memory and I/O Devices, Virtual Clusters and Resource Management, Virtualization for Data Centre Automation.

UNIT–III:

Cloud Platform Architecture: Cloud Computing and service Models, Architectural Design of Compute and Storage Clouds, Public Cloud Platforms, Inter Cloud Resource Management, Cloud Security and Trust Management. Service Oriented Architecture, Message Oriented Middleware.

UNIT–IV:

Cloud Programming and Software Environments: Features of Cloud and Grid Platforms, Parallel & Distributed Programming Paradigms, Programming Support of Google App Engine, Programming on Amazon AWS and Microsoft Azure, Emerging Cloud Software Environments. **Storage Systems**: Evolution of storage technology, storage models, file systems and database, distributed file systems, general parallel file systems. Google file system., Apache Hadoop, BigTable, Megastore, Amazon Simple Storage Service(S3).

UNIT-V:

Cloud Resource Management and Scheduling : Policies and Mechanisms for Resource Management Applications of Control Theory to Task Scheduling on a Cloud, Stability of a Two Level Resource Allocation Architecture, Feedback Control Based on Dynamic Thresholds. Coordination of Specialized Autonomic Performance Managers, Resource Bundling, Scheduling Algorithms for Computing Clouds, Fair Queuing, Start Time Fair Queuing, Borrowed Virtual Time, Cloud Scheduling Subject to Deadlines, Scheduling MapReduce Applications Subject to Deadlines.

Text Books:

1. Distributed and Cloud Computing, Kai Hwang, Geoffry C. Fox, Jack J. Dongarra MK Elsevier.
2. Cloud Computing, Theory and Practice, Dan C Marinescu, MK Elsevier.
3. Cloud Computing, A Hands on approach, Arshadeep Bahga, Vijay Madisetti, University Press

Reference Books:

1. Cloud Computing: A Practical Approach. Anthony T.Velte. Toby J.VeFte, Robert Elsenpeter. Tata McGraw Hill. rp2Oll.
2. Enterprise Cloud Computing Gautam Shroif, Cambridge University Press. 2010.
3. Cloud Computing: Implementation, Management and Security, John W. Rittinouse, James F Ransome. CRC Press, rp2012.
4. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud. George Reese, O'Really SPD, rp2Oll.
5. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance, Tim Mather, Subra Ktriaraswamy, Shahed Latif, O'Redç SPD, rp2Oll.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Image Processing**

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To comprehend the relation between human visual system and machine perception and processing of digital images.
- To provide a detailed approach towards image processing applications like enhancement, segmentation, and compression.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Analyze the computational methods on digital images.	**K4**
CO2	Explain the implement the spatial and frequency domain image transforms on enhancement and restoration of images.	**K2**
CO3	Elaborate understanding on machine learning techniques.	**K6**
CO4	Expected to Define and Apply the need for compression and evaluate the basic compression algorithms	**K5,1**
CO5	Make use of introduce object tracking approaches.	**K3**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		1	2						1	
CO2					2	1				
CO3	1					1	2			
CO4										
CO5					2					

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I:

Image Formation and Coordinate Transformations Camera Matrix, Motion/Stereo Pin-hole model, Human eye / cognitive aspects of colour / 3D space; illumination; Sampling and Quantization Coordinate transformations and camera parameters

UNIT – II:

Image Processing - Noise Removal, Blurring, Edge Detection: Canny / Gaussian/ Gabor/Texture Edges/ Curvature / Corner Detection.

UNIT – III:

Segmentation - Concept of Figure vs. Ground, Watershed, Change Detection, Background Subtraction, Texture Segmentation Gaussian Mixture Models - Applications in Color/Motion based Image Segmentation, Background Modeling and Shape Clustering

UNIT – IV:

Machine Learning techniques in Vision Bayesian Classification, Maximum Likelihood Methods, Neural Networks; Non-parametric models; Manifold estimation Support Vector Machines ; Temporal sequence learning.

UNIT – V:

Introduction to Object Tracking - Exhaustive vs. Stochastic Search Shapes, Contours, and Appearance Models. Mean-shift tracking; Contour-based models, Object Modeling and Recognition Fundamental matrix / Epipolar geometryAdaboost approaches: Face Detection / Recognition Large Datasets; Attention models.

Text Books

1. [FP]: David Forsyth and Jean Ponce, Computer Vision: A modern Approach,
   Prentice
2. Christopher Bishop, Pattern Recognition and Machine Learning, Springer, 2008.

Reference Books:

1. E.R. Davies, Machine Vision, Theory Algorithms Practicalities, Elsevier 2005
2. Milan Sonka, Vaclav Hlavac and Roger Boyle, Image Processing, Analysis, and Machine Vision. Brooks/Cole / Thomson 1999
3. Basics of some image processing aspects. Texture Chapter 24 (Perception) of Russell and Norvig: AI: A modern Approach, Prentice Hall 2000.
4. Richard Hartley and Andrew Zisserman, Multiple View Geometry in Computer Vision, Cambridge Univ Press 2000 More detailed treatment of 3D structure recovery
5. Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification, 2nd ed., Wiley Asia, 2002

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 Principles of Cryptography and Network Security

SEMESTER - III L-T-P-C: 3-0-0-3

Course Objectives:

- To learn various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms,
- To Familiar in design issues and working principles of various authentication protocols and various secure communication standards including Kerberos, IPsec, and S/MIME

Course Outcomes: At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Explain Basic Principles, different security threats, countermeasures, foundation course of cryptography mathematics and Symmetric Encryption.	**K2**
CO2	Classify the basic principles of Asymmetric key algorithms and operations of asymmetric key cryptography.	**K4**
CO3	Design Cryptographic Hash Functions as SHA-3 and Digital Signatures as Elgamal	**K6**
CO4	Explain the concept of Revise Key Management and Distribution and User Authentication	**K3**
CO5	Determine the knowledge of Network and Internet Security Protocols such as S/MIME	**K5**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**
CO1	2									
CO2		1						1		
CO3		2								1
CO4	1		1							
CO5	1								1	1

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT I:

Basic Principles: Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography. **Symmetric Encryption:** Mathematics of Symmetric Key Cryptography, Introduction to Modern Symmetric Key Ciphers, Data Encryption Standard, Advanced Encryption Standard.

UNIT II:

Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography-Primes, primality Testing, Factorization, Asymmetric Key Cryptography-RSA Cryptosystem, Rabin Cryptosystem, ElGamal Cryptosystem, Elliptic Curve Cryptosystem

UNIT III:

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Two Simple Hash Functions Requirements and Security Hash Functions Based on Cipher Block Chaining, Secure Hash Algorithm (SHA), SHA-3. **Digital Signatures:** Elgamal Digital Signature Scheme, Schnorr Digital Signature, NIST Digital Signature Algorithm

Unit IV:

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates. **User Authentication:** User Authentication, Remote User-Authentication Principle, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User-Authentication Using Asymmetric Encryption

Unit V: Network and Internet Security

Electronic Mail Security: Internet Mail Architecture, Email Formats, Email Threats and Comprehensive Email Security, S/MIME. **IP Security:** IP Security Policy, Encapsulating Security Payload, Combining Security Associations Internet Key Exchange

Text Books:

1. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security, William Stallings, Global Edition, 7e Pearson, 2017

Reference Books:

1. Network Security and Cryptography, First Edition, Bernard Meneges, Cengage Learning, 2018

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **SOFTWARE TESTING**

SEMESTER - III **L-T-P-C: 3-0-0-3**

COURSE OBJECTIVES

- Fundamentals for various testing methodologies.
- Describe the principles and procedures for designing test cases.
- Provide supports to debugging methods.
- Acts as the reference for software testing techniques and strategies.

COURSE OUTCOMES

- Understand the basic testing procedures.
- Able to support in generating test cases and test suites.
- Able to test the applications manually by applying different testing methods and automation tools.
- Apply tools to resolve the problems in Real time environment.

UNIT I

Introduction: Purpose of Testing, Dichotomies, Model for Testing, Consequences of Bugs, Taxonomy of Bugs.

Flow graphs and Path testing: Basics Concepts of Path Testing, Predicates, Path Predicates and Achievable Paths, Path Sensitizing, Path Instrumentation, Application of Path Testing.

UNIT II

Transaction Flow Testing: Transaction Flows, Transaction Flow Testing Techniques.

Dataflow testing: Basics of Dataflow Testing, Strategies in Dataflow Testing, Application of Dataflow Testing.

UNIT III

Domain Testing: Domains and Paths, Nice & Ugly Domains, Domain testing, Domains and Interfaces Testing, Domain and Interface Testing, Domains and Testability.

UNIT IV

Paths, Path products and Regular expressions: Path Products & Path Expression, Reduction Procedure, Applications, Regular Expressions & Flow Anomaly Detection.

Logic Based Testing: Overview, Decision Tables, Path Expressions, KV Charts, Specifications.

UNIT V:

State, State Graphs and Transition Testing: State Graphs, Good & Bad State Graphs, State Testing, Testability Tips.

Graph Matrices and Application: Motivational Overview, Matrix of Graph, Relations, Power of a Matrix, Node Reduction Algorithm, Building Tools.

Text Books:

1. Software testing techniques – Boris Beizer, Dreamtech, second edition.

Reference Books :

1. The craft of software testing - Brian Marick, Pearson Education.

2. Software Testing- Yogesh Singh, Camebridge

3. Software Testing, 3rd edition, P.C. Jorgensen, Aurbach Publications (Dist.by SPD).

4. Software Testing, N.Chauhan, Oxford University Press.

5. Introduction to Software Testing, P.Ammann & J.Offutt, Cambridge Univ. Press.

6. Effective methods of Software Testing, Perry, John Wiley, 2nd Edition, 1999.

7. Software Testing Concepts and Tools, P.Nageswara Rao, dreamtech Press

8. Win Runner in simple steps by Hakeem Shittu,2007 Genixpress.

9. Foundations of Software Testing, D.Graham & Others, Cengage Learning

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
MOBILE APPLICATION DEVELOPMENT

SEMESTER - III **L-T-P-C: 3-0-0-3**

Course Objectives:

- Design, implement and evaluate a User Interface for a mobile application using J2ME.
- Create a small but realistic working mobile application for small computing devices.
- Categorize the challenges posed by developing mobile applications and be able to propose and evaluate and select appropriate solutions.

Unit I

J2ME Overview, Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices, Small Computing Technology: Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants

Unit II

J2ME Architecture and Development Environment, J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2ME Style, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit, J2ME Best Practices and Patterns: The Reality of Working in a J2ME World, Best Practices

Unit III

Commands, Items, and Event Processing, J2ME User Interfaces, Display Class, the Palm OS Emulator, Command Class, Item Class, Exception Handling

High-Level Display: Screens: Screen Class, Alert Class, Form Class, Item Class, List Class, Text Box Class, Ticker Class

Low-Level Display: Canvas: The Canvas, User Interactions, Graphics, Clipping Regions, Animation

Unit IV

Record Management System: Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener

JDBC Objects: The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions

JDBC and Embedded SQL: Model Programs, Tables, Indexing, Inserting Data into Tables, Selecting Data from a Table, Metadata, Updating Tables, Deleting Data form a Table, Joining Tables, Calculating Data, Grouping and Ordering Data, Subqueries, VIEWs

Unit V

Generic Connection Framework The Connection, Hypertext Transfer Protocol, Communication Management Using HTTP Commands, Session Management, Transmit as a Background Process

TEXT BOOKS:

1. J2ME: The Complete Reference, James Keogh, Tata McGrawHill.
2. Programming for Mobile and Remote Computers,G.T.Thampi,dreamtech press.

REFERENCE BOOKS:

1.Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, Pearson Education, 2004

2. Beginning Java ME Platform, Ray Rischpater, Apress, 2009

3.Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, Apress, 2005

4.Kicking Butt with MIDP and MSA:Creating Great Mobile Applications,1st edition,J.Knudsen,Pearson.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
WEB TECHNOLOGIES LAB

SEMESTER - III **L-T-P-C: 0-0-3-1.5**

Course Objectives:

- The primary objective of the course is to learn web programming by designing and developing some web based applications.

List of Sample Problems

1. Develop static pages (using Only HTML) of an online Book store. The pages should resemble: www.amazon.com The website should consist the following pages.
   Home page, Registration and user Login
   User Profile Page, Books catalog
   Shopping Cart, Payment By credit card
   Order Conformation

2. Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

3. Create and save an XML document at the server, which contains 10 users information. Write a program, which takes User Id as an input and returns the user details by taking the user information from the XML document.

*4. Bean Assignments

a. Create a JavaBean which gives the exchange value of INR(Indian Rupees) into equivalent American/Canadian/Australian Dollar value.

b. Create a simple Bean with a label - which is the count of number of clicks. Than create a BeanInfo class such that only the “count” property is visible in the Property Window.

c. Create two Beans-a)KeyPad .b)DisplayPad .After that integrate the two Beans to make it work as a Calculator.

d. Create two Beans Traffic Light(Implemented as a Label with only three background colours-Red,Green,Yellow) and Automobile(Implemented as a TextBox which states its state/movement). The state of the Automobile should depend on the following Light Transition Table.

Light Transition	Automobile State
Red ---> Yellow	Ready
Yellow ---> Green	Move
Green --> Red	Stopped

5. Install TOMCAT web server. Convert the static web pages of assignments 2 into dynamic web pages using Servlets and cookies. Hint: Users information (user id, password, credit card number) would be stored in web.xml. Each user should have a separate Shopping Cart.

6. Redo the previous task using JSP by converting the static web pages of assignments 2 into dynamic web pages. Create a database with user information and books information. The books catalogue should be dynamically loaded from the database. Follow the MVC architecture while doing the website.

7. Implement the “Hello World!” program using JSP Struts Framework.

8.Redo the problem 5 using PHP.

Additional Assignment Problems for the WT Lab.:

Write an HTML page including any required Javascript that takes a number from one text field in the range of 0 to 999 and shows it in another text field in words. If the number is out of range, it should show “out of range” and if it is not a number, it should show “not a number” message in the result box.

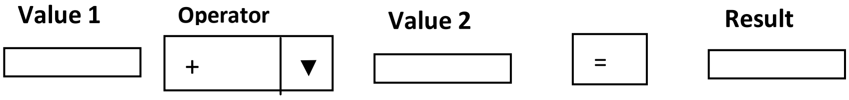
Write a java swing application that takes a text file name as input and counts the characters, words and lines in the file. Words are separated with white space characters and lines are separated with new line character.

Write a simple calculator servlet that takes two numbers and an operator (+, -, /, * and %) from an HTML page and returns the result page with the operation performed on the operands. It should check in a database if the same expression is already computed and if so, just return the value from database. Use MySQL or PostgreSQL.(Do the same problem using PHP)

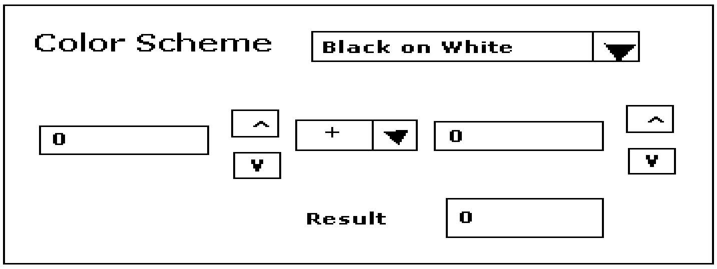
Write an HTML page that contains a list of 5 countries. When the user selects a country, its capital should be printed next to the list. Add CSS to customize the properties of the font of the capital (color, bold and font size).

Write a servlet that takes name and age from an HTML page. If the age is less than 18, it should send a page with “Hello <name>, you are not authorized to visit this site” message, where <name> should be replaced with the entered name. Otherwise it should send “Welcome <name> to this site” message. (Do the same problem using PHP)

Write a calculator program in HTML that performs basic arithmetic operations (+, -, /, * and %). Use CSS to change the foreground and background color of the values, buttons and result display area separately. Validate the input strings using JavaScript regular expressions. Handle any special cases like division with zero reasonably. The screen may look similar to the following:

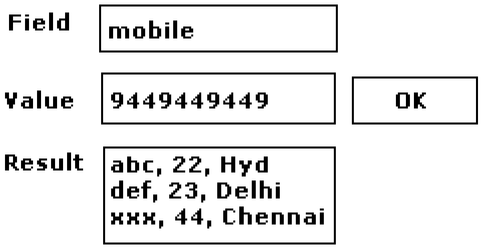


Write a Java program that creates a calculator GUI, as shown in figure. Extra components may be added for convenience:



The Color Scheme may be Black on White or Blue on Yellow (selectable) and accordingly all components colors must be changed. The values can be either entered or increased or decreased by a step of 10. The operators are +, -, / and * (selectable). Once any change takes place, the result must be automatically computed by the program.

Write a Java Application that will read an XML file that contains personal information (Name, Mobile Number, age and place. It reads the information using SAX parser. After reading the information, it shows two input Text Fields in a window, one for tag name and the other for value. Once these two values are given, it should list all the records in the XML file that match the value of the given field in a text area (result box). For example, if the two text boxes are entered with “name” and “ABCD” then it should show all the records for which name is “ABCD”? An Illustration is given below that takes a mobile number and lists all the records that have the same mobile number.



Field: mobile

Value: 9449449449

Result:
abc, 22, Hyd
def, 23, Delhi
xxx, 44, Chennai

Consider the following web application for implementation:
The user is first served a login page which takes user's name and password. After submitting the details the server checks these values against the data from a database and takes the following decisions.
If name and password matches, serves a welcome page with user's full name.
If name matches and password doesn't match, then serves “password mismatch” page
If name is not found in the database, serves a registration page, where users full name, present user name (used to login) and password are collected. Implement this application using:

1. Pure JSP
2. Pure Servlets
3. Struts Framework
4. PHP

Implement a simple arithmetic calculator with +, -, /, *, % and = operations using Struts Framework The number of times the calculator is used should be displayed at the bottom (use session variable).

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective-2 **Machine Learning with Python Lab**
SEMESTER - III L-T-P-C: 0-1-2-2

Course Objectives:

- Make use of Data sets in implementing the machine learning algorithms
- Implement the machine learning concepts and algorithms in any suitable language of choice.
- Design Python programs for various Learning algorithms.

Course Outcomes(COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Implement procedures for the machine learning algorithms	**K4**
CO2	Design Python programs for various Learning algorithms	**K6**
CO3	Apply appropriate data sets to the Machine Learning algorithms	**K3**
CO4	Identify and apply Machine Learning algorithms to solve real world problems	**K2**

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	1								
CO2		1	3							
CO3								1	2	
CO4	3							1	1	

(Levels of Correlation: 1-low, 2-medium 3-high)

Experiment 1:

Implement and demonstrate the FIND-S algorithm for finding the most specific hypothesis based on a given set of training data samples. Read the training data from a .csv file

Experiment 2:

For a given set of training data examples stored in a .csv file, implement and demonstrate the Candidate-Elimination algorithm to output a description of the set of all hypotheses consistent with the training examples

Experiment 3:
Write a program to demonstrate the working of the decision tree based ID3 algorithm. Use an appropriate data set for building the decision tree and apply this knowledge to classify a new sample.

Experiment 4:
Write a Python program to implement k-Nearest Neighbour algorithm to classify the iris data set. Print both correct and wrong predictions.

Experiment 5:
Build an Artificial Neural Network by implementing the Backpropagation algorithm and test the same using appropriate data sets.

Experiment 6:
Write a program to implement the naive Bayesian classifier for a sample training data set stored as a .csv file. Compute the accuracy of the classifier, considering few test data sets.

Experiment 7:
Write a Python program to construct a Bayesian network considering medical data. Use this model to demonstrate the diagnosis of heart patients using standard Heart Disease Data Set.

Experiment 8:
Assuming a set of documents that need to be classified, use the naive Bayesian Classifier model to perform this task. Built-in Java classes/API can be used to write the program. Calculate the accuracy, precision and recall for your data set.

Experiment 9:
Apply EM algorithm to cluster a set of data stored in a .CSV file. Use the same data set for clustering using k-Means algorithm. Compare the results of these two algorithms and comment on the quality of clustering using Python Programming.

Experiment 10:
Implement the non-parametric Locally Weighted Regression algorithm in order to fit data points. Select appropriate data set for your experiment and draw graphs.

INTERNET OF THINGS LABORATORY

Semester – III **L-T-P-C: 0-0-3-1.5**

Practicals:

1. Select any one development board (Eg., Arduino or Raspberry Pi) and control LED using the board.
2. Using the same board as in (1), read data from a sensor. Experiment with both analog and digital sensors.
3. Control any two actuators connected to the development board using Bluetooth.
4. Read data from sensor and send it to a requesting client. (using socket communication)
   Note: The client and server should be connected to same local area network.
5. Create any cloud platform account, explore IoT services and register a thing on the platform.
6. Push sensor data to cloud.
7. Control an actuator through cloud.
8. Access the data pushed from sensor to cloud and apply any data analytics or visualization services.
9. Create a mobile app to control an actuator.
10. Design an IoT based air pollution control system which monitors the air pollution by measuring carbon monoxide, ammonia, etc and gives alarm or sends message when the pollution level is more than permitted range.
11. Design an IoT based system which measures the physical and chemical properties of the water and displays the measured values.
12. Identify a problem in your local area or college which can be solved by integrating the things you learned and create a prototype to solve it (Mini Project).
13. Design a business model canvas for a digital display

Text Book:

2. Adrian McEwen, Hakim Cassimally - Designing the Internet of Things, Wiley Publications, 2012.
3. Alexander Osterwalder, and Yves Pigneur – Business Model Generation – Wiley, 2011

Reference Books:

3. Arshdeep Bahga, Vijay Madisetti - Internet of Things: A Hands-On Approach, Universities Press, 2014.
4. The Internet of Things, Enabling technologies and use cases – Pethuru Raj, Anupama C. Raman, CRC Press.

Reference sites:

https://www.arduino.cc/

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
MOBILE APPLICATION DEVELOPMENT LABORATORY

Semester – III **L-T-P-C: 0-0-2-1**

List of Experiments:

Week - 1: Installation of Java Wireless Toolkit (J2ME)

1) If the Java Development Kit (JDK) is not there or only having the Java Runtime Environment (JRE) installed, install the latest JDK from http://java.sun.com/javase/downloads/index.jsp. Current stable release of Java is JDK 6 Update 7 but check the web page in case there are newer non-beta releases available.
2) Next, download the Java Wireless Toolkit (formerly called J2ME Wireless Toolkit) from: http://java.sun.com/products/sjwtoolkit/download.html.
3) Run the installer (for example, for Windows it is: sun_java_wireless_toolkit- 2_5_2-windows.exe). The installer checks whether a compatible Java environment has been pre-installed. If not, it is necessary to uninstall old versions of Java and perform Step 1 again.

Once after successful installation of Java and the tool kit compile this program and run the following program in the toolkit.

Steps to run this program in toolkit:

1. Start -> All Programs -> Sun Java Tool Kit -> Wireless Tool Kit
2. Click New Project – Enter Project Name -> Enter Class Name -> Click on Create Project.
3. Choose appropriate API Selection and Configurations.
4. Place Java Source file in WTK2.1 / WTK2.2\apps\projectname\src
5. Build the Project.
6. Run the Project.

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;
public class HelloWorld extends MIDlet{
  private Form form;
  private Display display;
public HelloWorld(){
super();
  }
public void startApp(){
form = new Form("Hello World");
   String msg = "Hello World!!!!!!!";
form.append(msg);
   display = Display.getDisplay(this);
display.setCurrent(form);
  }
public void pauseApp(){}
public void destroyApp(boolean unconditional){
notifyDestroyed();
```

} }

Week - 2 Working with J2ME Features:

Working with J2ME Features: Say, creating a Hello World program Experiment with the most basic features and mobile application interaction concepts (lists, text boxes, buttons, radio boxes, soft buttons, graphics, etc)

2.1 Create a program which creates to following kind of menu.

* cut
* copy
* past
* delete
* select all
* unselect all

2.2 Event Handling.

Create a menu which has the following options:

* cut - can be on/off
* copy - can be on/off
* paste - can be on/off
* delete - can be on/off
* select all - put all 4 options on
* unselect all - put all 4 options off

2.3. Input checking

Create an MIDP application which examine, that a phone number, which a user has entered is in the given format.

* Area code should be one of the following: 040, 041, 050, 0400, 044
* There should 6-8 numbers in telephone number (+ area code)

Week - 3 Threads & High Level UI:

3.1. Create a slide show which has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide program returns to the first slide.

3.2 High-level UI

Create a MIDP application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts and shows to the user how many right answers were right and shows them to user.

3.3 Create a MIDP application, where the user can enter player name and points. The program saves the information to the record using RMS at MIDP device. Program should also print out the top 10 player list to the end user. You can use this class in your game if you made own class for saving and reading record sets.

Week - 4 Working on Drawing and Images

4.1 Create a slide show which has three slides, which includes pictures at PNG format. Program should change to the new slide other 5 seconds.

4.2 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array.

4.3 Create a MIDP application, which draws a bar graph to the display. Data values can be given at int[] array. You can enter four data (integer) values to the input text field.

Week - 5 Developing Networked Applications using the Wireless Toolkit

Creating a Simple Client-Server Application

Create, compile and run a basic UDP-based client-server application.

Creating the Datagram Server project

1) Click on Wireless Toolkit 2.5.2 under the group: All Programs→Sun Java (TM) Wireless Toolkit 2.5.2.

2) Click on 'New Project...' button.

3) Enter project name as 'DatagramServer'. Enter MIDlet name as 'DatagramServer'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class, otherwise the application won‘t run.

4) Another window pops up where it is required to select a target platform. Select 'MIDP 1.0' from the drop down list.

5) After clicking OK, the project is created; and the Wireless Toolkit tells that the name of the folder where source code files are created. The path of the source code folder is displayed in the debug output window.

Creating and Compiling the Datagram Server source files

The Wireless Toolkit does not come with an IDE by default so Use any IDE or a text editor like Notepad.

1) Create a new text file called DatagramServer.java in the source folder of the project. The exact path of this folder is displayed in the Wireless Toolkit window.

2) Paste contents DatagramServer.java from into the source file.

Running your Server application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of your application highlighted on its screen as shown below.

3) To start the application, click on the right soft-key (marked with a dot) below the ‗Launch‘ command.

4) The phone simulator might ask if it is OK to run the network application. Select ‗Yes‘ by clicking on the appropriate soft-key. The server is now up and running.

5) Keep the server running during the creation, compilation and running of the Datagram Client application.

Creating the Datagram Client project

1) Use the same instance of the Wireless Toolkit that is used for creating and compiling the Datagram Server project.

2) Click on 'New Project...' button.

3) A new window pops up. Enter project name as 'DatagramClient'. Enter MIDlet name as 'DatagramClient'. Note that the Midlet name is the same as the name of the class in the source code, which extends the MIDlet class.

4) Another window pops up where one has to select a target platform. Select 'MIDP 1.0' from the drop down list.

5) After clicking OK, the project is created and the Wireless Toolkit tells where to place the source code files. The path of the source code folder is displayed in the debug output window as explained before.

Creating and Compiling the Datagram Client source files

1) Create a new text file called DatagramClient.java in the source folder of the project.

2) Paste contents DatagramClient.java into the source file.

3) Then click on the Build button in the Wireless Toolkit window. If the compilation is OK, it will say Build Complete in the window's debug output window, otherwise it will show the errors. Note: In the source code, use the System.out.println() statement to output debug information to this window.

Running your Client application on the Phone simulator

1) After compiling the project successfully, click on the Run button in the Wireless Toolkit window.

2) A graphical window depicting a phone handset will appear with the name of the application highlighted on its screen.

3) To start the application, click on the right soft-key (marked with a dot) below the _Launch‘ command.

4) The phone simulator might ask if it is OK to run the network application. Select _Yes‘ by clicking on the appropriate soft-key. The client is now up and running.

5) When the client executes on the phone simulator, one should see a text box with the caption 'Message'. Enter any message and press the right soft-key (corresponding to Send). If the client- server application is working properly, the screen of the server phone will display the message sent by the client and the client screen will now display a message sent by the server in response. The response message from the server is the original client message in reverse. 6) Try various features of the phone simulator including the different look-and feel options.

Week - 6 Authentication with a Web Server

6.1 Write a sample program to show how to make a SOCKET Connection from j2me phone.

This J2ME sample program shows how to how to make a SOCKET Connection from a J2ME Phone. Many a times there is a need to connect backend HTTP server from the J2ME application. shows how to make a SOCKET connection from the phone to port 80.

6.2 Login to HTTP Server from a J2ME Program

This J2ME sample program shows how to display a simple LOGIN SCREEN on the J2ME phone and how to authenticate to a HTTP server.

Many J2ME applications for security reasons require the authentication of the user. This free J2ME sample program, shows how a J2ME application can do authentication to the backend server.

Note: Use Apache Tomcat Server as Web Server and Mysql as Database Server.

Text Books:

1. J2ME: The Complete Reference, James Keogh, TMH.

References:

4. Enterprise J2ME: Developing Mobile Java Applications, Michael Juntao Yuan, Pearson Education, 2004.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective:3
Block Chain technologies

SEMESTER - IV L-T-P-C: 3-0-0-3

Course Objectives:

- Impart strong technical understanding of Blockchain technologies
- Develop familiarity of current technologies, tools, and implementation strategies
- Introduce application areas, current practices, and research activity

Course Outcomes (Cos): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Demonstrate the foundation of the Blockchain technology and understand the processes in payment and funding.	K2
CO2	Identify the risks involved in building Blockchain applications.	K5
CO3	Review of legal implications using smart contracts.	K4
CO4	Choose the present landscape of Blockchain implementations and Understand Cryptocurrency markets.	K3
CO5	Examine how to profit from trading cryptocurrencies.	K3

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	3						2			
CO2								3		
CO3				2					1	
CO4										
CO5										2

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT – I:

The consensus problem, Asynchronous Byzantine Agreement, AAP protocol and its analysis, Nakamoto Consensus on permission-less, nameless, peer-to-peer network, Abstract Models for BLOCKCHAIN, GARAY model, RLA Model, Proof of Work (PoW) as random oracle, formal treatment of consistency, liveness and fairness - Proof of Stake (PoS) based Chains, Hybrid models (PoW + PoS).

UNIT – II:
cryptographic basics for cryptocurrency, A short overview of Hashing, signature schemes, encryption schemes and elliptic curve cryptography

UNIT – III:
Bitcoin, Wallet, Blocks, Merkley Tree, hardness of mining, transaction verifiability, anonymity, forks, double spending, mathematical analysis of properties of Bitcoin.

UNIT – IV:
Ethereum: Ethereum Virtual Machine (EVM),Wallets for Ethereum, Solidity, Smart Contracts, some attacks on smart contracts

UNIT – V:
(Trends and Topics): Zero Knowledge proofs and protocols in Blockchain, Succinct non interactive argument for Knowledge (SNARK), pairing on Elliptic curves ,Zcash.

Text Books:

1. Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. Bitcoin and cryptocurrency technologies: a comprehensive introduction. Princeton University Press, 2016. (Free download available)

Reference Books:

1. Joseph Bonneau et al, SoK: Research perspectives and challenges for Bitcoin and cryptocurrency, IEEE Symposium on security and Privacy, 2015 (article available for free download) {curtain raiser kind of generic article, written by seasoned experts and pioneers}.
2. J.A.Garay et al, The bitcoin backbone protocol - analysis and applications EUROCRYPT 2015 LNCS VOl 9057, (VOLII), pp 281-310. (Also available at eprint.iacr.org/2016/1048). (serious beginning of discussions related to formal models for bitcoin protocols).

R.Pass et al, Analysis of Blockchain protocol in Asynchronous networks , EUROCRYPT 2017, (eprint.iacr.org/2016/454) . A significant progress and consolidation of several principles).

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective:3
Deep Learning

SEMESTER - IV L-T-P-C: 3-0-0-3

UNIT 1:

Introduction to TensorFlow :Computational Graph, Key highlights, Creating a Graph, Regression example, Gradient Descent, TensorBoard, Modularity, Sharing Variables,Keras Perceptrons: What is a Perceptron, XOR Gate

UNIT 2:

Activation Functions : Sigmoid,ReLU, Hyperbolic Fns,Softmax Artificial Neural Networks : Introduction, Perceptron Training Rule, Gradient Descent Rule

UNIT 3:

Gradient Descent and Backpropagation: Gradient Descent, Stochastic Gradient Descent, Backpropagation, Some problems in ANN Optimization and Regularization :Overfitting and Capacity, Cross Validation, Feature Selection, Regularization, Hyperparameters

UNIT 4:

Introduction to Convolutional Neural Networks: Introduction to CNNs, Kernel filter, Principles behind CNNs, Multiple Filters, CNN applications Introduction to Recurrent Neural Networks: Introduction to RNNs, Unfolded RNNs, Seq2Seq RNNs, LSTM, RNN applications

UNIT 5:

Deep Learning applications: Image Processing, Natural Language Processing, Speech Recognition, Video Analytics

Text Book 1. Goodfellow, I., Bengio,Y., and Courville, A., Deep Learning, MIT Press, 2016.

References 1. Bishop, C. ,M., Pattern Recognition and Machine Learning, Springer, 2006.

2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.

3. Golub, G.,H., and Van Loan,C.,F., Matrix Computations, JHU Press,2013.

4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

(w.e.f 2020-21)

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)

Program Elective:3

Design Patterns

SEMESTER - IV **L-T-P-C: 3-0-0-3**

Course Objectives:

- Understand the concept of Design patterns and its importance.
- Understand the behavioral knowledge of the problem and solutions.
- Relate the Creational, Structural, behavioral Design patterns.
- Apply the suitable design patterns to refine the basic design for given context.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Illustrate the appropriate design patterns to solve object-oriented design problems.	K2
CO2	Apply structural patterns to solve design problems.	K3
CO3	Evaluate the design solutions by using behavioral patterns.	K5
CO4	Develop design solutions using creational patterns	K6
CO5	Demonstrate about Advanced Patterns like Pattern Catalogs	K3

#*Based on suggested Revised BTL*

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		3					2			
CO2		3					1			
CO3					2		1			
CO4								2		
CO5			1					2		

(Levels of Correlation: 1-low, 2-medium 3-high)

Unit-I:

Introduction: History and Origin of Patterns, Design Patterns in MVC, Describing Design Patterns, How Design Patterns Solve Design Problems, selecting a Design Pattern, Using a Design Pattern

Unit-II:

Design Patterns-1: Creational, Abstract Factory-Builder, Factory Method, Prototype-Singleton

Unit- III:

Design Patterns-2: Structural Patterns: Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy

Unit-IV:

Design Patterns-3: Behavioural Patterns, Chain of Responsibility, Command-Interpreter, Iterator- Mediator, Memento, Observer, State, Strategy, Template Method, Visitor

Unit-V:

Advanced Patterns: Pattern Catalogs and Writing Patterns, Patterns and Case Study: Designing a Document Editor Anti-Patterns - Case Studies in UML and CORBA, Pattern Community.

Text Books:

1. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design patterns: Elements of Reusable object-oriented software, Addison-Wesley, 1995.
2. James W Cooper, Java Design Patterns - A Tutorial, Addison-Wesley, 52000.

Reference Books:

1. Craig Larman, Applying UML and Patterns: An Introduction to object-Oriented Analysis and Design and iterative development,3rd Edition, Pearson,2005.
2. Thomas J Mowbray and Raphael Malveau, CORBA and Design Patterns, John Wiley, 1997.
3. William J Brown, Anti-Patterns: Refactoring Software, Architectures and Projects in Crisis, John Wiley, 1998.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective-4
Big Data Analytics

SEMESTER - IV L-T-P-C: 3-0-0-3

Course Objectives:

- To know the fundamental concepts of big data and analytics.
- To explore tools and practices for working with bigdata
- To learn about stream computing.
- To know about the research that requires the integration of large amounts of data.

Course Outcomes (COs): At the end of the course, student will be able to

Course Outcomes		Knowledge Level (K)#
CO1	Identify the need-based tools, viz., Pig and Hiveand to handle and formulate an effective strategy to implement a successful Data analytics project	K3
CO2	Organize the existing technologies and the need of distributed files systems to analyze the big data	K3
CO3	To Discuss the cluster and classification techniques	K5
CO4	Analyze the concepts of strem memory and spark models.	K4
CO5	Explain the use of NoSQL database in data analytics.	K5

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1			3					1		
CO2	1				2					
CO3					2					
CO4								3		
CO5			3							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

Introduction to Big Data- Evolution of Big data, Best Practices for Big data Analytics, Big data characteristics, Validating, The Promotion of the Value of Big Data,Big Data Use Cases, Characteristics of Big Data Applications, Perception and Quantification of Value, Understanding Big Data Storage, A General Overview of High, Performance Architecture, HDFS, MapReduce and YARN, Map Reduce Programming Model

UNIT-II:

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 Zoo Keeper, IBM InfoSphere Big Insights and Streams

UNIT-III:
Clustering and Classification-Advanced Analytical Theory and Methods: Overview of Clustering, K-means, Use Cases - Overview of the Method, Determining the Number of Clusters, Diagnostics, Reasons to Choose and Cautions. Classification: Decision Trees, Overview of a Decision Tree, The General Algorithm, Decision Tree Algorithms, Evaluating a Decision Tree, Decision Trees in R, Naïve Bayes, Baye'sTheorem, Naïve Bayes Classifier.

UNIT- IV:
Stream Memory and Spark- Introduction to Streams Concepts, Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Introduction to Spark Concept, Spark Architecture and components, spark installation, spark RDD(Resilient Distributed Dataset), spark RDD operations.

UNIT-V:
NOSQL Data Management for Big Data and Visualization- NoSQL Databases: Schema-less Models‖: Increasing Flexibility for Data Manipulation, Key Value Stores, Document Store, Tabular Stores, Object Data Stores, Graph Databases Hive, Sharding, Hbase, Analyzing big data with twitter, Big data for E-Commerce Big data for blogs, Review of Basic Data Analytic Methods using R.

Text Books:
1. Anand Rajaraman and Jeffrey David Ullman, "Mining of Massive Datasets", Cambridge University Press, 2012.
2. David Loshin, "Big Data Analytics: From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph", Morgan Kaufmann/El sevier Publishers, 2013.

Reference Books:
1. MichaelBerthold,DavidJ.Hand,"IntelligentDataAnalysis",Springer,2007.
2. TomWhite"Hadoop:TheDefinitiveGuide"ThirdEdition,O'reillyMedia,2012.
3. Chris Eaton, Dirk DeRoos, Tom Deutsch, George Lapis, Paul Zikopoulos, "Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data", McGrawHill Publishing,2012.
4. BillFranks,"TamingtheBigDataTidalWave:FindingOpportunitiesinHugeD ata Streams with Advanced Analytics", John Wiley& sons,2012.
5. GlennJ.Myatt,"MakingSenseofData",JohnWiley&Sons,2007.
6. Pete Warden, "Big Data Glossary", O'Reilly,2011.
7. Jiawei Han, Micheline Kamber "Data Mining Concepts and Techniques", 2nd Edition, Elsevier, Reprinted2008.

JNTUA College of Engineering (Autonomous) Ananthapuramu
Department of Computer Science and Engineering
MCA (R20)
Program Elective-4

Software Defined Networks

SEMESTER - IV L-T-P-C: 3-0-0-3

Course Objectives:

- To learn threats and risks within context of the cyber security architecture.
- Student should learn and Identify security tools and hardening techniques.
- To learn types of incidents including categories, responses and timelines for response.

Course Outcomes: At the end of the course, student will be able to

	Course Outcomes	Knowledge Level (K)#
CO1	Explain the key benefits of SDN by the separation of data and control planes	K5
CO2	Interpret the SDN data plane devices and Openflow Protocols	K3
CO3	Apply the operation of SDN control plane with different controllers	K3
CO4	Apply techniques that enable applications to control the underlying network using SDN	K4
CO5	Design Network Functions Virtualization components and their roles in SDN	K6

#Based on suggested Revised BTL

Mapping of course outcomes with program outcomes

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1		1						3		
CO2	1		3							
CO3			2					1		
CO4										
CO5			2							

(Levels of Correlation: 1-low, 2-medium 3-high)

UNIT-I:

SDN Background and Motivation-Evolving network requirements-The SDN Approach: Requirements, SDN Architecture, Characteristics of Software-Defined Networking, SDN and NFV-Related Standards: Standards-Developing Organizations, Industry Consortia, Open Development Initiatives.

UNIT-II:

SDN Data plane and OpenFlow-SDN data plane: Data plane Functions, Data plane protocols, Openflow logical network Device: Flow table Structure, Flow Table Pipeline, The Use of Multiple Tables, Group Table- OpenFlow Protocol.

UNIT-III:

SDN Control Plane-SDN Control Plane Architecture: Control Plane Functions, Southbound Interface, Northbound Interface, Routing, ITU-T Model- Open Daylight-REST- Cooperation and Coordination among Controllers.

UNIT-IV:

SDN Application Plane-SDN Application Plane Architecture: Northbound Interface, Network Applications, User Interface- Network Services Abstraction Layer :Abstractions in SDN, Frenetic-Traffic Engineering Measurement and Monitoring- Security- Data Center Networking- Mobility and Wireless.

UNIT-V:

Network Functions Virtualization- Background and Motivation for NFV-Virtual Machines- NFV Concepts: Simple Example of the Use of NFV,NFV Principles, High-Level NFV Framework, NFV Benefits and Requirements-NFV Reference Architecture: NFV Management and Orchestration

Text Books:

1. William Stallings, “Foundations of Modern Networking”,Pearson Ltd.,2016.
2. Software Defined Networks: A Comprehensive Approach by Paul Goransson and Chuck Black,Morgan Kaufmann Publications, 2014
3. SDN - Software Defined Networks by Thomas D. Nadeau & Ken Gray, O'Reilly, 2013

Reference Books:

1. Feamster, Nick, Jennifer Rexford, and Ellen Zegura. "The road to SDN: an intellectual historyof programmable networks." ACM SIGCOMM Computer Communication Review 44.2 (2014): 87-98.
2. Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." Proceedingsof the IEEE 103.1 (2015): 14-76.

Web Reference:

https://www.coursera.org/learn/sdn