ACADEMIC REGULATIONS
CURRICULUM STRUCTURE
and
DETAILED SYLLABI

for

Two Year PG Programme

MCA
(Master of Computer Applications)

(Applicable for batches admitted from 2020)

Department of Computer Science & Engineering
University College of Engineering Kakinada(A), JNTU Kakinada

VISION OF THE INSTITUTE

To be a premier institute of excellence developing highly talented holistic human capital that contributes to the nation through leadership in technology and innovation through engineering education.

MISSION OF THE INSTITUTE

1. To impart Personnel Skills and Ethical Values for Sustainable Development of the Nation.

2. To create Research & Industry oriented centre of excellence in all engineering disciplines.

3. To be a renowned IPR generator and repository for innovative technologies.

4. To develop Research and Industry oriented technical talent.

5. To benchmark globally the academic & research output.

VISION OF THE DEPARTMENT

Department of Computer Science and Engineering strives rigorously to impart intellectual environment with global standards that fosters the search for new knowledge in a highly dynamic computing-centric society through research & applied efforts.

MISSION OF THE DEPARTMENT

- To provide quality education in both theoretical and applied foundations of computer science and train the students to solve the real world problems effectively thus enhancing their potential for high quality careers.
- To facilitate the students and faculty to inculcate the research culture to advance the state art of computer science and integrate research innovations in multi-disciplinary fields.
- To equip student / faculty with excellent teaching learning capabilities through advanced learning tools and technologies.
- To produce students with critical thinking and lifelong learning capabilities to apply their knowledge to uplift the living standards of the society.
- To produce students with enriched skill set, professional behavior, strong ethical values and leadership capabilities so as to work with commitment for the progress of the nation.

Programme Educational Objectives (PEOs)

After 3-5 years of graduation the graduate shall be able to

PEO 1 To produce IT professionals with in depth knowledge in software design, programming and analytical skills to cater the challenging industrial and societal needs in an effective manner with ethics and human values.

PEO 2 To produce Sustained learner to bring out creative and innovative ideas by addressing the research issues/ to serve as faculty for IT education.

PEO 3 To produce entrepreneurs in IT with good interpersonal and managerial skills to survive in multidisciplinary fields.

Programme Outcomes (POs)

After completion of MCA course students will attain the following programme outcomes

Programme Outcomes

PO1 Apply the knowledge of mathematics and computing fundamentals to various real life applications

PO2 Design and develop applications to solve complex problems for the betterment of the society

PO3 Integrate and apply efficiently the contemporary IT tools to all computer applications

PO4 Function effectively both as a team leader and team member on multi disciplinary projects with best professional ethical practices, positive attitude and social concern.

PO5 Communicate effectively and present technical information in oral and written reports

PO6 Involve in perennial learning for a continued career development and progress as a computer professional

PO7 Able to devise and conduct experiments, interpret data and provide well informed conclusions

PO8 Develop an ability to identify, analytically study, frame and develop computer applications

PO9 Apply the inherent skills with absolute focus to function as an successful entrepreneur

PO10 Understand and commit to professional ethics and cyber regulations for professional computing practices

Programme Specific Outcomes (PSOs)

PSO1 Design, develop and implement interdisciplinary application software projects to meet the demands of industry requirements using modern tools and technologies.

PSO2 Analyze the societal needs to provide novel solutions through technological based research.

PSO3 Inculcate employability and entrepreneur skills among students who can develop customized solutions for small to large Enterprises

REVISED Bloom's Taxonomy Action Verbs

Definitions	I. Remembering	II. Understanding	III. Applying	IV. Analyzing	V. Evaluating	VI. Creating
Bloom's Definition	Exhibit memory of previously learned material by recalling facts, terms, basic concepts, and answers.	Demonstrate understanding of facts and ideas by organizing, comparing, translating, interpreting, giving descriptions, and stating main ideas.	Solve problems to new situations by applying acquired knowledge, facts, techniques and rules in a different way.	Examine and break information into parts by identifying motives or causes. Make inferences and find evidence to support generalizations.	Present and defend opinions by making judgments about information, validity of ideas, or quality of work based on a set of criteria.	Compile information together in a different way by combining elements in a new pattern or proposing alternative solutions.
Verbs	• Choose • Define • Find • How • Label • List • Match • Name • Omit • Recall • Relate • Select • Show • Spell • Tell • What • When • Where • Which • Who • Why	• Classify • Compare • Contrast • Demonstrate • Explain • Extend • Illustrate • Infer • Interpret • Outline • Relate • Rephrase • Show • Summarize • Translate	• Apply • Build • Choose • Construct • Develop • Experiment with • Identify • Interview • Make use of • Model • Organize • Plan • Select • Solve • Utilize	• Analyze • Assume • Categorize • Classify • Compare • Conclusion • Contrast • Discover • Dissect • Distinguish • Divide • Examine • Function • Inference • Inspect • List • Motive • Relationships • Simplify • Survey • Take part in • Test for • Theme	• Agree • Appraise • Assess • Award • Choose • Compare • Conclude • Criteria • Criticize • Decide • Deduct • Defend • Determine • Disprove • Estimate • Evaluate • Explain • Importance • Influence • Interpret • Judge • Justify • Mark • Measure • Opinion • Perceive • Prioritize • Prove • Rate • Recommend • Rule on • Select • Support • Value	• Adapt • Build • Change • Choose • Combine • Compile • Compose • Construct • Create • Delete • Design • Develop • Discuss • Elaborate • Estimate • Formulate • Happen • Imagine • Improve • Invent • Make up • Maximize • Minimize • Modify • Original • Originate • Plan • Predict • Propose • Solution • Solve • Suppose • Test • Theory

MCA Programme (R20)

Course Structure & Syllabi

MCA Course Structure

I Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	R20MCA1101	Data Structures	PC	3	0	0	3
2	R20MCA1102	Computer Organization	PC	3	0	0	3
3	R20MCA1103	Database Management Systems	PC	3	0	0	3
4	R20MCA1104	Operating Systems	PC	3	0	0	3
5	R20MCA1105	Mathematical and Statistical Foundations	BS&H	3	1	0	4
6	R20MCA1106	Data Base Management Systems Lab	PC	0	0	3	1.5
7	R20MCA1107	Data Structures using C Lab	PC	0	0	4	2
8	R20MCA1108	Operating Systems and Linux Lab	PC	0	0	3	1.5
			Total	15	1	10	21

II Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	R20MCA1201	Computer Networks	PC	3	0	0	3
2	R20MCA1202	Principles of Cryptography and Network Security	PC	3	0	0	3
3	R20MCA1203	Object oriented Programming using JAVA	PC	3	0	0	3
4	R20MCA1204	Software Engineering	PC	3	0	0	3
5	R20MCA1205	Design and Analysis of Algorithm	PC	3	0	0	3
6	R20MCA1206	**Program Elective-1** 1. Artificial Intelligence 2. Advanced Unix Programming 3. Data Warehousing and Data mining 4. MOOCS- 1 (NPTEL /SWAYAM)	PC / PE	3	0	0	3
7	R20MCA1207	Object Oriented Programming using JAVA Lab	PC	0	0	3	1.5
8	R20MCA1208	Networks and Security lab	PC	0	0	3	1.5
9	R20MCA1209	Employability Skills-1$	AC	1	0	0	0
10	R20MCA1210	Physical Fitness Activities*$	AC	0	0	0	0
			Total	16	1	8	21

*This will be conducted in Zero Hour.$ Internal Evaluation

III Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	R20MCA2101	Machine Learning with Python	PC	3	0	0	3
2	R20MCA2102	Web Technologies	PC	3	0	0	3
3	R20MCA2103	Unified Modeling Languages	PC	3	0	0	3
4	R20MCA2104	**Program Elective-2** 1. Cloud Computing 2. Image Processing 3. Internet of Things 4. MOOCS-2 (NPTEL /SWAYAM)	PE	3	0	0	3
5	R20MCA2105	Human Resource Management	BS&H	3	0	0	3
6	R20MCA2106	Web Technologies Lab	PC	0	0	3	1.5
7	R20MCA2107	Machine Learning with Python Lab	PC	0	0	4	2
8	R20MCA2108	Unified Modeling Languages Lab	PC	0	0	3	1.5
9	R20MCA2109	Employability Skills – 2$	AC	1	0	0	0
10	R20MCA2110	Internship / Industry Oriented Mini Project#	MC	0	0	0	2
			Total				22

This can be done during *semester break* and evaluated at the end of 3rd Sem.

IV Semester

S.No	Course Code	Course Name	Category	L	T	P	Credits
1	R20MCA2201	**Program Elective-3** 1. Multimedia Application Development 2. Block Chain technologies 3. Design Patterns 4. MOOCs-3 (NPTEL/SWAYAM) 1.Full Stack Technologies 2. Any recommended course	PE	3	0	0	3
2	R20MCA2202	**Program Elective-4** 1. Big Data Analytics 2. Cyber security 3. Software Defined Networks 4. MOOCs-4 (NPTEL/SWAYAM) 1.Data Science 2. Any recommended course	PE	3	0	0	3
3	R20MCA2203	Project Work/ Dissertation		0	0	0	10
			Total	6	0	0	16

***Students going for Industrial Project/Thesis will complete these courses through MOOCs (or even in earlier semester)**

MCA I Semester

Data Structures

Code: R20MCA1101

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.

MCA I Semester

Computer Organization
Code: R20MCA1102

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 IV:

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.

MCA I Semester

Data Base Management Systems

Code: R20MCA1103

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

MCA I Semester

Operating Systems
Code: R20MCA1104

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

MCA I Semester

Mathematical and Statistical Foundations

Code: R20MCA1105

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 without 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

MCA I Semester

Data Base Management Systems Lab

Code: R20MCA1106

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

MCA I Semester

Data Structures using C Lab

Code: R20MCA1107

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

MCA I Semester

Operating Systems and Linux Lab

Code: R20MCA1108

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

MCA II Semester

Computer Networks

Code: R20MCA1201

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
3. An Engineering approach to compute networking, Kesha, Pearson

MCA II Semester

Principles of Cryptography and Network Security
Code: R20MCA1202

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

MCA II Semester

Object Oriented Programming using JAVA

Code: R20MCA1203

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
2. Big Java2,3/e, Cay.S. Horstmann, Wiley
3. Object Oriented Programming through Java, P.Radha Krishna, University Press
4. JAVA& Object Orientation an Introduction, 2/e, John Hunt, Springer
5. Introduction to JAVA Programming, 7/e, Y. Daniel Liang, Pearson., TMH

MCA II Semester

Software Engineering
Code: R20MCA1204

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.
5. An Integrated Approach to Software Engineering, Pankaj Jalote, Published by Narosa Publishing House, 3rd Edition, 2007

MCA II Semester

Design and Analysis of Algorithms
Code: R20MCA1205

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

MCA II Semester

Artificial Intelligence
Code: R20MCA1206

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

MCA II Semester

Advanced UNIX Programming

Code: R20MCA1206

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

MCA II Semester

Data Warehousing and Data Mining

Code: R20MCA1206

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

MCA II Semester

Object Oriented Programming using JAVA Lab

Code: R20MCA1207

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

MCA II Semester

Networks and Security Lab
Code: R20MCA1208

Course Objectives:

- To learn basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- To understand and implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher.

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

	Course Outcomes	Knowledge Level (K)#
CO1	Apply the knowledge of symmetric cryptography to implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher	**K3**
CO2	Demonstrate the different algorithms like DES, BlowFish, and Rijndael, encrypt the text "Hello world" using Blowfish Algorithm.	**K2**
CO3	Analyze and implement public key algorithms like RSA, Diffie-Hellman Key Exchange mechanism, the message digest of a text using the SHA-1 algorithm	**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			1					1
CO3			2						1	

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

List of Experiments:

1. Implement the data link layer farming methods such as character stuffing and bit stuffing.
2. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
3. Implement Dijkstra's algorithm to compute the Shortest path thru a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table art each node using distance vector routing algorithm
5. Take an example subnet of hosts. Obtain broadcast tree for it.

6. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should XOR each character in this string with 0 and displays the result.
7. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result
8. Write a Java program to perform encryption and decryption using the following algorithms:
    a) Ceaser Cipher
    b) Substitution Cipher
    c) Hill Cipher
9. Write a Java program to implement the DES algorithm logic
10. Write a C/JAVA program to implement the BlowFish algorithm logic
11. Write a C/JAVA program to implement the Rijndael algorithm logic.
12. Using Java Cryptography, encrypt the text "Hello world" using BlowFish.
13. Create your own key using Java key tool.

- Write a Java program to implement RSA Algorithm
- Write a Java program to implement Public key Algorithm like El Gamal
- Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
- Calculate the message digest of a text using the SHA-1 algorithm in JAVA.
- Calculate the message digest of a text using the MD5 algorithm in JAVA.

MCA III Semester

Machine Learning with Python

Code: R20MCA2101

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

MCA III Semester

Web Technologies
Code: R20MCA2102

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
4. Beginning Web Programming-Jon Duckett WROX.

MCA III Semester

Unified Modeling Language

Code: R20MCA2103

Course Objectives:

- To understand the Object-based view of Systems
- To develop robust object-based models for Systems
- To explain necessary skills to handle complexity in software design

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

Course Outcomes		Knowledge Level (K)#
CO1	Identify OOAD practices in developing software projects with management perspective	**K3**
CO2	Identify, analyze, and model structural and behavioral concepts of the system.	**K4**
CO3	Analyze and apply use of concepts in Collaboration diagrams.	**K4,5**
CO4	Develop, explore the behaviour model into various scenarios and applications	**K3**
CO5	Apply the concepts of architectural design for deploying the code for software.	**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							1		1
CO3		2					1			
CO4		1						2		
CO5	2							1	1	

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

UNIT- I:

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

UNIT-II:

Basic structural Modeling: Classes, relationships, common mechanisms, diagrams, **Advanced structural modeling:** advanced relationships, interfaces, types & roles, packages, instances, **Class & object diagrams**: Terms, concepts, examples, modeling techniques, class & Object diagrams.

UNIT-III:
Collaboration diagrams: Terms, Concepts, depicting a message, polymorphism in collaboration diagrams, iterated messages, use of self in messages, **Sequence diagrams:** Terms, concepts, differences between collaboration and sequence diagrams, depicting synchronous messages with/without priority call back mechanism broadcast message.

UNIT- IV:
Behavioral Modeling: Interactions, use cases, use case diagrams, activity diagrams, **Advanced Behavioral Modeling**: Events and signals, state machines, processes & threads, time and space, state chart diagrams.

UNIT–V:
Architectural Modeling: Terms, concepts, examples, modeling techniques for component diagrams and deployment diagrams.

Text Books:

1. The Unified Modeling Language User Guide, 2nd Edition, Grady Booch, Rambaugh, Ivar Jacobson, PEA, 2005
2. Fundamentals of Object Oriented Design in UML, 1st Edition, Meilir Page Jones, Addison Wesley,2000

Reference Books:

1. Head First Object Oriented Analysis & Design, 1st Edition, Mclaughlin, SPD O'Reilly, 2006
2. Object oriented Analysis & Design Using UML, 1st Edition, Mahesh, PHI, 2008
3. The Unified Modeling Language Reference Manual, 2nd Edition, Rambaugh, Grady Booch, etc., PEA, 2004

MCA III Semester

Cloud computing

Code: R20MCA2104

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.

MCA III Semester

Image Processing

Code: R20MCA2104

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

MCA III Semester

Internet of Things

Code: R20MCA2104

Course Objectives:

- Identify problems that are amenable to solution by AI methods, and which AI methods may be suited to solving a given problem.
- Formalize a given problem in the language/framework of different AI methods (e.g., as a search problem, as a constraint satisfaction problem, as a planning problem, as a Markov decision process, etc).
- Design and carry out an empirical evaluation of different algorithms on problem formalization, and state the conclusions that the evaluation supports.

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

Course Outcomes		Knowledge Level (K)#
CO1	Explain the definition and usage of the term 'the internet of things' in different contexts	**K2**
CO2	Discover the various network protocols used in IoT	**K2**
CO3	Define the role of big data, cloud computing and data analytics in a typical IoT system.	**K3**
CO4	Compare and contrast the threat environment based on industry and/or device type	**K2**
CO5	Design a simple IoT system made up of sensors, wireless network connection, data analytics and display/actuators, and write the necessary control software	**K6**

Mapping of course outcomes with program outcomes

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

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

UNIT I:

The Internet of Things: An Overview of Internet of things, Internet of Things Technology, behind Io Ts Sources of the Io Ts, M2M Communication, Examples of IoTs, Design Principles For Connected Devices Internet Connectivity Principles, Internet connectivity, Application Layer Protocols: HTTP, HTTPS, FTP, Telnet.

UNIT II:
Business Models for Business Processes in the Internet of Things ,IoT/M2M systems LAYERS AND designs standardizations ,Modified OSI Stack for the IoT/M2M Systems ,ETSI M2M domains and High-level capabilities ,Communication Technologies, Data Enrichment and Consolidation and Device Management Gateway Ease of designing and affordability

UNIT III:
Design Principles for the Web Connectivity for connected-Devices, Web Communication protocols for Connected Devices, Message Communication protocols for Connected Devices, Web Connectivity for connected-Devices.

UNIT IV:
Data Acquiring, Organizing and Analytics in IoT/M2M, Applications /Services /Business Processes, IOT/M2M Data Acquiring and Storage, Business Models for Business Processes in the Internet Of Things, Organizing Data, Transactions, Business Processes, Integration and Enterprise Systems.

UNIT V:
Data Collection, Storage and Computing Using a Cloud Platform for IoT/M2M Applications/Services, Data Collection, Storage and Computing Using cloud platform Everything as a service and Cloud Service Models, IOT cloud-based services using the Xively (Pachube/COSM), Nimbits and other platforms Sensor, Participatory Sensing, Actuator, Radio Frequency Identification, and Wireless, Sensor Network Technology, Sensors Technology ,Sensing the World.

Text Books:

1. Internet of Things: Architecture, Design Principles And Applications, 1st ed, Rajkamal, McGraw Hill Higher Education, 2017.
2. Internet of Things, 1st ed, A.Bahgya and V.Madisetti, Univesity Press, 2014

Reference Books:

1. Designing the Internet of Things, 1st ed, Adrian McEwen and Hakim Cassimally, Wiley, 2013.
2. Getting Started with the Internet of Things, 1st ed, CunoPfister , Oreilly, 2011.

MCA III Semester

Human Resource Management

Code: R20MCA2105

Course Objectives:

- Contribute to the development, implementation, and evaluation of employee recruitment, selection, and retention plans and processes.
- Administer and contribute to the design and evaluation of the performance management program.
- Develop, implement, and evaluate employee orientation, training, and development programs.
- Facilitate and support effective employee and labour relations in both non-union and union environments.

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

Course Outcomes		Knowledge Level (K)#
CO1	Explain the importance of human resources and their effective management in organizations	**K2**
CO2	Demonstrate a basic understanding of different tools used in forecasting and planning, human resource need.	**K2**
CO3	Describe the meanings of terminology and tools used in managing employees effectively	**K4**
CO4	Make use of Record governmental regulations affecting employees and employers	**K3**
CO5	Analyze the key issues related to administering the human elements such as motivation, compensation, appraisal, career planning, diversity, ethics, and training	**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						
CO2	1		2							
CO3		1						2		
CO4				1	1					
CO5					1					3

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

UNIT I:

HRM: Significance - Definition and Functions – evolution of HRM- Principles - Ethical Aspects of HRM- - HR policies, Strategies to increase firm performance - Role and position of HR department –aligning HR strategy with organizational strategy - HRM at global perspective -challenges – cross-cultural problems – emerging trends in HRM.

UNIT II:

Investment perspectives of HRM: HR Planning – Demand and Supply forecasting - Recruitment and Selection- Sources of recruitment - Tests and Interview Techniques - Training and Development – Methods and techniques – Training evaluation - retention - Job Analysis – job description and specifications - Management development - HRD concepts.

UNIT III:

Wage and Salary Administration: Concept- Wage Structure- Wage and Salary Policies- Legal Frame Work- Determinants of Payment of Wages- Wage Differentials - Job design and Evaluation- Incentive Payment Systems. Welfare management: Nature and concepts – statutory and non-statutory welfare measures – incentive mechanisms.

UNIT IV:

Performance Evaluation: Importance – Methods – Traditional and Modern methods – Latest trends in performance appraisal - Career Development and Counseling- Compensation, Concepts and Principles- Influencing Factors- Current Trends in Compensation- Methods of Payments - compensation mechanisms at international level.

UNIT V:

Managing Industrial Relations: Trade Unions - Employee Participation Schemes-Collective Bargaining–Grievances and disputes resolution mechanisms – Safety at work – nature and importance – work hazards – safety mechanisms - Managing work place stress.

Text Books:

1. K Aswathappa: “Human Resource and Personnel Management”, Tata McGraw Hill, New Delhi, 2013
2. N.Sambasiva Rao and Dr. Nirmal Kumar: “Human Resource Management and Industrial Relations”, Himalaya Publishing House, Mumbai
3. Mathis, Jackson,Tripathy:“Human Resource Management:Asouth-Asin Perspective”, Cengage Learning, New Delhi, 2013
4. Subba Rao P: “Personnel and Human Resource Management-Text and Cases”, Himalaya Publications, Mumbai, 2013.
5. Madhurima Lall, Sakina Qasim Zasidi: “Human Resource Management”, Excel Books, New Delhi, 2010 www.universityupdates.in || www.android.universityupdates.in || www.ios.universityupdates.in www.universityupdates.in || www.android.unive

MCA III Semester

Web Technologies Lab

Code: R20MCA2106

Course Objectives:

- To implement the web pages using HTML and apply styles.
- Able to develop a dynamic webpage by the use of java script.
- Design to create structure of web page, to store the data in web document, and transport information through web.
- Able to write a well formed / valid XML document.

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

Course Outcomes		Knowledge Level (K)#
CO1	Create dynamic and interactive web pages using HTML, CSS & Java Script	**K6**
CO2	Experiment with Learn and implement XML concepts	**K3**
CO3	Develop web applications using PHP	**K3**
CO4	Show the Install Tomcat Server and execute client-server programs	**K2**
CO5	Implement programs using Ruby programming	**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							
CO2	1	1						1		
CO3					1		2			
CO4	1	2							1	
CO5			1				3			

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

Experiment 1:

Develop static pages (using HTML and CSS) of an online book store. The pages should resemble: www.flipkart.com The website should consist the following pages.

a) Home page
b) Registration and user Login
c) User Profile Page
d) Books catalog
e) Shopping Cart
f) Payment By credit card
g) Order Conformation

Experiment 2:
Create and save an XML document on 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.

Experiment 3:
Write a PHP script to merge two arrays and sort them as numbers, in descending order.

Experiment 4:
Write a PHP script that reads data from one file and write into another file.

Experiment 5:
Write a PHP script to print prime numbers between 1-50.

Experiment 6:
Validate the Registration, user login, user profile and payment by credit card pages using JavaScript.

Experiment 7:
Write a PHP script to: a. Find the length of a string. b. Count no of words in a string. c. Reverse a string. d. Search for a specific string.

Experiment 8:
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.

Experiment 9:
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.

Experiment 10:
Install a database(Mysql or Oracle). Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form). Practice 'JDBC' connectivity. Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries. Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page .

Experiment 11:
Write a JSP which does the following job: Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate

the user when he submits the login form using the user name and password from the database.

Experiment 12:
Create a simple visual bean with a area filled with a color. The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false. The color of the area should be changed dynamically for every mouse click.

MCA III Semester

Machine Learning with Python Lab
Code: R20MCA2107

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.

MCA III Semester

Unified Modeling Languages Lab

Code: R20MCA2108

Course Objectives:

- To know the practical issues of the different object oriented analysis and design concepts
- Explain the art of object oriented software analysis and design
- Apply forward and reverse engineering of a software system

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

Course Outcomes		Knowledge Level (K)#
CO1	Make Use OOAD and UML concepts to identify Classes, Use Cases and their relationships	**K2**
CO2	Develop Use case diagrams and Class Diagrams	**K3**
CO3	Construct Interaction diagrams and packages	**K2**
CO4	Develop State chart, Activity, Component and Deployment Diagrams	**K3**
CO5	Experiment with Construct case studies using UML diagrams	**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		2			
CO2			3							
CO3		1					1		1	
CO4		1						3		
CO5							3			

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

Note: For performing the experiments consider any case study (ATM / Banking / Library / Hospital management systems)

Experiment 1:

Familiarization with Rational Rose or Umbrella environment

Experiment 2:

a) Identify and analyze events
b) Identify Use cases
c) Develop event table

Experiment 3:

a) Identify & analyze domain classes
b) Represent use cases and a domain class diagram using Rational Rose
c) Develop CRUD matrix to represent relationships between use cases and problem domain classes

Experiment 4:

a) Develop Use case diagrams
b) Develop elaborate Use case descriptions & scenarios
c) Develop prototypes (without functionality)

Experiment 5:

a) Develop system sequence diagrams and high-level sequence diagrams for each use case
b) Identify MVC classes / objects for each use case
c) Develop Detailed Sequence Diagrams / Communication diagrams for each use case showing interactions among all the three-layer objects

Experiment 6:

a) Develop detailed design class model (use GRASP patterns for responsibility assignment)
b) Develop three-layer package diagrams for each case study

Experiment 7:

a) Develop Use case Packages
b) Develop component diagrams
c) Identify relationships between use cases and represent them
d) Refine domain class model by showing all the associations among classes

Experiment 8:

Develop sample diagrams for other UML diagrams - state chart diagrams, activity diagrams and deployment diagrams

MCA IV Semester

Multimedia Application Development
Code: R20MCA2201

Course Objectives:

- To analyze and explain various technologies involved to support multimedia application development.
- To understand multimedia authoring and Understanding the constraints on multimedia systems and the range of technologies available to multimedia systems designers and integrators.
- To demonstrate how the quality of multimedia systems is perceived and how this Relate to the design of multimedia input, output and editing systems.
- To distinguish compression principles and different compression technique and to know the mathematics involved in digital and analog conversion of components of multimedia.
- To design and develop multimedia systems according to the requirements of multimedia application and understand the particular issues of virtual reality.

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

Course Outcomes		Knowledge Level (K)#
CO1	Achieve an in-depth understanding of the impact of multimedia on personal and distributed computer systems, the range of media types and tools to support their digital conversion and manipulation of images, audio and video and their compression.	K1
CO2	Design and implementation of hypermedia & multimedia systems.	K6
CO3	Develop an interactive multimedia application to display their ability to use multimedia tools including multimedia authoring.	K6
CO4	Evaluate and Discuss Excellent and ineffective interactive multimedia design.	K5
CO5	Design and implement a number of multimedia network applications and Evaluate and discuss virtual reality systems.	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					
CO2		3								
CO3			3					2		
CO4	2									
CO5			1							

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

UNIT- I:

Multimedia Authoring, Graphics and Image and Data Representation: What is Multimedia, Multimedia and Hypermedia, World Wide Web, Overview of Multimedia Software Tools, Graphics/Image Data Types, File Formats, **Color in Image and Video:** Color Science, Color Models in Images, Color Models in Video.

UNIT- II:

Fundamental Concepts in Video and Digital Audio: Types of Video Signals, Analog Video, and Digital Video, Digitization of Sound, MIDI, Quantization and Transmission of Audio.

UNIT-III:

Action Script I: Action Script Features, Object-Oriented Action Script, Data types and Type Checking, Classes, Authoring an Action Script Class, **Action Script II:** Inheritance, Authoring an Action Script 2.0 Subclass, Interfaces, Packages, Exceptions, **Application Development:** An OOP Application Frame work, Using Components with Action Script Movie Clip Subclasses.

UNIT- IV

Multimedia Data Compression: Lossless Compression Algorithms: Run-Length Coding, Variable Length Coding, Dictionary Based Coding, Arithmetic Coding, Lossless Image Compression, **Lossy Compression Algorithm:** Quantization, Transform Coding, Wavelet-Based Coding, Embedded Zero tree of Wavelet Coefficients, Set Partitioning in Hierarchical Trees (SPIHT), **Basic Video Compression Techniques:** Introduction to Video Compression, Video Compression Based on Motion Compensation, Search for Motion Vectors, **MPEG**: MPEG – 1,2,4 and 7, **Basic Audio Compression Techniques**: ADPCM, G. 726 ADPCM, Vocoders.

UNIT-V

Multimedia Networks: Basics of Multimedia Networks, **Multimedia Network Communications and Applications:** Quality of Multimedia Data Transmission, Multimedia over IP, Multimedia over ATM Networks, Transport of MPEG- 4, Media-on- Demand (MOD).

Text Books:

1. Fundamentals of Multimedia, Ze-NianLi , Mark S. Drew, PHI/ PEA, 2004
2. Essentials Action Script 2.0, Colin Moock, SPD O, Reilly, 2004

Reference Books:

1. Multimedia Applications, Steinmetz, Nahrstedt, Springer,2004
2. Multimedia Systems, Parag Havaldar, Gerard Medioni, cengage, 2006
3. Digital Multimedia, Nigel Chapman, Jenny Chapman, Wiley- Dreamtech, 2009

Web References:

https://onlinecourses.swayam2.ac.in/nou20_cs05/preview

MCA IV Semester

Block Chain Technologies

Code: R20MCA2201

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).
3. 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).

MCA IV Semester

Design Patterns
Code: R20MCA2201

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.

MCA IV Semester

Big Data Analytics
Code: R20MCA2202

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:FindingOpportunitiesinHugeData 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.

MCA IV Semester

Cyber Security
Code: R20MCA2202

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	Apply cyber security architecture principles.	K3
CO2	Demonstrate the risk management processes and practices.	K2
CO3	Appraise cyber security incidents to apply appropriate response	K5
CO4	Distinguish system and application security threats and vulnerabilities.	K4
CO5	Identify security tools and hardening techniques	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									1
CO2		2								
CO3				2						3
CO4					2			1		
CO5					3					

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

UNIT–I:

Introduction to Cyber Security-Cyber security objectives, roles, differences between information security and cyber security, Cyber security principles**-** confidentiality, integrity, availability, authentication and non-repudiation

UNIT–II:

Information Security within Lifecycle Management-Lifecycle management landscape, Security architecture processes, Security architecture tools, Intermediate lifecycle management concepts, **Risks & Vulnerabilities-**Basics of risk management, Operational threat environments, Classes of attacks

UNIT–III:

Incident Response-Incident categories, Incident response,Incident recovery, **Operational security protection**-Digital and data assets, ports and protocols, Protection technologies, Identity and access Management, configuration management

UNIT–IV:

Threat Detection and Evaluation Monitoring-Vulnerability management, Security logs and alerts, Monitoring tools and appliances, **Analysis-**Network traffic analysis, packet capture and analysis

UNIT–V:

Introduction to backdoor System and security-Introduction to metasploit, backdoor, demilitarized zone (DMZ), Digital signature, Brief study on Harding of operating system.

Text Books:

1. NASSCOM: Security Analyst Student Hand Book, Dec 2015
2. Information Security Management principles, David Alexander, Amanda Finch, David Sutton, BCS Publishers, 2013

Reference Books:

1. Cyber Security Fundamentals-Cyber Security, Network Security and Data Governance Security, 2nd Edition, ISACA Publishers, 2019

MCA IV Semester

Software Defined Networks

Code: R20MCA2202

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:

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