Scheme and Syllabus
of
Master of Computer Applications (MCA)

2 YEARS PROGRAMME

CHOICE BASED CREDIT BASED SYSTEM

(70:30)

(w.e.f. session 2020-21)

Department of Computer Science& Engineering
Guru Jambheshwar University of Science & Technology, Hisar

Bridge Course(s) for MCA 2-Year Programme

(For students having non-IT background only)

Course No.	Course	C	L	T	P	Max. Marks	Exams (Internal)
MCA-BC-01	MCA Bridge Course – I	3	3	-	-	100	3 Hours
MCA-BC-02	MCA Bridge Course – II	3	3	-	-	100	3 Hours
	Total					200	

Note:

1. Both courses are qualifying in nature. However, no student will be awarded MCA degree without qualifying them.
2. Duration – Three/Four weeks
3. Both bridge courses shall be completed by the student(s) as prescribed by the Department/University. MCA degree shall not be awarded unless the students successfully complete the bridge course(s). Bridge course(s) examination will be conducted by the Department/University. The student has to secure 40% marks in examination in order to pass the bridge course(s). The respective University Teaching Department/Affiliated College(s) shall arrange for the contact sessions for completing the bridge course(s). The University/ Affiliated College(s) shall not charge any additional fee for the conduct of bridge course(s). However, the contact classes for bridge course(s) shall count towards teaching workload.

MCA-BC-01
MCA Bridge Course I

L - T - P **Max. Marks- 100**
3 – 0 – 0 **Credits – 3**
Exams(I): 3 Hours

Note: - Total 09 Questions are to be set by the examiner. First question will be compulsory consisting of 5 short (each 4 marks) questions covering entire syllabus uniformly. In addition, 8 more questions will be set unit wise comprising 2 questions from each unit of the given syllabus. A candidate is required to attempt five questions in all selecting one question from each Unit - Including the compulsory question.

Course Objectives: The main objective of the course is to bridge the gap between courses studied by the students having non-IT background. The students taking this bridge course shall be taught in foundational mathematics, fundamental concepts of computers and C programming language.

Learning Outcomes: After successful completion of this bridge course, the students will be able to:

- Understand basic programming skills in C language and develop computer programs in C.
- Working knowledge of computer system, its components and OS.
- Understand the knowledge of mathematical structures used in computer science and computer applications.

Unit-I

Elements of C language: C character set, identifiers & keywords, data types: declaration & definition.
Operators: Arithmetic relational, logical, bitwise, unary, assignment and conditional operators & their hierarchy & associativity, Data input/output.
Control statements: Sequencing, Selection: if and switch statement; iteration, repetition: for, while, and do-while loop; break, continue, goto statement.

Unit-II

Functions in C language: Definition, prototype, passing parameters, recursion, Data structure: arrays, structures, union, string, data files.
Pointers: Delectation, operations on pointers, array of pointers, pointers to arrays.
C preprocessors

Unit-III

Computer Software: introduction, relationship between hardware and software, types of software, planning the computer program: purpose of program planning, algorithm, flowcharts, decision tables, pseudo codes, application software packages.
Operating System, types of Operating System; process, process states, major components of an OS - file system, scheduler, and device drivers. Basic tasks of an OS-file management, memory management, process management, handling input and output and controlling peripheral devices such as disk drives and printers.
Data Communications and Computer Networks: Introduction, data transmission modes, data transmission speed, transmission media, digital and analog transmission, the internet, multimedia.

Unit-IV

Foundational Mathematics: Types of numbers and their properties, natural numbers, whole numbers, integers, real numbers, rational numbers, irrational numbers, complex numbers, imaginary numbers.

Set theory: Basic concept, set types, Venn Diagrams, cardinality, and notation, Basic counting principles.

Group theory: Basic concept, Binary Operations, Properties of Binary Operations.

Graph theory: Introduction to graphs, types, Degree of vertex, degree sequence path, connectivity, adjacency matrix.

References:

1. C Programming, Yashwant Kanitkar, Let us C, BPB Publications.
2. Pradeep k. Sinha & Priti Sinha, Computer Fundamentals, BPB Publications
3. Behrouz, Frozen, Introduction to Data Communications and Networking- Tata MC-Graw Hill.
4. Rajaraman V, Fundamentals of Computers, PHI
5. Seymour Lipschutz, Marc Lars Lipson, Discrete mathematics, McGraw-Hill international editions, Schaum's series.

MCA-BC-02
MCA Bridge Course II

L - T - P **Max. Marks- 100**
3 – 0 – 0 **Credits – 3**
Exams(I): 3 Hours

Note: - Total 09 Questions are to be set by the examiner. First question will be compulsory consisting of 5 short (each 4 marks) questions covering entire syllabus uniformly. In addition, 8 more questions will be set unit wise comprising 2 questions from each unit of the given syllabus. A candidate is required to attempt five questions in all selecting one question from each Unit - Including the compulsory question.

Course Objectives: The main objective of the course is to bridge the gap between courses studied by the students having non-IT background. The students taking this bridge course shall be taught topics in C++ programming language, digital electronics and Computer organization.

Learning Outcomes: After successful completion of this bridge course, the students will be able to:

- Understand the concept of object orientation and programming skills in C++ language and develop simple computer programs in C++.
- Understand the concept of Boolean algebra and corresponding electronics
- Understand the working of computer system through its organisation in terms of components.

Unit-I

Object oriented concept: Data abstraction, encapsulation, classes and objects modularity, hierarchy, typing, concurrency, object-oriented methodology: advantages and disadvantages of OO methodologies. aggregation, generalization and inheritance, abstract class, meta data, object diagram.

Unit-II

C++ Programming: Data types, structures vs classes, static data and member function, constant parameters and destruction, dynamic objects, operator overloading, function overloading, abstract class, virtual class, inheritance, virtual functions, template functions & template classes, exception handling, I/O streams.

Unit-III

Digital Fundamentals: Information representation - number systems, codes, binary arithmetic operations; number systems - non positional number system, positional number system, number system conversion, fractional number conversion; computer codes - BCD code, EBCDIC code, ASCII code, binary arithmetic - addition, subtraction, multiplication, division; binary logic - Boolean algebra, Boolean functions, truth table, simplification of Boolean functions, digital logic gates.

Unit-IV

Computer Organisation: Combinational logic - adders, subtractors, encoder, decoder, multiplexer, demultiplexer and comparators; processor organisation - machine instructions, instruction cycles, instruction formats and addressing modes, microprogramming concepts, microprocessor sequence; sequential logic - flip flops, shift registers and counters; I/O organisation - I/O interface, interrupt structure, transfer of information between CPU, memory and I/O devices.

References:

1. Stroustrup, B., The C++ programming language, Addison –Wesley1993.
2. Balaguruswami, object oriented programming in C++ : Tata McGraw
3. Rumbaugh. J.et. al., Object oriented modeling and design, Prentice Hall of India 1998.
4. Pradeep K. Sinha & Priti Sinha, Computer Fundamentals, BPB Publications
5. Rajaraman V, Fundamentals of Computers, PHI
6. Mano. M. Morris Digital Logic & Computer systems Design, Prentice Hall of India Pvt. Ltd., 2000.

Scheme of Examination for
Master of Computer Applications (MCA)
Two-Year Programme under CBCS Scheme
w.e.f. Academic Session 2020-2021

MCA SEMESTER-I

Course Code	Course Title	Credit	Int.	Ext.	Total
MCA-11	Database Management System	3	30	70	100
MCA-12	Web Designing	3	30	70	100
MCA-13	Java Programming	3	30	70	100
MCA-14	Software Engineering	3	30	70	100
MCA-15	Computer Networks and Internet Protocols	3	30	70	100
MCA-16	Database Management System Lab	2	30	70	100
MCA-17	Web Designing Lab	2	30	70	100
MCA-18	Java Programming Lab	2	30	70	100
Total		**21**	**240**	**560**	**800**

MCA SEMESTER –II

Course No.	Course Title	Credit	Int.	Ext.	Total
MCA-21	Data Structures and Algorithms	3	30	70	100
MCA-22	Python Programming	3	30	70	100
MCA-23	Artificial Intelligence	3	30	70	100
MCA-24	Computer System Architecture	3	30	70	100
MCA-25	Discrete Mathematics and Optimization	3	30	70	100
MCA-26	Data Structures and Algorithm Lab	2	30	70	100
MCA-27	Python Programming Lab	2	30	70	100
MCA-28	Artificial Intelligence Lab	2	30	70	100
Total		**21**	**240**	**560**	**800**
Four to six week training at the end of second semester					

MCA SEMESTER-III

Course No.	Course Title	Credit	Int.	Ext.	Total
MCA-31	Machine Learning	3	30	70	100
MCA-32	Advanced Operating Systems	3	30	70	100
MCA-33	Data Analytics	3	30	70	100
MCA-34	Cyber Security	3	30	70	100
MCA-35	Elective – I	3	30	70	100
MCA-36	Machine Learning Lab	2	30	70	100
MCA-37	Linux and Shell Script Lab	2	30	70	100
MCA-38	Data Analytics Lab	2	30	70	100
MCA-39	Evaluation of Industrial Training	2	100	---	100
Total		**23**	**340**	**560**	**900**

MCA – 35 Elective – I List of Courses

MCA – 35(i) Computer Graphics
MCA – 35(ii) Theory of Computations
MCA – 35(iii) High Speed Networks
MCA – 35(iv) Any MOOC Course with the permission of chairperson from the list approved by department.

MCA SEMESTER-IV

Course No.	Course Title	Credit	Int.	Ext.	Total
MCA-41	IoT and Cloud Computing	3	30	70	100
MCA-42	Mobile Application Development	3	30	70	100
MCA-43	Elective – II	3	30	70	100
MCA-44	Elective – III	3	30	70	100
MCA-45	IoT and Cloud Computing Lab	2	30	70	100
MCA-46	Android Programming Lab	2	30	70	100
MCA-47	Project Work	6	30	70	100
Total		**22**	**210**	**490**	**700**

MCA – 43 Elective – II List of Courses

MCA – 43(i) Big Data Analytics
MCA – 43(ii) Software Project Management
MCA – 43(iii) Digital Image Processing
MCA – 43(iv) Any MOOC Course with the permission of chairperson from the list approved by department.

MCA – 44 Elective –III List of Courses

MCA – 44(i) Soft Computing
MCA – 44(ii) Compiler Design
MCA – 44(iii) Data Mining Techniques
MCA – 44(iv) Any MOOC Course with the permission of chairperson from the list approved by department.

Total Programme Credits
MCA 2–Year under CBCS
w.e.f. Academic Session 2020-2021

Semester	Max. Marks	Credits
I	800	21
II	800	21
III	900	23
IV	700	22
Programme Total	**3200**	**87**

Note: -

1. A Bridge course (qualifying in nature) of duration 3-4 weeks will be given to students of non-IT background before the commencement of first year/semester classes. No student will be awarded MCA degree without qualifying them.

2. Evaluation of Industrial Training is done by two members internal committee constituted by chairperson.

3. MOOCS through SWAYAM/NPTEL courses are also allowed to be offered as at least one elective course to the students as per the university guidelines.

MCA-11 Database Management System

General Course Information

Course Code: MCA-11 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite: Knowledge of UNIX, Windows, a programming language and data structures

About the Course:
This course includes a detailed coverage of principles of database design and models. Students learn querying a database using SQL, normalization techniques, transaction processing etc.

Course Outcomes: By the end of the course students will be able to:

CO1. **Describe** fundamental elements of Database Management System.
CO2. **Discuss** principles of relational Database modelling.
CO3. **Apply** SQL for designing queries for Relational Databases.
CO4. **Contrast** various concurrency control and recovery techniques with concurrent transactions in DBMS.
CO5. **Design** models of databases using ER modelling and normalization for real life applications.

Course Content

Unit - I

Overview: Overview of File Systems and Database Systems, Characteristics of the Data Base Approach, Database users, Advantages and Disadvantages of a DBMS, Responsibility of Database Administrator.
Data Base Systems Concepts and Architecture: DBMS architecture and various views of Data, Data Independence, Database languages, Data Models: Relational Database Model, Hierarchical Data Model, Network Data Model, Schemas and Instances.

Unit - II

E-R Model: Entity Types, Attributes & Keys, Relationships, Roles and Structural Constraints, E-R Diagrams, Reduction of an E-R Diagram to Tables. Relational Model and Query Language: Overview of Relational Database, Key Integrity Constraints, Relational Algebra, Relational Calculus, SQL fundamentals, Basic Operators, Missing information and NULL values, Advanced SQL features

Unit - III

Relational Database Design: Overview of normalization, Database Anomalies, Candidate and Super Key, Functional Dependencies, Integrity Constraints, Decomposition, Normal forms: First, Second, Third Normal, Boyce Codd, Normal Form, Multi-valued Functional Dependencies and Fourth

Normal Form, Join Dependencies and Fifth Normal Form, Denormalization.

Unit - IV

Concurrency Control Techniques: Overview of database Transactions, Transaction states, ACID properties of a Transaction, Transaction Recovery, Concurrency Control, Locking Techniques, Time-stamp ordering, Multi-version Techniques, Deadlock, Recovery Techniques in centralized DBMS.
DDBMS Design: Replication and Fragmentation Techniques.

Text and Reference Books:

1. Elmasri, R., and Navathe, S. B., Fundamentals of Database Systems, 3rd Edition, Addison Wesley, 2002.
2. Silberschatz, A., Korth, H. F., and Sudarshan, S., Database System Concepts, McGraw Hill, 2011.
3. Pannerselvam R., Database Management Systems, 2nd Edition, PHI Learning, 2011.
4. Desai, B. C., An Introduction to Database System, Galgotia Publication, 2010.
5. Leon, A., and Leon, M., Database Management Systems, Ist Edition, Vikas Publishing, 2009.
6. Mata-Toledo, R., Cushman, P., Sahoo, D., Database Management Systems, Schaums' Outline series, TMH, 2007.

MCA-12 Web Designing

General Course Information

Course Code: MCA-12 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite:
Basic knowledge of HTML, XML, ASP, JSP and Web Designing.

About the Course:
This course includes a detailed coverage of HTML. Students learn HTML, XML and design various web pages and ASP, JSP and its uses in web designing process with HTML. They study about Client Side Programming and Server Side Programming.

Course Outcomes: By the end of the course students will be able to:
CO1. **Describe** Web Designing Complete Process
CO2. **Discuss** Client Side and Server Side Programming.
CO3. **Apply** Stylesheets with HTML& ASP and JSP with HTML in web design.
CO4. **Contrast** All Markup Languages.
CO5. **Design** Web Pages using HTML and XML.

Course Contents

Unit - I

Information Architecture The Role of Information Architect, Collaboration and Communication, Organizing information, Organizational challenges, Organizing Web Sites and Intranets, Creating Cohesive Organization Systems, Designing Navigation Systems, Types of navigation Systems, Integrated Navigation Elements, Remote Navigation Elements, Designing Elegant Navigation Systems, Designing the Search Interface, Indexing the Right Stuff, Grouping Content, Conceptual Design; High-Level Architecture Blueprints, Architectural Page Mockups, Design Sketches.

Unit - II

Dynamic HTML and Web Designing HTML Basic Concepts, Good Web Design, Process of Web Publishing, Phases of Web Site development, Structure of HTML documents, HTML Elements - Core attributes, absolute and relative links, ordered and unordered lists, Linking Basics, Linking in HTML, Images and Anchors, Anchor Attributes, Image Maps, Semantic Linking Meta Information, Image Preliminaries, , Images as Buttons, Introduction to Layout: Backgrounds, Colors and Text, Fonts, Layout with Tables, Advanced Layout : Frames and layers, HTML and other media types, FORMS, Forms Control, New and emerging Form Elements.

Separating style from structure with style sheets: Internal style specifications within HTML, External linked style specification using CSS, page and site design considerations, Positioning with Style sheets.

Unit - III

Client side programming: Introduction to the JavaScript syntax, the JavaScript object model, Event handling, Output in JavaScript, Forms handling, miscellaneous topics such as cookies, hidden fields, and images; Applications.

Server side programming: Introduction to Server Side Technologies CGI / ASP / JSP, Programming languages for server Side Scripting, Configuring the server to support CGI, applications; Input/ output operations on the WWW, Forms processing, (using PERL/VBSCRIPT/JavaSCIPT)

Unit - IV

Java Server Pages and Active Server Pages: Basics, Integrating Script, JSP/ASP Objects and Components, configuring and troubleshooting,: Request and response objects, Retrieving the contents of a an HTML form, Retrieving a Query String, Cookies, Creating and Reading Cookies. Using application Objects and Events.

Overview of advance features of XML, XML Relationship between HTML, SGML, and XML, The future of XML.

T**ext and Reference Books:**

1. Thomas A Powell, HTML-The Complete Reference,Tata McGraw Hill.
2. Scott Guelich, Shishir Gundavaram, Gunther Birzniek; CGI Programming with Perl 2/e. O'Reilly.
3. Doug Tidwell, James Snell, Pavel Kulchenko; Programming Web Services with SOAP, O'Reilly.
4. Pardi, XML in Action, Web Technology, PHI.
5. Yong, XML Step by Step, PHI.
6. Aaron Weiss, Rebecca Taply, Kim Daniels, Stuven Mulder, Jeff Kaneshki, Web Authoring Desk Reference, Techmedia Publications.
7. HTML The complete Reference, TMH

MCA-13 Java Programming

General Course Information

Course Code: MCA-22 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: The course assumes knowledge of basic knowledge of Object Oriented Technology like C++.

About the Course:

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. The aim of this course is to provide the students basic knowledge about object-oriented development and in-depth knowledge about syntax and programming techniques in Java. The course is very comprehensive and covers all the important Java concepts, e.g., Java basics, Object-Oriented Programming, Multithreading, File handling, Exception handling and more.

Course Outcomes: By the end of the course students will be able to:

CO1. **List** object oriented characteristics peculiar to JAVA programming, objects, members of a class and the relationships among them which are needed for a specific problem.

CO2. **Describe** the concept of OOPs as well as the purpose and usage principles of encapsulation and method overloading.

CO3. **Apply** object-oriented principles for solving problems using JAVA.

CO4. **Identify** various real life problems and solve these problems using Console as-well-as Graphical User Interface.

CO5. **Create** various Java application programs and identifying and handling the various exceptions.

Course Contents

Unit –I

Overview, Control and Looping Structure: Objective Oriented Technology, Introduction to Java Programming, Difference between C++ and Java, Abstraction, Encapsulation, Polymorphism, Inheritance, Data Types and Operators, Java Run-Time Environment, Running Java Application, Java Programming Editors, Control Statements- if, if-else, if-else-if ladder, switch-case statement, Looping Statements – for, while, do while.

Unit - II

Inheritance, Polymorphism and Multithreading: Visibility controls, class and methods in Java, constructors, Final keyword, Array- Single and Multidimensional, String, Vector, Inheritance and its types, Abstract Class, Interfaces and their implementation, Interface Inheritance, Polymorphism

– Overloading and Overriding, Multiple inheritance in Java, Packages, Creating user-defined packages, Multithreading concept in Java.

Unit –III

Exceptions and File Handling: Exceptions in Java, try, catch and finally block, Handling user-defined errors, Study of various Exception Classes, Input and Output Streams: Streams Concept, Byte Stream Classes – FileInputStream, FileOutputStream, Character Streams Classes – FileReader, FileWriter, String TokenizerClass, Handling Primitive Data Types.

Unit– IV

GUI Programming:AWT and SWING Components – Creating a Frame, using Labels, TextFiled, Buttons, ComboBox, CheckBox, Radio Button, JOptionPane, Events and Its Types in Java, Mouse Events, Key Events, Other Events with Frame and Controls, Listeners, Creating Menus and Submenus.

Text and Reference Books:

1. Programming with Java A Primer, E. Balagurusamy, 5th Edition, McGraw Hill Education Pvt. Limited.
2. The Complete Reference Java, Herbert Schildt, 7th Edition, Tata McGraw Hill Education India.
3. A Programmer's Guide to Java Certification, Mughal K. A., Rasmussen R. W., Addison – Wesley.
4. Java – How to Program, Paul Deitel and Harvey Deital, 11th Edition, Pearson Education.

MCA-14 Software Engineering

General Course Information

Course Code: MCA-14 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite: Basic Programming Skills and Innovative assessment.

About the Course:

This course of Software Engineering and Testing will help the learners to understand the systematic approach of all software development phases i.e. from initial stage to final stage of software systems. Learners will gain knowledge about the various processes which are used in software industry for the development of software product and about the testing methods, tools for creating good test cases to improve the quality of software.

In this course, the learners will be able to develop expertise related to:

1. Knowledge about software development life cycle phases and paradigms used in software industry as per the requirements.
2. Acquaintance about software requirements for manual and automated real world systems and to provide an idea about the problem analysis, modelling and design methodologies as per requirements.
3. Knowledge about testing process and computer aided software engineering tools.
4. Knowledge about the testing techniques and quality assurance models.

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** the concepts related to software engineering and to comprehend about the stages of Software Development Life Cycle.

CO2. **Demonstrate** the selection of Software Process Models as per the requirements and to assess the various processes of requirement analysis for software engineering problems.

CO3. **Apply** the software requirement analysis and design process to model the system as per the requirements and to comprehend the principles, processes of Software Project Management including the Software Configuration Management and Risk Management.

CO4. **Plan** the test cases and **apply** the testing techniques for software engineering problems.

CO5. **Predict** software quality based on quality parameters and quality models.

Course Contents

Unit – I

Introduction to Software Engineering: Evolution of Software Engineering, Software Crisis, Types of Software Products, Software Development Life Cycle Phases. Validation and Verification, Software Characteristics, Software Engineering Paradigms: Classical Waterfall Model, Iterative Waterfall Model, Prototyping Model, Evolutionary Model, Spiral Model, Selection of Life Cycle Model.

Software Project Management: Software Project Management Plan (SPMP), Metrics for Project Size estimation- Lines of Code, Function Point Metric, Software Cost estimation - COCOMO, Project Scheduling, Personnel Planning , Organization and Team Structures, Software Configuration Management (SCM), Software Risks, Software Risk Management.

Unit – II

Software Requirements: Functional and non-functional Requirements, User and interface requirements, Software Requirement Specification (SRS), Requirement Engineering Process.
Problem Analysis: Structured Analysis, Data Flow Diagrams (DFD), Decision Tables, Decision Trees, Data Dictionary, Structured Charts, Object Oriented Analysis, System Models: Context Models, Data Modelling, Behavioural Modelling, Object Models, Structured Models.
Software Design: Software Design Fundamentals, Design Principles, Function-Oriented Software Design, Object Oriented Design.
Characteristics of good user interface, Coding Standards and guidelines, Code Review.

Unit –III

Software Testing: Software Testing Basics, Necessity and Objectives of Testing, Difference between Inspection and Testing, Testing vs. Debugging, Testing Life Cycle, Test Artifacts, Test Plan, Test Case Design, Software Testing Strategies, The V-Model of Software Testing, Levels of Software Testing- Unit Testing, Integration Testing-Top down Integration Testing and Bottom-up Integration Testing, Regression Testing, Smoke Testing, System Testing- Recovery Testing, Security Testing, Stress Testing, Performance Testing, Acceptance Testing- Alpha Testing, Beta Testing, Gamma Testing, Software Test Report (STR), Software Testing Tools, Static and Dynamic Testing tools.
Computer Aided Software Engineering (CASE): CASE Environment, advantages of CASE, CASE support in Software Life Cycle, Characteristics of CASE tools.

Unit– IV

Software Testing Methods: Black Box Testing Methods: Equivalence class partitioning, Boundary-value analysis, Error guessing, graph- based testing methods, White Box Testing Methods: Statement coverage, Condition coverage, Path testing, Data flow testing. Object Oriented Testing, Web Testing, GUI testing.
Software Quality: Software Quality Concepts, ISO 9126 Quality Factors, McCall's Quality Factors, Software Quality Assurance (SQA) Activities, Software Reviews-Walkthroughs, Formal Technical Review (FTR), Defect Amplification Model, ISO 9000 series Quality Standards, Capability Maturity Model (CMM), Software Reliability.

Text and Reference Books:

1. Rajib Mall, Fundamentals of Software Engineering, PHI Learning Pvt. Ltd., Third Edition, 2009
2. K. K. Aggarwal & Yogesh Singh, Software Engineering Programs Documentation Operating Procedures, A New Age International Publishers.
3. Pankaj Jalote, An Integrated Approach to Software Engineering, Narosa Publications, Third Edition, 2007.
4. Roger S. Pressman, Software Engineering A Practitioner's Approach, McGraw Hill International Edition
5. M. G. Limaye, Software Testing: Principles, Techniques and Tools, TMH, 2009.
6. Renu Rajani & Pradeep Oak, Software Testing Effective Methods Tools and Techniques, McGraw Hill Education Pvt. Limited, Second Edition, 2018.
7. Nina S. Godbole, Software Quality Assurance Principles and Practice, Narosa Publications, 2011.
8. Yogesh Singh, Software Testing, Cambridge University Press, 2016.

MCA-15 Computer Networks and Internet Protocols

General Course Information

Course Code: MCA-15 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite: Basic knowledge of Digital and Analog Communication.

About the Course:
This course has been designed with an aim to provide students with an overview of the concepts and fundamentals of data communication and computer networks. The learner is given an opportunity to grasp various algorithms for routing of data, forwarding data and switching the data from hop to hop. Layered Architecture adds value to the subject contents.

Course Outcomes: By the end of the course students will be able to:

CO1.**Recall** various models, topologies and devices of Computer Networks.
CO2. **Explain** the functions of various layers in Network Reference Model.
CO3**.Apply** different network concepts in various network communication protocols.
CO4.**Analyse** performance of various protocols in different scenarios.
CO5.**Design** network for an organisation.

Course Content

Unit - I

Data communication: Components. Network: Uses, Topologies, Network Services, OSI and TCP/IP Reference Models; Network categories: LAN, MAN, WAN; Guided Transmission Media, Wireless Transmission Media, Switching Techniques: Circuit Switching, Packet Switching, Message Switching, Networking Devices: Hubs, Repeaters, Bridges, Modems, Switches, Routers, and Gateways.

Unit - II

Data Link Layer-design issues, Framing & Error Handling: Framing Protocols, Error detection and correction mechanisms; Flow Control Protocols: Stop-and-wait, Sliding Window protocols: Go-back-N and Selective Repeat.
Medium Access sub layer: Channel allocation methods, Multiple Access Communication: Random Access-ALOHA, Slotted-ALOHA, CSMA, CSMA-CD, LAN Standards: Ethernet, Fast Ethernet & Gigabit Ethernet.

Unit - III

Network Layer-Design issues, store and forward packet switching connection less and connection oriented networks, Routing algorithms: optimality principle, shortest path, flooding, Distance

Vector Routing, Count to Infinity Problem, Link State Routing, Hierarchical Routing, Congestion control algorithms, admission control. Internetworking: IPV4 and IPV6, IP Addressing (Classful Addressing, Private IP Addresses, Classless Addressing, Sub-netting).

Unit - IV

Transport Layer: Transport layer Services: Addressing, Multiplexing, Flow control, Buffering. Internet Transport Protocols: UDP& TCP. TCP Segmentation & TCP Connection management. Application Layer: Introduction to DNS, HTTP, SMTP, Electronic Mail, WWW

Text and Reference Books:

1. Andrew S Tanenbaum, Computer Networks, 5th Edition, Pearson publications, 2010.
2. Forouzan, Data Communication and networking ,5th Edition, Tata McGrawHill, 2012.
3. William Stalling, Data & Computer Communication 6th edition, LPE Pearson Education, 2013.
4. Todd Lammle, CCNA Study Guide, 6th Edition, 2013.
5. RFCs and Internet Drafts available from Internet Engineering Task Force.

MCA-16 Database Management System Lab.

General Course Information

Course Code: MCA-16 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Exposure to a programming language, MS Access.

About the Course:

This lab. course on DBMS involves a rigorous training on Oracle programming. It provides a strong formal foundation in database concepts, technology and practice to the students to groom them into well-informed database application developers. The objective of the lab course is to develop proficiency in the execution of commands of the database design and query using Oracle.

Course Outcomes: By the end of the course students will be able to:

CO1. **Implement** database problems using DML/DDL commands.
CO2. **Enforce** integrity constraints on a database using a state-of-the-art RDBMS.
CO3. **Analyse** the design of a relational database.
CO4. **Design** a relational database for a given schema.
CO5. **Create** lab assignment record that includes problem definitions, solutions, results and conclusions.
CO6. **Demonstrate** ethical practices, self-learning and team spirit.

List of experiments/assignments:

1. Use oracle software and login with valid userid and password. Explore its GUI and practice some basic commands of it.
2. Three assignments related to creation of database with tables having different fields and data types.
3. Two assignments on the creation of table with different types of constraints.
4. Two assignments on insert, delete and modify records from the tables.
5. Two assignments on modifying the table using the alter command.
6. Two assignments on exploring select statement using various clauses like where, order by, group by, having and aggregate functions.
7. Two assignments on the use of set operations to query the tables.
8. Two assignments on creating joins and views on the tables.
9. One assignment on generating sub-queries.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-17 Web Designing Lab.

General Course Information

Course Code: MCA-17 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Basic programming skills and knowledge of surfing internet.

About the Course:

This lab. course on web development involves learning web-based programming languages. It incorporates the development of web pages by structuring information provided for the website design. The objective of the lab course is to equip the students to design web pages using modern web development tools.

Course Outcomes: By the end of the course students will be able to:

CO1. I**mplement** object models for website design using modern tools like HTML, XML and JAVA scripting etc.
CO2. **Analyse** the design of websites.
CO3. **Test** the design of websites.
CO4. **Design** websites that consider socio-cultural values.
CO5. **Create** a written report for website designed.
CO6. **Use** ethical practices and socio-cultural values while designing websites.

List of experiments/assignments

1. Create a simple webpage using HTML.
2. Designing of registration form with table and use of hyperlink.
3. Design a page with frames to include Images and Videos.
4. Add a cascading style sheet for designing the web page.
5. Use user defined function to get array of values and sort them in ascending order on webpage
6. Design a dynamic web page with validation of form field using Java Script.
7. Design a catalogue in ASP.
8. Event Handling Validation of registration form.
9. Open a Window from the current window on Mouse Over event.
10. Create a simple application to demonstrate Servlets Request and Response object.
11. Demonstrate Array Objects and Date Object's predefined methods
12. Display calendar for the month and year selected from combo box
13. Create a welcome Cookie (Hit for a page) and display different image and text content each time when the user hit the page.
14. Demonstrate Request and Response object using HTML Form.
15. Database Connection to display all the values in the table.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment

should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-18 Java Programming Lab.

General Course Information

Course Code: MCA-27 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: The course assumes knowledge of Object-Oriented Concepts and programming.

About the Course:

This Java course will provide a strong understanding of basic Java programming elements and data abstraction using problem representation and the object-oriented framework. The objective of the lab course is to inculcate proficiency in students to design and develop market-based software applications.

Course Outcomes: By the end of the course students will be able to:

CO1. **Implement** Java programs using object oriented concepts for problem solving.
CO2. **Detect** syntax and logical errors in java programs.
CO3. **Apply** exception handling for making robust JAVA code.
CO4. **Design** java applications using File I/O and GUI.
CO5. **Create** lab record of the solutions of assignments that includes problem definitions, solutions and conclusions.
CO6. **Demonstrate** ethical practices, self-learning and team spirit.

List of experiments/assignments:

1. Use eclipse or NetBeans platform and acquaint with the various menus, create a test project, add a test class and run it to see how you can use auto suggestions and auto fill functionalities. Try code formatter and code refactoring like renaming variables, methods and classes. Try debug step by step with a small program of about 10 to 15 lines which contains at least one if else condition and a for loop.
2. Two assignments illustrating class, objects, methods, arrays and various data types in java.
3. Two assignments on the use of control, looping statements and user defined functions.
4. One assignment illustrating the implementation of various forms of inheritance.
5. One assignment on method overloading.
6. One assignment on polymorphism and method overriding.
7. One assignment on implementing exception handling.
8. One assignment to illustrate interfaces in java.
9. One assignment to create package in java.
10. One assignment to design of multithreaded programs in java.
11. One new assignment on event handling.
12. Two assignments related to java applets.
13. One assignment to design a GUI application.
14. One assignment to access and update data from a database using JDBC.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-21 Data Structures and Algorithms

General Course Information:

Course Code: MCA-21 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite: ElementaryProgramming skills in C, C++ etc.

About the Course

Data Structure and Algorithms is a core and an essential course for every graduate in Computer Science and Engineering. This course introduces data structures like arrays, linked lists, trees and graphs etc. and various operations to be implemented on these data structures for solving real world problems. It includes various sorting and searching algorithms as well. Further, it incorporates complexity analysis of algorithms implemented on various data structures.

Course Outcomes: By the end of the course students will be able to:

CO1. **List or describe** types of data structures and operations that can be implemented on these data structures.
CO2. **Select** appropriate data structures as applied to specified problem definition.
CO3. **Apply** appropriate data structures with respect to effective storage of data and efficiency of the required operations on data for solving real-world problems.
CO4. **Implement** operations like searching, insertion, and deletion, traversing mechanism etc. on various data structures.
CO5. **Apply** appropriate sorting/searching technique for given problem and analyse the time complexity of algorithms.
CO6. **Compare** and contrast algorithms based on worst, average and worst case complexities.

Course Contents

Unit - I

Data Structures Basics: Structure and Problem Solving, Data Structures and Their Types, Data structure operations, Abstract Data Types.
Linear lists: Arrays and linked lists: memory representations, implementing operations like traversing, searching, inserting and deleting etc., types of arrays and linked lists, Applications of arrays and linked lists.

Unit - II

Stack and Queue: Introduction, sequential and linked implementations, Operations and representative applications, Circular queues, De-queue, Priority Queues, Applications of Queues.
Application of stacks: Infix to postfix Transformation, Evaluating Arithmetic Expressions.
Trees: Binary Trees, terminology, representation and traversals- pre, post & in-order traversals. Binary Search Trees implementation and operations.

Unit - III

Heapsort - Heaps, Maintaining the heap property, Building a heap, heapsort algorithm
Advanced tree data structures such as Height Balanced or AVL trees, Multiway Trees or B Trees, red and black trees, splay trees.
Graphs: Graph definitions and related terminology, memory representations for graphs and associated algorithms for searching, inserting and deleting nodes and related algorithms, Graph traversals and applications (DFS, BFS). Shortest path algorithms: Dijkstra's and Warshall's algorithms.

Unit - IV

Sequential and Binary search, Sorting algorithms: Bubble sort, Selection sort, Insertion sort, Quick sort, Merge sort, Internal and external sorting and stable sorting techniques.
Hash Tables - Direct-address tables, Hash tables, Hash functions, Open addressing, Perfect hashing.
Algorithm: Role of Algorithms in Computing, Analyzing and Designing Algorithms, Time- space tradeoffs, asymptotic notations, Standard notations.
Comparison of searching and sorting techniques based on their complexity analysis,

Text and Reference Books:

1. Aho, A. V., Ullman, J. D., and Hopcroft, J. E., Data Structures and Algorithms, Addison-Wesley, 1983.
2. LangsamYedidyah, Augenstein J Moshe, Tenenbaum M Aaron, Data Structures using C and C++, 3rdedition, PHI, 2009.
3. Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C., Introduction to Algorithms, MIT Press, 2009.
4. Weiss, M. A., Data Structures and Algorithm Analysis in C++, Addison-Wesley, 2007.
5. Sahni, S., "Data Structures, Algorithms, and Applications in C++", WCB/McGraw-Hill, 2001.

MCA-22 Python Programming

General Course Information

Course Code: MCA-13 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisite: Exposure to programming languages

About the Course:

Python is a popular open source programming language used for both standalone programs and scripting applications in a wide variety of domains. It is free, portable, and powerful and is both relatively easy and remarkably fun to use. In today's era Python has found great applicability in machine learning, data analytics and many other data science application. This is introductory course and covers most of the basic concepts required for basic python programming. Some of the contents are advanced may be useful for data analytics purpose.

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** various basic programming constructs including operators, character sets, basic data types and control statements.
CO2. **Explain** Python packages and their functionalities for data analysis.
CO3. **Solve** problems using python programming.
CO4. **Analyse** the results of data analysis or machine learning programs
CO5. **Evaluate** solutions according to the problem definition.
CO6. **Develop** database applications in Python.

Course Content

Unit - I

Introduction to Python, History of Python, Features of Python, Python Identifiers, Python Character Set, Keywords and Indentation, Comments, Command Line Arguments, Assignment Operator, Operators and Expressions, print() Function, input() Function, eval() Function, Python Data Types: int, float, complex, Variables, Mutable vs Immutable variables, Namespaces, Decision Statements: Boolean Type, Boolean Operators, if statement, else statement, Nested Conditionals Statements, Multi-way Decision Statements (elseifstatement).

Unit - II

Loop Control Statements: While loop, range() Function, For Loop, Nested Loops, Infinite Loop, Break Statement, Continue Statement, Pass Statement, Introduction to Strings, String Operations: Indexing and Slicing, Lists: Operations on List: Slicing, In-built Functions for Lists, List Processing: Searching and Sorting, Dictionaries: Need of Dictionary, Operations on Directories: Creation, Addition, Retrieving Values, Deletion; Tuples, operations on Tuples, Inbuilt Functions for

Tuples, Introduction to Sets, operations onsets.

Python Functions, Inbuilt functions, Main function, User Defined functions, Defining and Calling Function, Parameter Passing, Actual and Formal Parameters, Default Parameters, Global and Local Variables, Recursion, Passing Functions as Data, Lamda Function, Modules, Importing Own Module, Packages.

Unit - III

Operations on File: Reading text files, read functions, read(), readline() and readlines(), writing Text Files, write functions, write() and writelines(),Manipulating file pointer using seek, Appending to Files.

Python Object Oriented: Overview of OOP, Classes and objects, Accessing attributes, Built-in Class Attributes, Methods, Class and Instance Variables, Destroying Objects, Polymorphism, Overlapping and Overloading of Operators, Class Inheritance: super(), Method Overriding, Exception Handling, Try-except-else clause, Python Standard Exceptions, User-Defined Exceptions

Unit - IV

Databases in Python: Create Database Connection, create, insert, read, update and delete Operation, DML and DDL Operation with Databases.

Python for Data Analysis: numpy: Creating arrays, Using arrays and Scalars, Indexing Arrays, Array Transposition, Universal Array Function, Array Processing, Array Input and Output

Pandas: Series, Data Frame, Panel, Index objects, Re-indexing, Iteration, Sorting. Matplotlib: Python for Data Visualization, Visualization Section, Sklearn: loading of dataset, learning and predicting, Model Persistence.

Text and Reference Books:

1. Ashok Namdev Kamthane, Programming and Problem Solving with Python, McGraw Hill Education Publication, 2018.
2. John Guttag, Introduction to Computation and Programming using Python, Springer, Revised and Expanded version (Referred by MIT), 2013.
3. Lutz, M., Learning Python: Powerful Object-Oriented Programming. O'Reilly Media, Inc., 2013.
4. Michael T Goodrich and Robertto. Thamassia, Micheal S Goldwasser, Data Structures and Algorithms in Python, Wiley, 2016.
5. Y. Daniel Liang, Introduction to Programming Using Python, Pearson, 2013.
6. Reema Thareja, Python Programming Using Problem Solving Approach, Oxford Publications,2017.
7. Dr. R. Nageswara Rao, Allen B. Downey, Core Python Programming, Think Python, O'Reilly Media, 2012.
8. Kenneth A. Lambert, The Fundamentals of Python: First Programs, Cengage Learning, 2011.

MCA-23: Artificial Intelligence

General Course Information:

Course Code: MCA-23 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Basic knowledge of HTML, XML, ASP, JSP and Web Designing.

About the Course

In this course, the learners will be able to develop expertise related to general purpose problem solving, Representation of knowledge, Reasoning under uncertainty, Planning and Natural Language processing

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** various Artificial Intelligence techniques.
CO2. **Illustrate** reasoning under uncertainty.
CO3. **Apply** search and knowledge representation techniques to solve AI problems.
CO4. **Compare** strengths and weaknesses of AI algorithms
CO5. **Combine** various AI techniques to solve intelligent systems' problems.

Course Contents

Unit – I

Introduction to AI: Introduction, AI problems, AI Techniques, State Space Search, production systems

Problem Solving Using Search: Blind search techniques - Breadth first search, Depth first search. Heuristic search techniques - Generate and test, Hill Climbing, Best first search, A* Algorithm, AO* Algorithm, Constraint Satisfaction, The Minimax Search Procedure, Adding Alpha-Beta Pruning.

Unit – II

Knowledge Representation: Introduction, Knowledge Representation- Representation and Mappings, Symbolic Logic - Propositional logic, Predicate logic- Representing simple facts in logic, Representing Instances and ISA Relationship, Computable functions and Predicates, Unification, Resolution.

Representing Knowledge Using Rules: Procedural versus Declarative Knowledge, Logic Programming, Forward versus Backward Reasoning, Matching, Control Knowledge.

Unit –III

Reasoning Under Uncertainty: Introduction to Nonmonotonic Reasoning, Probability and Baye's Theorem, Certainty Factors and Rule-based Systems, Bayesian Networks.
Fuzzy logic systems: Introduction, Crisp Set, Fuzzy Sets, Fuzzy Logic Control.

Unit– IV

Planning: Introduction, Components of Planning System, Goal Stack Planning, Nonlinear Planning using Constraint Posting, Hierarchical Planning.
Natural Language Processing: Introduction, Syntactic Processing, Semantic Analysis, Discourse and Pragmatic Processing

Text and Reference Books:

1. Elaine Rich, Kevin Knight and Shivashankar B Nair, Artificial intelligence, McGraw Hill Education. 3rd edition, 2009.
2. Stuart Russel and Peter Norvig, Artificial intelligence: A modern Approach, Pearson Education, 3rd edition, 2015.
3. Dan W. Patterson, Introduction to Artificial Intelligence and Expert System, Pearson Education.1st edition, 2007.
4. Deepak Khemani, A first course in Artificial Intelligence, McGraw Hill Education. 3rd edition, 1st edition, 2013.
5. George F. Luger, Artificial Intelligence: Structures and Strategies for Complex Problem Solving, Pearson Education, 5th edition, 2009.

MCA-24 Computer System Architecture

General Course Information

Course Code: MCA-24 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Basic Computer System and Digital Electronics

About the Course:

Computer Architecture and organization describes the role of instruction set architecture in digital computer, main memory, and input/output devices. It illustrates the simple data path and control design for processors. It helps to understand the different operations and concept of instructions. It would enable the students to learn the basic function and architecture of modern computer systems.

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** the basic hardware and software issues of computer organization and the representation of data at machine.
CO2. **Discuss** the basic components and their interfacing.
CO3. **Apply** instructions for performing different operations.
CO4. **Analyse** the effect of addressing modes on the execution time of a program.
CO5. **Contrast** different types of memory, their architecture and access methods.
CO6. **Design** of simple computer with different instruction sets.

Course Content

Unit - I

Basic Principles: Boolean algebra and Logic gates, Combinational logic blocks (Adders, Subtractors, Multiplexers, Encoders, decoders, de-multiplexers, K-Maps), Sequential logic blocks (Flip-Flops, Registers, Counters); Flynn's classification of computers (SISD, MISD, MIMD); Performance metrics: MIPS, MFLOPS; CPU Architecture types: computer register, (accumulator, register, stack, memory/ register) detailed data path of a typical register based CPU.

Unit - II

Computer Organization: Store program control concept, Instruction codes, timing and control, instruction cycle; type of instructions: memory reference, register reference, I/O reference; Basics of Logic Design, accumulator logic, Control memory; Micro Programmed Control: address sequencing, micro-instruction formats, micro-program sequencer, Implementation of control unit.

Unit - III

Instruction Set Architecture & Parallelism: Instruction set based classification of processors (RISC,

CISC, and their comparison); Stack Organization, Instruction Formats; addressing modes: register, immediate, direct, indirect, indexed; Operations in the instruction set: Arithmetic and Logical, Data Transfer, Control Flow; Types of interrupts; Introduction to Parallelism: Goals of parallelism (Exploitation of concurrency, throughput enhancement); Amdahl's law; Instruction level parallelism (pipelining, super scaling –basic features); Processor level parallelism (Multiprocessor systems overview).

Unit - IV

Memory Hierarchy & I/O Techniques: The need for a memory hierarchy (Locality of reference principle, Memory hierarchy in practice: Cache, main memory and secondary memory, Memory parameters: access/cycle time, cost per bit); Main memory (Semiconductor RAM & ROM organization, memory expansion, Static & dynamic memory types); Cache memory (Associative & direct mapped cache organizations; input-output interface, mode of transfer, DMA (Direct memory transfer).

Text and Reference Books:

1. Mano, M. Morris, Digital Logic and Computer Design, Prentice Hall of India Pvt. Ltd., 1981.
2. M. Morris Mano, Computer System Architecture, Prentice Hall of India Pvt. Ltd., 1993.
3. Milles J. Murdocca, Vincent P. Heuring, Computer Architecture and Organization, An Integrated Approach, John Wiley& Sons Inc., 2007.
4. William Stallings, 10th edition, Computer Organization and Architecture, Prentice Hall, 2016.
5. Heuring, V. P., Jordan, H.F., Computer Systems Design and Architecture, Addison Wesley, 1997.
6. R.P Jain, Modern Digital Electronics, 3rd Edition, Tata McGraw Hill, 2003.

MCA-25 Discrete Mathematics and Optimization

General Course Information:

Course Code: MCA-25 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Basic knowledge of Pre-calculus, Algebra and Trigonometry.

About the Course

The purpose of this course is to understand and use discrete structures that are backbones of computer science. Introduction to Discrete Mathematics is a course designed for students interested in information technology and programming that includes topics in set theory, algebraic structures, and graph theory. On the completion of this course, the students will be able to explain and apply the basic methods of discrete mathematics in Computer Science.

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** various discrete structures and the related operations.
CO2. **Illustrate** different discrete structures with the help of examples.
CO3. **Apply** appropriate techniques to solve problems related to discrete structures.
CO4. **Develop** proficiency in business study and decide the feasibility of system.
CO5. **Justify** the solutions with the help of proofs.
CO6. **Combine** techniques related to discrete structures and optimization for solving real world problems and profitable solution for industries.

Course Contents

Unit - 1

Set Theory: Introduction to Set Theory, Venn Diagrams, Set Operations, Algebra of Sets, Duality, Finite, Infinite Sets and Counting Principle, Classes of Sets, Power Sets, Partitions, Multi Sets, Relations: Cartesian Product, Representation of Relations, Types of Relation, Equivalence Relations, Functions: Definition, Types of Functions, Composition of Functions, Inverse Function, Posets.

Unit - II

Logic and Propositional Calculus: Introduction, Propositions and Compound Propositions, Basic Logical Operations, Propositions and Truth Tables, Tautologies and Contradictions, Logical Equivalence, Algebra of Propositions, Conditional and Bi-conditional Statements, Algebraic Structures: Group Axioms, Monoid, Semi-Groups, Subgroups, Abelian Group, Cosets, Normal Subgroup, Cyclic Group, Lagrange's Theorem.

Unit - III

Graphs Theory: Introduction to Graphs, Multi Graph, Directed and Undirected Graphs, Subgraphs, Bipartite Graphs, Regular Graphs, Connected Graphs, Homomorphic and Isomorphic Graphs, Cut points and Bridges, Paths and Circuits, Euler Graph, Hamiltonian Graph, Planar Graph, Euler Formula, Weighted Graphs, Dijkstra's Shortest Path Algorithm for Weighted Graphs, Trees, Spanning Trees, Minimum Spanning Tree (Prim's and Kruskal's Algorithm).

Unit – IV

Introduction to Optimization Techniques, Origin & Development of O.R., Nature & Characteristic features of O.R., Models & Modeling in Operation Research. Methodology of O.R.Linear Programming : Formulation, Graphical solution, standard and matrix forms of linear programming problems, Simplex method and its flow chart, Two phase Simplex method, Degeneracy.

Text and Reference Books:

1. S. Lipschutz and M. Lipson, Discrete Mathematics, Tata McGraw Hill, 3rd Edition, 2010.
2. C. L. Liu, Elements of Discrete Mathematics, Tata McGraw Hill, 3rd Edition, 2008.
3. Kenneth H. Rosen, Discrete Mathematics and its applications, 6th Edition, Tata McGraw Hill, 2011.
4. B. Kolman, R. C. Busby and S. C. Ross, Discrete Mathematical structures, 6th Edition, PHI, 2010.
5. J.P. Trembley and R. Manohar, Discrete Mathematical Structures with Applications to Computer Science, Tata McGraw Hill – 13th reprint, 2012.
6. Sharma, S.D., Operations Research, KedarNath and Ram Nath, Meerut.
7. Taha, H.A., Operation Research - An Introduction, McMillan Publishing Co, New York.
8. Gupta P.K., Hira and D.S., Operation Research, Sultan Chand & Sons, New Delhi.

MCA-26 Data Structures and Algorithms Lab.

General Course Information

Course Code: MCA-26 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Programming in C/C++ language.

About the Course

This lab. course involves implementation of basic and advance data structures and various operations on these data structures. The objective of the lab course is to train the students to solve the problems related to data structures and choose the appropriate data structure for solving computational problem efficiently.

Course Outcomes: By the end of the lab course a student would be able to:

CO1. **Implement** various data structures and the related operations.
CO2. **Analyse** space and time complexity of algorithms.
CO3. **Compare** solutions on the basis of the appropriateness of data structure used and the efficiency of the operations implemented.
CO4. **Integrate** knowledge of data structures to solve real world problems related to data structure and algorithms.
CO5. **Create** written records for the given assignments with problem definition, design of solution and conclusions.
CO6. **Demonstrate** ethical practices while solving problems individually or in groups

List of experiments/assignments

1. Two assignments related to creating and manipulating matrices and linear lists.
2. Two assignments associated with linked list, operations on linked lists and their applications.
3. Two assignments on array and linked implementation of stacks and queues.
4. Two assignments on trees and their applications.
5. Two assignments on graphs and their applications.
6. Two assignments on different searching and sorting methods with their complexity analysis.
7. One assignment on challenging problems on data structures to be given in groups.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-27 Python Programming Lab.

General Course Information

Course Code: MCA-17 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Basic programming skills

About the Course:

Python is a scripting programming language known for both its simplicity and wide breadth of applications. For this reason it is considered one of the best languages for beginners. Used for everything from web development to scientific computing Python is referred to as a general purpose language by the greater programming community. The major objective of Python language is to make the students solve real word problem efficiently using python library.

Course Outcomes: By the end of the course students will be able to:

CO1. **Implement** solutions to the given assignments in Python.
CO2. **Use** various Python packages for solving different programming problems.
CO3. **Devise** solutions for complex problems of data analysis and machine learning.
CO4. **Evaluate** the output of data analysis and machine learning models.
CO5. **Create** lab records of the solutions for the given assignments.
CO6. **Demonstrate** use of ethical practices, self-learning and team spirit.

List of experiments/assignments

1. Install Python and explore various popular IDE like IDLE, PyCharm, andAnaconda.
2. Assignments to perform various number operationslike
    a. Find maximum from a list ofnumbers
    b. GCD of two number
    c. Square root of anumber
    d. Check number is prime ornot.
    e. Print first N primenumbers
    f. Remove duplicate numbers fromlist
    g. Print the Fibonacciseries.
3. Assignments to perform various operations on Strings like creation, deletion,concatenation.
4. Create a List L = [10, 20, 30]. Write programs to perform followingoperations:
    a. Insert new numbers to listL.
    b. Delete numbers from listL.
    c. Sum all numbers in list L.
    d. Sum all prime numbers in listL.
    e. Delete the listL.
5. Create a Dictionary D= {‘Name’: ‘Allen’, ‘Age’: 27, 5:123456}. Write programs to perform following operations:
    a. Insert new entry inD.

b. Delete an entry fromD.
c. Check whether a key present in D.
d. Update the value of akey.
e. Clear dictionaryD.

6. Two assignments on Sets to perform various operation like union, intersection, differenceetc.
7. Two assignments related to searching operation like linear search, binarysearch.
8. Three assignments related to sorting like selection sort, bubble sort, insertionsort.
9. Demonstrate the use of dictionary for measuring student marks in five subjects and you have to find the student having maximum and minimum average marks.
10. Two assignment on usage of different available packages like random package toperform
    a. Print N random numbers ranging from 100 to500.
    b. Print 10 random strings whose length between 3 and5.
11. Two assignments on usage of package such as Numpy, Pandas.
12. Implement and demonstrate the functions of a simplecalculator.
13. One assignment on implementing object oriented concept such as classes, inheritance, and polymorphism.
14. One assignment on file handling that how data is read and written to afile.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-28: Artificial Intelligence Lab

General Course Information:

Course Code: MCA-28 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Prerequisite

Basic knowledge of HTML, XML, ASP, JSP and Web Designing.

About the Course

In this course, the learners will be able to develop expertise related to general purpose problem solving, Representation of knowledge, Reasoning under uncertainty, Planning and Natural Language processing

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** various Artificial Intelligence techniques.
CO2. **Illustrate** reasoning under uncertainty.
CO3. **Apply** search and knowledge representation techniques to solve AI problems.
CO4. **Compare** strengths and weaknesses of AI algorithms.
CO5. **Combine** various AI techniques to solve intelligent systems' problems.

List of Experiments:

1. Write a program to implement BFS/DFS Traversal?
2. Write simple facts for the statements and querying it.
3. Write a program for Family-tree.
4. Write Program for Monkey-banana Problem.
5. Write a program to implement Tic-Tac-Toe game.
6. Write programs for computation of recursive functions like factorial Fibonacci numbers, etc.
7. Write program to solve 5-queens problem.
8. Write a Program for water jug problem.
9. Write a program for travelling salesman problem.
10. Write a program to implement all set operations.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-31: Machine Learning

General Course Information:

Course Code: MCA-31 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Basics of Linear Algebra and Statistics, Basics of Probability Theory, Basic programming constructs of an object oriented programming language like C++, Java or Python. Some fundamental data structures like array, string, list, tree, graph etc.

About the Course

This course introduces several fundamental concepts and methods for machine learning. This course includes various machine learning techniques including linear regression, clustering, classification, decision trees, Bayesian learning, artificial neural networks, support vector machine, data reduction techniques and latest topic like deep learning methods. The objective is to familiarize the students with the prevalent machine learning algorithms, techniques and their applications. The main objective of the course to equip the students with recent fundamental techniques of machine learning that will help him or her to become a proficient data scientist, data analyst or a skilled knowledge worker.

Course Outcomes: By the end of the course students will be able to:

CO1. **Describe** fundamental terminologies and different types of machine learning
CO2. **Interpret** the results and pattern obtained for different machine learning algorithms
CO3. **Apply** machine learning techniques to solve real world problems
CO4. **Analyse** the performance of different machine learning algorithms
CO5. **Compare** various machine learning techniques on different parameters
CO6. **Design** machine learning algorithms for data classification, pattern recognitions, optimization and searching problems

Course Contents

Unit – 1

Introduction: What is machine learning? Types of machine learning, Examples of machine learning applications: learning associations, classifications, regression, unsupervised learning, reinforcement learning.

Unsupervised learning : k-mean clustering, self organizing feature map (SOM algorithm)

Dimensional Reduction: Principal Component Analysis.

Unit – II

Decision tree: Introduction, decision tree representation, appropriate problem for decision tree learning algorithm, basic decision tree learning algorithm, entropy measures ,information gain measures, Example problem for illustrating ID3.
Regression: Linear regression, linear regression examples.

Unit –III

Artificial neural network: Introduction, biological motivation, neural network representation, appropriate problem for neural network learning, perceptron, representation power of perceptron, perceptron training rule, gradient descent and delta rule, multilayer network and backpropagation algorithm, a differentiable threshold unit, the backpropagation algorithm, convergence and local minima, deep learning.

Unit– IV

Bayesian learning: Introduction, Bayes theorem, Naive Bayes classifiers.
Instance based learning-nearest neighbour learning, remarks on k-nearest neighbour algorithm.
Support Vector Machines: optimal separation, kernels, extensions to the support vector machine

Text and Reference Books:

1. Tom M. Mitchell, Machine Learning, McGraw-Hill, 1997
2. Stephen Marsland, Machine Learning, Chapman and Hall /CRC, 2009
3. Ethem Alpaydin, Introduction to Machine Learning,PHI,2004
4. Bishop Christopher, Pattern Recognition and Machine Learning, Springer Verlag, 2006.
5. Trevor Hastie, Robert Tibshirani, Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference and Prediction, Springer, 2nd edition, 2009
6. J. Han and M. Kamber, Data Mining Concepts and Techniques, 3rd Edition, Elsevier, 2012.

MCA-32 Advanced Operating Systems

General Course Information

Course Code: MCA-32 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Knowledge of computer fundamentals, Data Structures and programming in C. C is the primary implementation language for systems that we will analyze, requiring reading fluency; user space C programs will also be written and may be extended as part of lab exercises. Be comfortable with UNIX command-line environment.

About the Course

The objective of this course is to help students become familiar with the fundamental and advanced concepts of operating system and provide them with enough understanding of operating system design.

Course Outcomes: By the end of the course students will be able to:

CO1. **List** various functions, types and architectural characteristics of operating systems.
CO2. **Explain** fundamental concepts of operating systems.
CO3. **Apply** operating system design concepts for solving problems regarding CPU scheduling, management of processes, file management, Distributed OS and deadlocks etc.
CO4. **Analyze** the issues related to various operating systems.
CO5. **Design** solutions for the memory and process management problems based on different algorithms.

Course Contents

Unit –1

Introductory Concepts: Operating systems functions and characteristics, Computer system organization, Computer system architecture, Operating system structure, Virtual machines, Protection & security, Operating system services and system calls, Types of Operating systems: Batch operating system, Time-sharing OS, Distributed operating system, Real time systems.NOS, Multiprocessor OS, Mobile OS, RTOS, Cloud OS

Unit - II

Processes: Process in memory, Process states, PCB, Process scheduling, Inter-process communication, CPU scheduling: Levels of Scheduling, Scheduling criteria, Scheduling algorithms, Multithreading models. Thrashing.
File Systems: Types of Files and their access methods, File allocation methods, Directory structure

Unit –III

Distributed OS- types of distributed operating systems, Network topology, Communication protocols. Issues in Distributed operating systems

Deadlocks- Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Recovery from deadlock.

Unit– IV

Memory: Basic hardware, Address binding, swapping, logical and physical address space, Contiguous memory allocation, Fragmentation, Paging, TLB, Segmentation, Virtual memory- Demand paging, Page replacement algorithms.
Case Studies: Comparative study of WINDOW, UNIX & LINUX system.

Text and Reference Books:

1. Advanced Concepts in Operating Systems, by Mukesh Singhal, Niranjan G. Shivaratri, TMH.
2. Operating System Concepts, (6th Edition), by Abraham Silberschatz, Peter Baer Galvin, Greg Gagne.
3. Theory and problem of programming with C, Byron C Gottfried, TMH
4. Teach yourself all about computers by Barry Press and Marcia Press, 2000, IDG Books India.
5. Using Computers and Information by Jack B. Rochester, 1996, Que Education & Training.

MCA-33: Data Analytics

General Course Information:

Course Code: MCA-33 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Basic programming skills, basic statistics.

About the Course

In this course, the learners will be able to develop expertise related to

1. Learn to load, clean/process, and transform the data
2. Learn to analyse and interpretation of the data using recent approaches
3. User of languages, algorithms as well as mathematical and statistical models.
4. Problem formulation and selection of appropriate models for data to solve the hidden solutions to business-related challenges.
5. Working with different types of method for data representation.
6. Incorporating data mining software to solve real-world problems.
7. Apply algorithms to build the machine more intelligence.

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** the basic terms related to data science
CO2. **Describe** data with statistical summaries and plots
CO3. **Build** predictive models.
CO4. **Analyse** the quality of a model fit
CO5. **Interpret** and evaluate predictive models
CO6. **Conclude** the findings of predictive modelling.

Course Contents

Unit – 1

Data science preliminaries: scales of measurements and their implementation. Working with vectors, matrices and tabular data (data frames), reading and writing tabular data from and to files. Packages for reading and writing data from and to EXCEL files. Describing data with statistical summaries (mean, median, mode, variance and standard deviation). Discriminating between sample and population, Quantile-Quantile plot. writing user-defined functions in R/Python.

Manipulating tabular data: Sorting, filtering cases, selecting variables, deriving new variables, grouping and summarizing data. working with packages (dplyr, tidyverse or any equivalent package in Python) for data manipulations and transformations, discovering correlation between attributes.

Unit – II

Exploratory data analysis: random and normally distributed variables, skewed normal distribution, z-score, detecting outliers in data, handling missing values.
Visualizing data through various plots and charts: pie chart, bar charts, histogram, frequency polygon, density plots, scatter plots, box & whisker plots, heat maps and contour plots., plotting the above graphs in R/Python, plotting with package- ggplot2 in R or any equivalent package in Python.

Unit –III

Predictive modeling: what is predictive modeling, estimating a function, the trade-off between model accuracy and prediction accuracy and model interpretability, regression versus classification, measuring the quality of fit, The bias and variance trade- off.
Simple and multiple linear regression modeling: estimating the coefficients, assessing the accuracy of the coefficient estimates, assessing the accuracy of the model. Logistic regression modeling, building regression models in R/Python.

Unit– IV

Classification Modeling: The process of classification, decision tree, bayesian, k-nearest neighbor, support vector machine classification models and their implementation in R/Python. evaluating a classification model: confusion matrix, accuracy, sensitivity, specificity, f-measure, kappa statistics, ROC and area under curve. accuracy and interpretability of classification models.
Evaluating the accuracy of a classifier: holdout or random sampling methods, cross-validation, bootstrap methods.

Text and Reference Books:

1. Han, J., Kamber, M, Pei, J., Data Mining Concepts and Techniques, Third edition, Morgan Kaufmann, 2012.
2. W. N. Venables, D. M. Smith and the R core Team, An introduction to R, Notes on R: A Programming Environment for Data Analysis and Graphics, version 3.3.2, 2016.
3. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with Applications in R, Springer, 2013.
4. Hadley Wickham and Garrett Grolemund, R for Data Science Import, Tidy, Transform and model Data, O'Reilly, 2017.
5. Roger D. Peng, R Programming for Data Science, Lean Publishing, 2015.
6. Beazley, D., & Jones, B. K. Python Cookbook: Recipes for Mastering Python 3. O'Reilly Media, 2013.
7. Muller, A. C., & Guido, S. Introduction to Machine Learning with Python: A Guide for Data Scientists. O'Reilly 2016
8. Paul Teeter, R Cookbook, O'Reilly, 2011.

MCA-34 Cyber Security

General Course Information:

Course Code: MCA-34 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisite

Computer networks, essentials of computer security, programming languages like python, java etc.

About the Course

The increase in techniques for unauthorized access into systems has led to variety of cyber-attacks. To mitigate the exploitation of the vulnerabilities leading to these attacks, we need to adopt robust security architecture into our premises. We have to choose between various cyber security technologies. In the current scenario, we require to secure end-to-end devices, networks, networking devices. The objective of this course is to enable students to get acquainted to cyber security principles to be followed while working online and offline.

Course Outcomes: By the end of the course students will be able to:

CO1. **Recognize** the terminology associated with cyber security.
CO2. R**epresent** cyber security in terms of ethics, principles, Intellectual property and Trademarks.
CO3. **Analyze** cyber activities on the internet to follow IT Act.
CO4. **Evaluate** cybercrime situations and recommend appropriate cyber security laws CO5 integrate frameworks to sustain critical infrastructures.

Course Contents

Unit - 1

Cyber Security Fundamentals: Network and Security Concepts: Firewalls, Virtualization, DNS, Radio- Frequency Identification, Attacker Techniques and Motivations: Tunneling Techniques, Fraud Techniques, Threat Infrastructure, Exploitation, Malicious code. Defense and Analysis Techniques.

Unit - II

Ethics in Cyber Security: Privacy, Intellectual property in the cyberspace, Professional ethics, Freedom of speech, Fair user and ethical hacking, Trademarks, Internet fraud, Electronic evidence, Forensic technologies, Digital evidence collections. Tools and methods used in cybercrime: Introduction, Password cracking, Keyloggers and spywares, Virus and worms, Phishing and identity theft, Trojan horses and backdoors, Steganography

Unit – III

Cyber crimes and Cyber security: Cyber crime and legal landscape around the world, Cyber laws, The Indian IT Act, Challenges, Digital signatures and Indian IT Act, Amendments to the Indian IT Act, Cyber crime and punishment, Cost of Cyber crimes and IPR Issues, Web threats for organizations, Social computing and associated challenges for organizations.

Unit – IV

Protecting Critical Infrastructures: Critical Infrastructures: Key Assets, Critical Infrastructure Interdependencies, Internet, Social Media and Cyber Attacks on Critical Infrastructures, Cyber Threat Spectrum- Cyberspace Attacks and Weapons, Framework for improving Critical Infrastructure Cyber security.

Text and Reference Books:

1. James Graham, Richard Howard, "Cyber Security Essentials", CRC Press, Taylor & Francis Group, ISBN: 978-1-4398-5126-5, 2011.
2. Thomas A. Johnson, "Cyber-Security Protecting Critical Infrastructures from Cyber Attack and Cyber Warfare", CRC Press, ISBN:978-1-4822-3923-2, 2015.
3. Nina Godhole and Sunit Belapure, Cyber Security, Wiley India, 2011.

MCA – 35 Elective – I Computer Graphics

General Course Information

Course Code: MCA-35 Elective I Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Programming skills in C/C++ and Data Structures.

About the Course:

This course involves studying graphic techniques, algorithms and imaging models. Moreover, students learn about the techniques for clipping, cropping, representing 2-D and 3-D objects.

Course Outcomes: By the end of the course students will be able to:

CO1. **State** basic concepts related to graphics.
CO2. **Describe** the principles of creating graphical objects and graphical user interface applications.
CO3. **Apply** 2-D and 3-D transformations (rotation, scaling, translation, shearing) on geometric objects.
CO4. **Use** different techniques for clipping and filling geometric objects.
CO5. **Compare** different graphics algorithms for different geometric objects.
CO6. **Create** user-friendly interfaces for computer applications.

Course Content

Unit - I

Introduction to Computer Graphics: What is Computer Graphics, Computer Graphics Applications, Computer Graphics Hardware and software, Two dimensional Graphics Primitives: Points and Lines, Line drawing algorithms: DDA, Bresenham's; Circle drawing algorithms: Using polar coordinates, Bresenham's circle drawing, mid-point circle drawing algorithm; Filled area algorithms: Scan-line: Polygon filling algorithm, boundary filled algorithm.

Unit - II

Two/Three Dimensional Viewing: The2-Dviewing pipeline, windows, viewports, window to view port mapping; Clipping: point, clipping line (algorithms):- 4 bit code algorithm, Sutherland-cohen algorithm, parametric line clipping algorithm (Cyrus Beck). Polygon clipping algorithm: Sutherland-Hodgeman polygon clipping algorithm.

Two dimensional transformations: transformations, translation, scaling, rotation, reflection, composite transformation.

Three dimensional transformations: Three-dimensional graphics concept, Matrix representation of 3-D Transformations, Composition of 3-D transformation.

Unit - III

Viewing in 3D: Projections, types of projections, the mathematics of planner geometric projections, coordinate systems.
Hidden surface removal: Introduction to hidden surface removal, Z- buffer algorithm, scanline algorithm, area sub-division algorithm.

Unit - IV

Representing Curves and Surfaces: Parametric representation of curves: Bezier curves, B-Spline curves. Parametric representation of surfaces; Interpolation method.
Illumination, shading, image manipulation: Illumination models, shading models for polygons, shadows, transparency. What is an image? Filtering, image processing, geometric transformation of images.

Text and reference books:

1. James D. Foley, Andeies van Dam, Stevan K. Feiner and Johb F. Hughes, Computer Graphics Principles and Practices, second edition, Addision Wesley, 2000.
2. Pradeep K Bhatia, Computer Graphics, 3rd edition, I K International Pub, New Delhi, 2013.
3. Donald Hearn and M. Pauline Baker, Computer Graphics 2nd Edition, PHI, 1999.
4. David F. Rogers, Procedural Elements for Computer Graphics Second Edition, T.M.H, 2001.
5. Alan Watt, Fundamentals of 3Dimensional Computer Graphics, Addision Wesley, 1999.
6. Corrign John, Computer Graphics: Secrets and Solutions, BPB, 1994.
7. Pilania & Mahendra, Graphics, GUI, Games & Multimedia Projects in C, Standard Pub., 2002.
8. N. Krishanmurthy, Introduction to Computer Graphics, T.M.H, 2002.

MCA – 35 Elective –I Theory of Computations

General Course Information

Course Code: MCA-35 Elective I Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites:

The students are expected to have a strong background in the fundamentals of discrete mathematics like in the areas of symbolic logic, set, induction, number theory, summation, series, combinatorics, graph, recursion, basic proof techniques.

About the Course:

Formal Languages and Automata theory presents the theoretical aspects of computer science, which lay the foundation for students of Computer Science. The course introduces some fundamental concepts in automata theory and formal languages including grammar, finite automaton, regular expression, formal language, pushdown automaton and Turing machine.

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** terminology related to theory of computation.
CO2. **Explain** the basic concepts and applications of Theory of Computation.
CO3. **Apply** the principles of Theory of Computation to solve computational problems.
CO4. **Compare and contrast** the hierarchy of grammars.
CO5. **Design** various types of automata for given problems.

Course Content

Unit - I

Finite Automata and Regular Expressions: Finite State Systems, Basic Definitions Non-Deterministic finite automata (NDFA), Deterministic finite automata (DFA), Equivalence of DFA and NDFA Finite automata with E- moves, Regular Expressions, Equivalence of finite automata and Regular Expressions, Regular expression conversion and vice versa, Conversion of NFA to DFA by Arden'sMethod.

Unit - II

Introduction to Machines: Concept of basic Machine, Properties and limitations of FSM. Moore and mealy Machines, Equivalence of Moore and Mealy machines.
Properties of Regular Sets: The Pumping Lemma for Regular Sets, Applications of the pumping lemma, Closure properties of regular sets, Myhill-Nerode Theorem and minimization of finite Automata, Minimization Algorithm.

Unit - III

Grammars: Definition, Context free and Context sensitive grammar, Ambiguity regular grammar, Reduced forms, Removal of useless Symbols and unit production, Chomsky Normal Form (CNF), Griebach Normal Form (GNF).
Pushdown Automata: Introduction to Pushdown Machines, Application of Pushdown Machines

Unit - IV

Turing Machines: Deterministic and Non-Deterministic Turing Machines, Design of T.M, Halting problem of T.M., PCP Problem.
Chomsky Hierarchies: Chomsky hierarchies of grammars, Unrestricted grammars, Context sensitive languages, Relation between languages of classes.
Computability: Basic concepts, Primitive Recursive Functions.

Text and Reference Books:

1. Hopcroaft & O. D. Ullman, R Mothwani, Introduction to automata theory, language & computations, AW, 2001.
2. K. L. P. Mishra & N. Chandrasekaran, Theory of Computer Sc.(Automata, Languages and computation), PHI, 2000.
3. Peter Linz, Introduction to formal Languages & Automata, Narosa, Publication, 2001.
4. Ramond Greenlaw and H. James Hoover, Fundamentals of the Theory of Computation-Principles and Practice, Harcourt India Pvt. Ltd., 1998.
5. H. R. Lewis & C. H. Papaditriou, Elements of theory of Computation, PHC, 1998.
6. John C. Martin, Introduction to Languages and the Theory of Computation, T.M.H., 2003.

MCA – 35 Elective – I High Speed Network Technologies

General Course Information

Course Code: MCA-35 Elective 1 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites

Basic knowledge of computer networks, layers of OSI reference model, protocols at different layers of OSI reference model.

About the course:

High Speed Network Technologies is a professional core course based around Network Architectures, protocols used across the layers, techniques used in communication and modes of data transfer. The course deals with creating High Speed Networks for any organization/institute with its various phases/life cycles.

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** different high speed network technologies.
CO2. **Explain** working of different wired / wireless technologies suitable for LAN and WAN communication.
CO3. **Illustrate** the mapping of OSI reference model to different high speed technologies and Internet Suite of Protocols.
CO4. **Analyze** the performance of different high speed technologies in different scenarios / situations.
CO5. **Design** a network for any organization using high speed technologies along with Internet connectivity.

Course Content

Unit - I

Gigabit Ethernet: Overview of fast Ethernet, Gigabit Ethernet – overview, specifications, layered protocol architecture, frame format, network design using Gigabit Ethernet, applications, 10GB Ethernet – overview, layered protocol architecture, frame format.
Fiber Channel: Fiber channel – overview, topologies, ports, layered protocol architecture, frame structure, class of service.

UNIT - II

Frame Relay: Protocol architecture and frame format.
ISDN & B-ISDN: Channels, interfaces, addressing, protocol architecture, services.
ATM: Virtual circuits, cell switching, reference model, traffic management.

Unit - III

Wireless Networks: Existing and emerging standards, Wireless LAN (802.11), Broadband Wireless (802.16), Bluetooth (802.15) their layered protocol architecture and security. Mobile Networks – GSM, CDMA.

Unit - IV

Internet Layer: IPV4 and IPV6, IP addressing, IP classes, CIDR.
Transport Layer: UDP/TCP protocols & architecture, TCP connection management.
Application Layer: DNS, E-Mail, Voice over IP.

Text and Reference Books:

1. Jochen Schiller, Mobile Communication, 2nd Edition,Pearson,2009.
2. Andrew S Tanenbaum, Computer Networks, 5th Edition, Pearson2013.
3. William C Y Lee, Mobile Communication Engineering: Theory and Applications, 2nd Edition, McGraw Hill, 1997.

MCA-36 Machine Learning Lab.

General Course Information

Course Code: MCA-36 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Programming in Java, Python, R and Octave/MATLAB.

About the Course:

In this lab. course, students learn to solve optimization, supervised and unsupervised learning problems using machine learning tools. Students will use machine learning tools available in WEKA, R, Python and Octave etc. The lab experiments involve downloading datasets and applying machine learning techniques on these datasets. The course has a special focus on interpreting and visualizing results of machine learning algorithms.

Course Outcomes: By the end of the course students will be able to:

CO1. **Implement** machine learning algorithms using modern machine learning tools.
CO2. **Analyse** the trends in datasets using descriptive statistics.
CO3. A**pply** descriptive and predictive modelling.
CO4. **Compare and contrast** machine learning algorithms for a given problem. (describe datasets using descriptive statistics.
CO5. **Create** lab records of assignment by incorporating problem definitions, design of solutions, results and interpretations.
CO6. **Demonstrate** use of ethical practices, self-learning and team spirit.

List of experiments/assignments

1. Install WEKA/R/Python/Octave and learn to use these software packages.
2. Two assignments related to classification algorithms and interpreting the results of these algorithms.
3. Two assignments related to clustering algorithms and interpreting the results of these algorithms.
4. Three assignments on designing neural networks for solving learning problems.
5. Two assignments on ranking or selecting relevant features.
6. Two assignments on linear regression and logistic regression.
7. One assignment to be done in groups.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-37 Linux and Shell Script Lab

General Course Information

Course Code: MCA-37 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Basic programming skills.

About the Course:

This lab. course on data science involves a rigorous training on various Linux utilities. The objective of the lab course is to make students aware of shell scripting, sed scripts, awk programming.

Course Outcomes: By the end of the course students will be able to:

CO1. **Apply** commands related to vi and Emacs editors, general utilities and file systems.
CO2. **Write** basic shell scripts and use sed commands as well as awk programming.
CO3. **Analyse**the results of memory management and disk management commands.
CO4. **Evaluate** solutions for different operating system problems such as scheduling, memory management and file management.
CO5. **Create** lab record for assignments that includes problem definitions, design of solutions and conclusions.
CO6. **Demonstrate** use of ethical practices, self-learning and team spirit.

List of experiments/assignments:

1. Study of WINDOWS and Linux operating system (Linux kernel, shell, basic commands pipe & filter commands).
2. Study vi editor.
3. Administration of LINUX Operating System.
4. Writing of Shell Scripts (Shell programming).
5. AWK programming.
6. Write a C program to simulate different scheduling algorithms
7. Write a C program to simulate different file allocation strategies

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-38 Data Analytics Lab

General Course Information

Course Code: MCA-38 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Basic programming skills.

About the Course:

In this course, the learners will be able to develop working expertise of

1. To provide an overview of language R/Python/Excel used for data science.
2. To introduce students to the programming environment and research environments
3. To introduce the extended use of libraries and packages
4. To familiarize students with how various statistics concepts for data exploration
5. To familiarize with classification regression algorithm.
6. To make understand about data visualization techniques in R.

Course Outcomes: By the end of the course students will be able to:

CO1. **Apply** basic concepts related to data science
CO2. **Implementation** of data with statistical summaries and plots.
CO3. **Designing** predictive models.
CO4. **Analyse** the quality of a designed model fit.
CO5. **Interpret** and evaluate the result of predictive models.
CO6. **Evaluation** of predictive modelling.

List of experiments/assignments:

1. To provide an overview of a language used for data science.
2. Implement statistics like mean median etc. can be collected for data exploration
3. Write a small program to implement all basic concepts
4. Exercise with file handling packages.
5. Write a script for statistics techniques (mean, mode, median, variance, standard deviation)
6. Design user define functions to apply processes on dataset.
7. Write a script to work with tabular/ data frame using dataset.
8. Exercise with different types of popular packages.
9. Script to detect outlier and missing values.
10. Write scripts to implement predictive model.
11. Write a code to apply/analyse predictive model fitness.
12. Write scripts on classification and regression algorithms
13. Script for interpretation of confusion matrix from model.
14. Implement the sampling methods for dataset.
15. Implement Bootstrap method.
16. Script for cross validation example.
17. Draw all types of graph with example in different scripts

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-39 Industrial Training

General Course Information

Course Code: MCA-39 Course Credits: 2 Mode: Self learning in industry.	**Course Assessment Methods (100)** An internal evaluation is done by the course coordinator. Significance and originality of the problem addressed and the solution provided: 20 Knowledge of the problem domain and the tool used (VIVA-VOCE):25 Report Writing: 20 Judgment of the tools learnt and quality of the solution developed:20 Level of Ethics followed: 15

Prerequisite: Knowledge of UNIX, Windows, a programming language and data structures

About the Minor Project:
Students do a minor project in industry after second semester. They are expected to learn any tool/software and develop applications that can be completed within 4 to 6 weeks.

Course Outcomes: After doing minor projects students will be able to:

CO1. **Identify** a suitable problem from the environment around.
CO2. **Survey** the design of similar problems..
CO3. **Select** suitable application area/specialisation and modern IT tools.
CO4. **Address** the problem in an original and innovative manner.
CO5. **Communicate** orally as well as in written (minor project report) about the application developed.
CO6. **Engage** in ethical practices, individual and team work, and lifelong learning.

MCA-41 IoT and Cloud Computing

General Course Information

Course Code: MCA-41 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Basics of Computer Network, Distributed System.

About the Course:
The objective of the course is to give students a comprehensive view understanding of the vision and impact of IoT and Cloud, cloud and IoT Market perspective and IoT and Cloud architecture and IoT

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** concepts related to internet of things and cloud computing.
CO2. **Express** the current status and expected future directions of the internet of things and cloud.
CO3. **Apply** cloud computing techniques for various applications.
CO4. **Analyse** cloud computing services used at various levels.
CO5. **Identify and propose** applications which advance the cloud and IoT.
CO6. **Develop** cloud based applications which advance the IoT.

Course Content

Unit - I

Overview of Cloud Computing: Brief history and evolution - history of cloud computing, evolution of cloud computing, traditional vs. cloud computing, cloud service models (IaaS, PaaS & SaaS), cloud deployment models (public, private, hybrid and community cloud), benefits and challenges of cloud computing, introduction to AWS public cloud vendor, cost optimization in AWS, basics of virtualization, virtualization technologies, server virtualization, VM migration techniques, role of virtualization in cloud computing, introduction to EC2 service of AWS.

Unit - II

Working with Private Cloud: Private cloud definition, characteristics of private cloud, private cloud deployment models, private cloud vendors - CloudStack, OpenStack, Eucalyptus Microsoft, private cloud ± benefits and challenges, private cloud implementation in Amazon EC2 service.

Unit - III

Working with Public Clouds: What is public cloud, why public cloud, when to opt for public cloud, public cloud service models, public cloud players, infrastructure as a service offering, IaaS

vendors, PaaS offerings, PaaS vendors, software as a service, demonstrating public cloud with AWS ± storage and database services, private vs. public cloud ± when to choose.

Unit - IV

IoT Architecture- Introduction, State of the art: Architecture reference model- introduction, reference model and architecture, IoT reference model.

IoT Reference Architecture: IoT reference architecture- introduction, functional view, information view, deployment and operational view, other relevant architectural views, real-world design constraints- introduction, technical design constraints-hardware is popular again, data representation and visualization, interaction and remote control.

Text and Reference Books:

1. Jan Holler, Vlasios Tsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, "From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence", 1st Edition, Academic Press, 2014.
2. Hwang Kai, Fox Geoffrey C, Dongarra Jack G, "Distributed and Cloud Computing, From Parallel Processing to the Internet of Things", Morgan Kaufmann Publishers, 2011.
3. Rittinghouse John W. and Ransome James F., "Cloud Computing: Implementation, Management, and Security", CRC Press,2009.
4. Velte Toby, Velte Anthony, Elsenpeter Robert, "Cloud Computing, A Practical Approach", TMH, 2013.
5. Vijay Madisetti and Arshdeep Bahga, "Internet of Things (A Hands-on- approach)", 1st Edition, VPT, 2014.

MCA-42 Mobile Application Development

General Course Information

Course Code: MCA-42 Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Java Programming and Object-Oriented programming, Knowledge of RDBMS and OLTP.

About the Course:

Mobile Application Development has been introduced as a Professional Elective course for Students keeping in view the Employers' requirements. Android Platform forms the basis for developing Mobile Applications since the last decade as compared to IOS Platform for Apple Products. The Environment requires User Interface to be developed using Buttons, Check-Boxes, Alert Dialog and its kind.

Course Outcomes: By the end of the course students will be able to:

CO1. **State** basics of Android, its Evolution and its Architecture.
CO2. **Demonstrate** the Lifecycle of Software for Android Mobile Applications.
CO3. **Prepare** Mobile Applications on the Android Platform.
CO4. **Compare** working with Buttons and other Widgets for Visual Environment.
CO5. **Develop** Mobile Applications using data storage in SQLite Database and evaluate its Performance.

Course content

Unit - I

Mobile OS Architecture: Android, Blackberry OS, Firefox OS, IOS, Window OS, ARM and MIPS processor, Challenges of the mobile platform, Hello Android example, Internal Details, Dalvik VM, Software Stack, Android Core Building Blocks, Android Emulator, Android Manifest.xml, R.java file, Hide Title Bar, Screen Orientation.

Unit - II

UI Widgets: Working with Button, Toast, Custom Toast, Button, Toggle Button, Switch Button, Image Button, CheckBox, Alert Dialog, Spinner, AutoCompleteTextView, RatingBar, DatePicker, TimePicker, ProgressBar, Quick Contact Budge, Analog Clock and Digital Clock, Working with hardware Button, File Download.

Unit - III

Activity, Intent & Fragment: Activity Lifecycle, Activity Example, Implicit Intent, Explicit

Intent, Fragment Lifecycle, Fragment Example, Dynamic Fragment.
Android Menu: Option Menu, Context Menu, Popup Menu
Layout Manager: Relative Layout, Linear Layout, Table Layout, Grid Layout.

Unit - IV

Adaptor: Array Adaptor, ArrayList Adaptor, Base Adaptor.
View: GridView, WebView, ScrollView, SearchView, TabHost, DynamicListView, Expanded ListView.
SQLite: SQLite API, SQLite Spinner, SQLite ListView
XML & JSON: XML Parsing SAX, XML Parsing DOM, XML Pull Parser, JSON basics, JSON Parsing.

Text and Reference Books:

1. Redazione Io Programmo, Android Programming, 2011
2. John Horton, Android Programming for Beginners, packt publishing, 2015
3. Jason Wei, Android Database Programming, packt publishing, 2012
4. MarkLMurphy,AndroidProgrammingTutorials,3rdEdition, 2010
5. Bill Phillips et al., Android Programming - The "Big Nerd Ranch" Guide 2017
6. Rick Rogers et al., Android Application Development: Programming with the Google SDK, 2009

MCA – 43 Elective – II Big Data Analytics

General Course Information

Course Code: MCA-43 Elective II Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Basics of statistics and data mining.

About the Course:

This course aims to provide students with the knowledge of current challenges, methodologies and technologies in processing big data. Emphasis will be placed on the students' understanding of the rationales behind the technologies and the students' ability to analyse big data using professional packages and tools.

Course Outcomes: By the end of the course students will be able to:

CO1. **Recall** the concepts of big data analysis.
CO2. **Interpret** the outcomes of big data analysis.
CO3. **Apply** technical skills and modern tools for descriptive and predicative modelling.
CO4. **Analyse** a framework for visualization of big data analytics for business user.
CO5. **Examine** critically the results of mining to support business decision-making.
CO6. **Design** schemes for big data analytics for solving big data problems in efficient manner.

Course Content

Unit - I

Introduction: Overviews of Big Data, State of the Practice in Analytics, The Data Scientist, Big Data Analytics in Industry Verticals, Data Analytics Lifecycle Challenges of Conventional Systems, Statistical Concepts: Sampling Distributions, Re-Sampling, Statistical Inference, Prediction Error, Regression Modelling, Multivariate Analysis, Bayesian Modelling.

Unit - II

Mining Data Streams: Stream Data Model and Architecture, Stream Computing, Sampling Data in a Stream, Filtering Streams, Counting Distinct Elements in a Stream, Estimating Moments, Counting Oneness in a Window, Decaying Window, Real time Analytics, Platform (RTAP) Applications, Case Studies, Real Time Sentiment Analysis, Stock Market Prediction

Unit - III

Frequent Itemset and Clustering: Mining Frequent Itemsets, Market Based Model: Apriori Algorithm, Handling Large Data Sets in Main Memory, Limited Pass Algorithm, Counting

Frequent Itemsets in a Stream, Clustering based Techniques: Hierarchical, K-Means etc., Clustering High Dimensional Data, CLIQUE And PROCLUS, Frequent Pattern based Clustering Methods, Clustering in Non-Euclidean Space, Clustering for Streams and Parallelism..

Unit - IV

Frame works and Visualization: Overview of Map Reduce, Hadoop, Hive, MapR, Sharding, No SQL Databases, S3, HADOOP, Distributed File System (HDFS), Visualizations: Visual Data Analysis Techniques, Interaction Technique and Applications.

Text and ReferenceBooks:

1. Michael Berthold, David J. Hand, Intelligent Data Analysis, Springer, 2007.
2. A. Rajaraman, J.D. Ullman, Mining of Massive Datasets, Cambridge University Press, 2012.
3. Bill Franks, Taming the Big Data Tidal Wave: Finding Opportunities in Huge Data Streams with Advanced Analytics, John Wiley & sons, 2012.
4. Glenn J. Myatt, Making Sense of Data, John Wiley & Sons, 2007
5. Pete Warden, Big Data Glossary, O'Reilly, 2011.

MCA – 43 Elective – II Software Project Management

General Course Information

Course Code: MCA-43 Elective II Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Preliminary knowledge of Software Engineering.

About the Course:

The course involves training students in software project management and project planning. It focuses on the need for careful planning, monitoring and control for delivering quality projects in time. Besides this student learn to measure the success of a project in meeting its objectives.

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** basic concepts related to stepwise project planning.
CO2. **Demonstrate** the knowledge about Quality Control, Standard and Risk Management.
CO3. **Illustrate** the Activity Planning, and Resource Allocation Process.
CO4. **Apply** the concept of team structure and organization structure.
CO5. **Compare** various Project Evaluation and Estimation Techniques.
CO6. **Plan** activities necessary for completing the software projects successfully.

Course Content

Unit - I

Introduction to Software Project Management(SPM): Definition of Software Project, Software Project Vs Other types of projects, activities covered by SPM, categorizing software projects, project as system, management control, Requirement specification, Information and control in organization, project management life cycle.

Stepwise Project Planning: Introduction, selecting a project, identifying project scope and objectives, identifying project infrastructure, analysing project characteristics, identifying the project products and activities, estimate efforts for each activity, identifying activity risk, allocate resources, review/publicize plan.

Unit - II

Project Evaluation and Estimation: Cost-Benefit analysis, cash flow forecasting, cost benefit evaluation techniques, Selection of an appropriate project, choosing technologies, choice of process models, rapid application development, waterfall model, V process model and spiral model, Albrecht function point analysis.

Activity Planning: Objectives of activity planning, project schedule, projects and activities, sequencing and scheduling activities, network planning model.

Unit - III

Risk Management: Introduction, the nature of risk, managing risk, risk identification, risk analysis, reducing the risks, evaluating risks to schedule, calculating z-values**.**
Resource Allocation: Introduction, the nature of resources, identifying resource requirements, scheduling resources, creating critical paths.

Unit - IV

Managing Contracts and People: Introduction, types of contract, stages in contract placement, terms of contract, contract management, acceptance, managing people and organizing teams: Introduction, understanding organization behaviour: a background, selecting the right person for job, instruction in best methods, motivation, working in groups, becoming a team, decision making, leadership, organization structures.
Software Quality: Introduction, the place of software quality in project planning, the importance of software quality, defining software quality, McCall's software quality factors, product versus process quality management, external standards, techniques to enhance software quality.

Text and Reference Books:

1. Bob Hughes and Mike Cotterell , Software Project Management, Sixth Edition, TMH, 2018.
2. Walker Royce, Software Project Management, , Addison Wesley, 1998.
3. Pankaj Jalote , Software Project Management in Practice, Pearson, 2002.
4. Ramesh, Managing Global Software Projects, TMH, 2005.

MCA – 43 Elective –II Digital Image Processing

General Course Information

Course Code: MCA-43 Elective II Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: knowledge of basic linear algebra, basic probability theory, basic programming techniques and Fourier Transforms.

About the Course:

Digital Image Processing is a Professional Elective course that provides a theoretical foundation of digital image processing concepts. This course provides a mathematical foundation for digital manipulation of images, image acquisition, pre-processing, enhancement, segmentation and compression. Students learn algorithms that perform basic image processing operations (e.g., histogram processing, noise removal and image enhancement and restoration). Algorithms for image analysis (e.g., image compression, image segmentation and image representation) are explained.

Course Outcomes: By the end of the course students will be able to:

CO1. **State** concepts related to image acquisition and processing.
CO2. **Illustrate** the principles and methods in image processing.
CO3. **Apply** mathematical functions for digital manipulation of images such as image acquisition, pre- processing, segmentation, compression and representation.
CO4. **Compare** various image processing techniques.
CO5. **Assess** the various image processing techniques for a given problem.
CO6. **Design** and implement algorithms for digital image processing operations such as histogram equalization, filtering, enhancement, restoration and denoising, segmentation, compression.

Course contents

Unit - I

Introduction and fundamental to digital image processing: What is digital image processing, Origin of digital image processing, Examples that use digital image processing, Fundamental steps in digital image processing, Components of digital image processing system, Image sensing and acquisition, Image sampling, Quantization and representation, Basic relationship between pixels. Image enhancement in spatial domain and frequency domain: Background, Basic gray level transformation, Histogram processing, Basics of spatial filtering, Smoothing and sharpening spatial and the frequency domain filters.

Unit - II

Image Restoration: Image degradation/restoration Process, Noise models, Restoration in presence of noise, inverse filtering, Minimum mean square filtering, Geometric mean filter, Geometric transformations. Color Image Processing: Color fundamentals, Color models, Basics of full color image processing, Color transformations.

Unit - III

Image Compression: Fundamentals, Image compression models, Error free compression, Lossy compression. Image Segmentation: Detection of discontinuities, Edge linking and boundary detection, Thresholding, Region based segmentation.

Unit - IV

Representation, Description and Recognition: Representation-chain codes, polygonal approximation and skeletons, Boundary descriptors-simple descriptors, shape numbers, Regional descriptors-simple, topological descriptors.
Recognition: Pattern and Pattern classes.

Text and Reference Books:

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, Pearson Education, Ed, 2001.
2. Anil K. Jain, Fundamentals of Digital Image Processing, Pearson Education, PHI, 2001.
3. Tinku Acharya and Ajoy K. Ray, Image Processing-Principles and Applications, John Wiley & Sons, Inc., 2005.
4. Chanda and D. Dutta Majumdar, Digital Image Processing and Analysis, PHI, 2003.
5. Milan Sonka, Vaclav Hlavac, Roger Boyle, Image Processing, Analysis and Machine Vision, 2nd edition, PWS Publishing Company, Thomson Learning, 1999.

MCA – 44 Elective –III Soft Computing

General Course Information

Course Code: MCA-44 Elective III Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisites: Basic knowledge of Probability Theory, Set Theory and, Data Structure and Computer Algorithms.

About the Course:

We need to learn soft computing techniques to make intelligent machines that possess human like abilities to reason, learn and handle the uncertainty and vagueness often inherent in real world problems. Unlike conventional computing, soft computing techniques are tolerant of imprecision, uncertainty and approximations, and provide low cost, robust and tractable solutions to the complex real-world problems where conventional methods fail to do so. This introductory course on soft computing is going to cover Genetic Algorithms, Artificial Neural Networks and Fuzzy Logic.

Course Outcomes: By the end of the course students will be able to:

CO1. **Define** the terminology and concepts related to soft computing techniques.
CO2. **Discuss** soft computing techniques including genetic algorithms, fuzzy systems and neural networks.
CO3. **Solve** problems related to Genetic algorithms, Fuzzy logic and Neural Networks.
CO4. **Analyse** the design of Genetic Algorithms, Neural Networks and Fuzzy Systems.
CO5. **Justify** the design of a soft computing algorithm for a given problem.
CO6. **Design** Genetic Algorithms and Neural Networks to solve optimization and pattern recognition problems.

Course Content

Unit - I

Introduction to Soft Computing and related definitions: Defining soft computing, Differentiating the situations for application of hard and soft computing; Working of a simple Genetic Algorithm: Representation/Encoding Schemes, initializing a GA population, evaluation function, genetic operators, Function optimization using GA.
Study of parameters of genetic algorithms and its performance, sampling and selection mechanisms. Scaling of GA population.

Unit - II

Designing Genetic Algorithms for different applications: Different types encoding schemes, role of fitness function, different types of genetic operators, Designing GAs for numerical optimization,

knapsack problem and travelling salesperson and other similar problems.

Unit - III

Fuzzysets:Basicterminologyanddefinitions,OperationsonFuzzysets,MFformulationsandparameterisation, MFs of one and two dimensions, Derivatives of parameterised MFs, Fuzzy numbers, Extension principle and fuzzy relations, Operations on Fuzzy relations, Linguistic variables, Fuzzy If-Then Rules, Compositional rule of inference.

Unit - IV

Neural networks: Basic terminology and definitions, Model of an artificial neuron, Sigmoid function, Neural Network Architectures, Rosenblatt's Perceptron, Fixed increment perceptron learning algorithm for a classification problem, Examples of learning of AND/OR gate by perceptron, XOR problem. Back Propagation Neural Networks: Architecture of a backpropagation network, Model for multi-layer perceptron, Back propagation learning, Delta or gradient descent learning rule and effect of learning rate, Back propagation learning algorithm.

Text and Reference Books:

1. David. E. Goldberg, Genetic Algorithms in Search, Optimization and machine learning, Addison Wesley, 1999.
2. Zbigniew Michalewicz, Genetic algorithms + Data Structures = Evolution Programs, Springers-Verlag, 1999.
3. M. Mitchell, An Introduction to Genetic Algorithms, Prentice-Hall, 1998.
4. S. Rajasekaran & G. A. Vijayalakshmi Pai, Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications, PHI, 2003.
5. S. N. Sivanandam & S. N. Deepa, Principles of Soft Computing, Wiley - India, 2007.
6. J-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, PHI, 1997.
7. Simon O. Haykin, Neural Networks, A Comprehensive Foundation, PHI, 1994.

MCA – 44 Elective –III Compiler Design

General Course Information

Course Code: MCA-44 Elective III Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Prerequisites: Brief knowledge of programming languages, Data Structure, and Algorithm Design

About the Course:

Compilers have become part and parcel of today's computer systems. These are responsible for making the user's computing requirements, specified as a piece of program, understandable to the underlying machine. There tools work as interface between the entities of two different domains – the human being and the machine. The actual process involved in this transformation is quite complex. Compiler design covers basic translation mechanism and, error detection and recovery. It includes lexical, syntax, and semantic analysis as front end, and code generation and optimization as back-end.

Course Outcomes: By the end of the course students will be able to:

CO1. **Describe** principles of compiler design.
CO2. **Illustrate** the essential phases for automatically converting source code into object code.
CO3. **Apply** lexical analysis, syntax analysis and code optimization techniques for solving problems.
CO4. A**nalyse** a parse tree and a given BNF grammar.
CO5. **Compare and contrast** syntax-oriented translation schemes
CO6. **Design** a lexical analyser from the specification of a language's lexical rules.

Course Content

Unit - I

Introduction To Compilers: Compilers and translators, need of translators, structure of compiler its different phases, Compiler construction tools.
Lexical Analysis: Role of lexical analyzer, design of lexical analyzer, regular expressions, Specification and recognition of tokens, input buffering, A language specifying lexical analyzer. Finite automata, conversion from regular expression to finite automata, and vice versa, minimizing number of states of DFA, Implementation of lexical analyzer.

Unit - II

Syntax Analysis: Role of parsers, context free grammars, definition of parsing. Parsing Technique: Shift- reduce parsing, operator precedence parsing, top down parsing, predictive parsing.

Unit - III

LR parsers, SLR, LALR and Canonical LR parser. Syntax Directed Translations: Syntax directed definition, construction of syntax trees, syntax directed translation scheme and implementation of syntax directed translation, three address code, quadruples and triples.

Unit - IV

Symbol Table & Error Detection and Recovery: Symbol tables, its contents and data structure for symbol tables; trees, arrays, linked lists, hash tables. Errors, lexical phase error, syntactic phase error, semantic error.

Code Optimization & Code Generation: Code generation, forms of objects code, machine dependent code, optimization, register allocation for temporary and user defined variables.

Text and Reference Books:

1. Alfread V. AHO, Ravi Sethi and J.D. Ullman, Compilers Principle, Techniques and Tools, Addison Wesley, 2007.
2. Tremblay and Sorenson, Theory and practice of compiler writing, McGraw Hill, 1985.
3. Dhamdare, System software, MGH, 1986.
4. Alfred V. Aho, Jeffrey D. Ullman, Principles of Compiler Design, Narosa Publication, 2002.

MCA – 44 Elective –III Data Mining Techniques

General Course Information

Course Code: MCA-44 Elective III Course Credits: 3 Type: Professional Core Contact Hours: 3 hours/week Mode: Lectures (L) Exam Duration: 3 hours	**Course Assessment Methods (internal: 30; external: 70)** Two minor examinations each of 20 marks, Class Performance measured through percentage of lectures attended (4 marks) Assignment and quiz (6 marks), and end semester examination of 70 marks. The syllabus is divided into four units. For the end semester examination, nine questions are to be set by the examiner. Question number one is compulsory and contains seven short answer questions covering entire syllabus. Rest eight questions are set by giving two questions from each of the unit of the syllabus. A candidate is required to attempt any of four questions selecting at least one from each of the four units. All questions carry equal marks.

Pre-requisites: Knowledge of database systems, elementary knowledge of statistics and probability.

About the Course:

Today's era is the era of information. Data is growing exponentially day by day. There is a need to process and analyse the data to extract knowledge from it, so that one can use that knowledge for decision making. This course provides introductory concepts of data mining and data warehousing. The course will be taught with a database as well as machine learning perspectives. The objective of the course is to provide a comprehensive understanding of data prep-processing, data mining tasks and evaluation of results obtained out of data mining processes.

Course Outcomes: By the end of the course students will be able to:

CO1. **Outline** various types of data mining and data warehouse concepts and techniques.
CO2. **Explain** characteristics, architecture of a data warehouse, OLAP operations and data mining tasks.
CO3. **Apply** various pre-processing and data mining techniques for extracting valuable information from data.
CO4. **Evaluate** the descriptive and predictive data mining models.
CO5. **Plan** a data mining process for discovering knowledge from real-world databases.

Course Content

Unit - I

Introduction to Data Mining: Kind of data to be mined, Data Mining Functionalities, Technologies used in Data Mining, Applications of data Mining, Major Issues in Data Mining.
Data Pre-Processing: Need for preprocessing, Data Objects and Attribute types, Statistical description of data, Data Visualization, Measuring similarity and dissimilarity of data, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization.

Unit - II

Data Warehouse: Introduction, Data Warehouse and Database Systems, Data Warehouse Architecture, Data Warehouse Models, Data Cube and OLAP, Multidimensional data Model, Concept Hierarchies, OLAP operations, Data Warehouse Implementation

Unit - III

Mining Associations and Correlations: Mining Frequent Patterns, Associations and Correlations, Frequent Itemset Mining using Apriori Algorithm, Generating Association Rules from Frequent Itemsets. Improving efficiency of Apriori, Pattern Growth Approach for Mining Frequent Itemsets, Pattern evaluation Methods.
Advanced Pattern Mining: Pattern Mining in Multilevel and Multidimensional Space, Constraint-Based Frequent Pattern Mining.

Unit - IV

Classification: Introduction, Classification using Decision Tree Induction, Bayesian Classification Methods, Rule Based Classification, Model Evaluation and Selection, Techniques to Improve Classification Accuracy. Classification by Backpropagation, Support Vector Machines and Lazy Learners.
Cluster Analysis: Introduction, Basic Clustering Methods, Partitioning Methods, Hierarchical Methods, Evaluation of Clustering.

Text and Reference Books:

1. Jiawei Han, Micheline Kamber and Jian Pei, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, Third Edition, July 2011.
2. Alex Berson, Stephen J. Smith, Data Warehousing, Data Mining & OLAP, Tata McGraw Hill, 2004.
3. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, Introduction to Data Mining, Pearson Education, 2014.
4. K. P. Soman, Shyam Diwakar and V. Ajay, Insight into Data Mining Theory and Practice, Easter Economy Edition, Prentice Hall of India,2009.
5. G. K. Gupta, Introduction to Data Mining with Case Studies, Prentice Hall of India, 2006.
6. Daniel T. Larose, Data Mining Methods and Models, Wiley, 2006.
7. W. H. Inman, Building the Data Warehouse, Wiley India, 2005.

MCA-45 IoT and Cloud Computing Lab.

General Course Information

Course Code: MCA-45 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Basic programming skills.

About the Course:

This lab. course on IoT and Cloud Computing helps students to learn how to use cloud services, implement virtualization and task scheduling, apply the vision of IoT and understand IoT in applied form.

Course Outcomes: By the end of the course students will be able to:

CO1. **Analyse** the cloud computing setup with its vulnerabilities and applications using different architectures.
CO2. **Design** different workflows according to requirements and apply map reduce programming model.
CO3. **Identify** and propose applications which advance the IoT.
CO4. **Develop** applications which advance the IoT.
CO5. **Create** lab record for assignments that includes problem definitions, design of solutions and conclusions.
CO6. **Demonstrate** use of ethical practices, self-learning and team spirit.

List ofexperiments/assignments:

1. Amazon Simple Storage Service (Amazon S3) and Amazon Glacier Storage
2. Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block Store
3. Amazon Virtual Private Cloud (Amazon VPC)
4. Elastic Load Balancing, Amazon CloudWatch, and Auto Scaling
5. AWS Identity and Access Management (IAM)
6. Databases and AWS
7. SQS, SWF, and SNS
8. Domain Name System (DNS) and Amazon Route 53
9. Amazon ElastiCache
10. Additional Key Services
11. Security on AWS
12. MQTT, REST/HTTP, CoAP, MySQL, apache for handling HTTP Requests, PHP & MySQL for data processing, MongoDB object type database
13. HTML, CSS & jQuery for UI designing, JSON lib for data processing, security & privacy during development, Working with arduino and intel galileo boards/Rsapberry Pi

Note:
The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-46 Android Programming Lab.

General Course Information

Course Code: MCA-46 Course Credits: 2 Type: Professional Core Lab. Course Contact Hours: 2 hours/week Mode: Lab practice and assignments	**Course Assessment Methods (internal: 30; external:70)** The internal and external assessment is based on the level of participation in lab. sessions and the timely submission of lab experiments/assignments, the quality of solutions designed for the assignments, the performance in VIVA-VOCE, the quality of lab. file and ethical practices followed. The internal examination is conducted by the course coordinator. The external examination is conducted by external examiner (appointed by the Controller of Examination) in association with the internal examiner appointed by the Chairperson of the Department.

Pre-requisites: Java Programming and Object-oriented programming, knowledge of XML, JSON and database concepts.

About the Course:

This lab. course on Android Programming helps students to learn how to develop android apps.

Course Outcomes: By the end of the course students will be able to:

CO1. **Analyse** the Development Environment and the working of Emulator for android application.
CO2. **Design** different activities and layouts of application.
CO3. **Identify** and embed JSON and XML file in application design.
CO4. **Develop** application based on SQLite and latest connection providers.
CO5. **Create** lab record for assignments that includes problem definitions, design of solutions and conclusions.
CO6. **Demonstrate** use of ethical practices, self-learning and team spirit.

List of experiments/assignments:

1. Setting up development environment, Dalvik Virtual Machine & .apk file extension, Fundamentals: a. Basic Building blocks - Activities, Services, Broadcast Receivers & Content providers b. UI Components – Views& notifications c. Components for communication -Intents & Intent Filters, AndroidAPI levels (versions & version names).
2. Emulator-Android Virtual Device, Launching emulator, Editing emulator settings, Emulator shortcuts, Logcat usage, Introduction to DDMS, Second App:- (switching between activities) Develop an app for demonstrating the communication between Intents.
3. Design a Basic of UI structure, Form widgets, Text Fields, Layouts, [dip, dp, sip, sp] versus px, Menu, Option menu, Context menu, Sub menu, menu from xml, menu via code.
4. Implementation of Intents (in detail), Explicit Intents, Implicit intents with Examples
5. Styles & Themes, styles.xml, drawable resources for shapes, gradients (selectors), style attribute in layout file, Applying themes via code and manifest file.
6. SQLite Programming, SQLite Open Helper, SQLite Database, Cursor, Reading and updating Contacts, Reading bookmarks.
7. Notifications, Broadcast Receivers, Services and notifications, Toast, Alarms.

Note:

The actual experiments/assignments will be designed by the course coordinator. One assignment should be designed to be done in groups of two or three students. The assignments must meet the objective of the course and the levels of the given course outcomes. The list of assignments and schedule of submission will be prepared by the course coordinator at the beginning of the semester.

MCA-47 PROJECT WORK

General Course Information

Course Code: MCA-47 Course Credits: 6 Mode: Self learning under the guidance of a faculty member.	**Course Assessment Methods (internal: 30; external: 70)** Evaluation is done by the internal examiner (project guide) and external examiner appointed by Controller of Examination. The criteria for evaluation are given below. 1. Review of literature related to problem domain: 15 2. Significance and originality of the solution presented: 15 3. Application of software engineering principles and project management: 15 4. Significance and Scope of results: 20 5. Organization and presentation of major project report: 20 6. Level of Ethics and societal issues covered: 15

About the Project Work:

Students start working on their project work in the beginning of fourth semester. Students do the background research for identifying appropriate problems, methodology and tools for their respective project works. They prepare a synopsis of the project work to be carried out. Each student is required to prepare a synopsis in the format provided and present it in front of a committee constituted by the Chairperson of the Department. Students can carry out projects in groups of two. In case of group project, the size of the problem should be significant, and members of the group must specify their individual contribution.

After approval by the internal committee, they continue working on their project work throughout 4th semester. They carry out implementation of their respective projects based on the problem identified, methodology and tools suggested in the approved synopsis. They are required to complete their project work by the end of 4th semester. They prepare the final project reports according to the format provided. At the end of fourth semester, each student is required to present his/her project work in front of internal project guide and external examiner appointed by Controller of Examination.

Course Outcomes: After doing Project students will be able to:

CO1. **Evaluate** critically the existing solutions and methodologies through reviewing literature.
CO2. **Plan** the project according to principles of project management.
CO3. **Devise** original solutions to complex problems using modern tools.
CO4. **Justify** the outcomes of the project work.
CO5. **Organize** and communicate (written and oral) ideas effectively.
CO6. **Develop** solutions that meet ethical, societal and legal considerations.

Department of Computer Science and Engineering

Guru Jambheshwar University of Science and Technology, Hisar-125001

Name of the Programme:_____ Semester:_____ Session:_____	Credits: 6 Total Marks: 100

Evaluation of Project Work (MCA-47)

SR. No.	Roll. No.	Review of literature related to problem domain CO1 (15)	Application of principles of software engineering and project management CO2 (15)	Significance and originality of the solution presented CO3 (15)	Significance and Scope of the Results CO4 (20)	Organisation and presentation of major project report CO5 (20)	Level of Ethics followed and societal issues covered CO6 (15)
1							
2							
3							

Name of the external examiner: Signature of the External Examiner: Date:	Name of the internal examiner: Signature of the internal Examiner: Date:	Total Candidates: No. of Candidates Present: No. of Candidates Absent:

Guidelines for preparing Project Work (Synopsis)

All the students are required to follow these guidelines for preparing their project synopsis.

General Guidelines

1. The student should follow ethical practices while doing the synopsis work. Any violation of ethical practices will lead to rejection of the synopsis. For instance, a plagiarized synopsis or a readymade synopsis purchased from market will be rejected straight away.
2. The synopsis must be submitted to the internal guide in soft binding at least 7 days before the presentation so that he/she can suggest changes.
3. Synopsis carried out in groups of two students must include the division of work.

Formatting Instructions

The formatting instructions are given in Table below.

Formatting Instructions		
Sr. No.	Item	Formatting
1.	No. of pages	Minimum 6 and maximum 10
2.	Paper size	A4
3.	Font Type	Times New Roman
4.	Normal text size	12
5.	Page numbering	Place: Centre Bottom
6.	Margins	Left margin: 3.75 cms (1.5 inch) Right, bottom, top= 2.5 cms (1 inch)
7.	References/Bibliography	IEEE format
8.	Binding	Soft binding of good quality

Contents of the Project Work (Synopsis)

The synopsis must be written in English. The ideas must be organized in a clear and concise fashion. Sections must be tentatively organized as below.

1. Contents Page
2. Introduction
3. Background Details and Literature Review
4. Problem Formulation and Objectives
5. Methodology and tools to be used
6. References/Bibliography

Signature
Name of Student
Registration Number
Department of Computer Science and Engineering
Guru Jambheshwar University of Science and Technology, Hisar

Signature
Supervisor
Designation
Department of Computer Science and Engineering
Guru Jambheshwar University of Science and Technology, Hisar

Format of Title Page

The format for the title page of the synopsis is given on the next page

TITLE OF THE PROJECT WORK (SYNOPSIS)

(Write inTimes New Roman, 16-point size, Bold and Centred andUppercase font)

Project Work (synopsis) submitted to
Guru Jambheshwar University of Science and Technology, Hisar
for the partial award of the degree

(Write inTimes New Roman, 12-point size font, Bold, Italics and Centred style after 4 lines gap with 12 font size from the title of the project)

of

(Write in Times New Roman, 12-point size font, Bold, Italics and Centred style after 2 lines gap with 12 fontsize from the text above in three lines)

Master of Computer Applications

(Write in Times New Roman, 14-point size, Bold, Centredstyle after "***of***" after 2 lines gap with 12 font size)

by

(Write in Times New Roman 12-point size, Bold, Italics, and Centredstyle after the name of the degree with 2 lines gap with 12 font size)

Your Name
(Enrolment Number)

Supervisor Name
Designation

(Write inTimes New Roman, 14-point size font, Bold, Centred style after 2 lines gap with 12 font from "***by***")

Department of Computer Science &Engineering
GURU JAMBHESHWAR UNIVERSITY OF SCIENCE AND TECHNOLOGY, HISAR

Month, Year

(Write in Times New Roman, 14-point size font, Bold, Centredstyle, after 2 lines gap from logo)

Guidelines for preparing Project Work Report

All the students are required to follow these guidelines for preparing their final project report.

General Guidelines:

1. The title of the project must be same as that of the title in the synopsis submitted.
2. The report must include a declaration by the student that he/she has followed ethical practices while doing the project work. Any violation of ethical practices will lead to rejection of the report. For instance, a plagiarized report or a readymade report purchased from market will be rejected straight away.
3. Project works carried out in groups of two students must include the individual contribution of the students.
4. A CD of the project work should be included in closed pocket inside the back cover page. The CD must bear the name, registration number and title of the project.
5. The report must be submitted to the internal guide in soft binding at least 10 days before the final examination so that he/she can suggest changes before the report is presented to external examiner.

Formatting Instructions

The formatting instructions are given in Table below.

Formatting Instructions		
Sr. No.	Item	Formatting
1.	Front Cover	Dark Green and contents in golden ink
2.	No. of pages	Minimum 40 and maximum 70 excluding front material
3.	Paper size	A4
4.	Font Type	Times New Roman
5.	Chapter Heading Font	16
6.	Font of Sections and Subsections	14 and 12 in bold style
7.	Numbering style for sections and subsections; Do not use more than three levels.	2., 2.1 and 2.1.1
8.	Normal text size	12
9.	Figures and Tables must be numbered chapter-wise. Table headings on the top of the tables and Figure heading at the bottoms of the figures.	For example for chapter 2, Figures should be numbered as Fig. 2.1, Fig. 2.2 etc. and Tables as Table 2.1 and Table 2,2 etc.
10.	Page numbering	Place: Centre Bottom Type: Front material in Roman numbers Body of the report: in Arabic numerals. Pagination must start with first page of the first chapter and continue throughout the end of the report.
11.	Margins	Left margin: 3.75 cms (1.5 inch) Right, bottom, top= 2.5 cms (1 inch)
12	References/Bibliography	IEEE format
13	Binding	Hard binding of good quality

Contents of the Project Report

The contents of the report should be organised as described below.

1. The first page in the report should be same as the cover page.
2. Declaration that the students has carried out his work on his own. It is his/her original creation, not plagiarised from any other source and due credit has been given to the source material used in the report through references and citations.
3. Acknowledgement
4. List of figures

5. List of Tables
6. List of Abbreviations
7. Contents

   Abstract (in not more than 250 words)

   This answers the question what have you done? How have you done and brief indication about the results.
8. Body of the Report

   The report must be written in English. The ideas must be organised in a clear and concise fashion. Chapters must be tentatively organised as below.

   Chapter 1. Introduction

   This includes introduction to relevant area of project, problem formulation objectives of the project, and structure of the project report.

   Chapter 2. Background Details and Literature Review

   Chapter 3. Design or Framework of the project work

   Methodology, Data Flow Diagrams, Entity Modelling etc.

   Chapter 4. Discussion and Analysis of Results

   Discussion and comparison of results.

   Chapter 5. Conclusion and Future Scope

   This includes relevance and scope of the project work, and its extensions.

   References/Bibliography
9. Appendices

Declaration to be Submitted

The format of declaration to be included in the project report is given on next page.

Format of the Title Page

The format of the title page for the Project Work is given on the next to next page

DECLARATION

I, *Your Name, Your Roll No.*, certify that the work contained in this project report is original and has been carried by me under the guidance of my supervisor. This work has not been submitted to any other institute for the award of any degree or diploma and I have followed the ethical practices and other guidelines provided by the Department of Computer Science and Engineering in preparing the report. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Signature
Name of Student
Registration Number
Department of Computer Science and Engineering
Guru Jambheshwar University of Science and Technology, Hisar

Signature
Supervisor
Designation
Department of Computer Science and Engineering
Guru Jambheshwar University of Science and Technology, Hisar

TITLE OF THE PROJECT REPORT

(Write in Times New Roman, 16-point size, Bold and Centred and Uppercase font)

Project report submitted to
Guru Jambheshwar University of Science and Technology, Hisar
for the partial award of the degree

(Write in Times New Roman, 12-point size font, Bold, Italics and Centred style after 4 lines gap with 12 font size from the title of the project)

of

(Write in Times New Roman, 12-point size font, Bold, Italics and Centred style after two lines gaps with 12 fontsize from the text above in three lines)

Master of Computer Applications

(Write in Times New Roman, 14-point size, Bold, Centred style after "***of***" after 2 line gaps with 12 font size)

by

(Write in Times New Roman 12-point size, Bold, Italics, and Centred style after the name of the degree with 2 lines gap with 12)

Your Name **Supervisor Name**

(Enrolment Number) **Designation**

(Write in Times New Roman, 14-point size font, Bold, Centred style after 2 lines gap with 12 font from "***by***")

Department of Computer Science &Engineering
GURU JAMBHESHWAR UNIVERSITY OF SCIENCE AND TECHNOLOGY, HISAR

Month,Year