SURESH GYAN VIHAR UNIVERSITY, JAIPUR
(CDOE, SGVU)

Program Project Report (PPR)

MASTER OF COMPUTER APPLICATION

Table of Contents

Program Project Report

1. Program Mission and Objectives

Suresh Gyan Vihar University, Jaipur, established in 2008, is a leading private University of Rajasthan. SGVU, Jaipur is accredited with Grade A+ by the National Assessment and Accreditation Council (NAAC), and offers courses like Engineering, Management, Hotel Management, Pharmacy, Arts, Humanities, Law, Agriculture, etc. in conventional mode. SGVU is renowned for its innovative academic practices, brilliance in technical education, and consultancy to high-profile industries.

The program's mission is to impart, train, and transform a student completely for high caliber competence through the latest concepts and technology and equip the students as per the demands of the industry.

The program aims to achieve the following objectives

i. To provide an opportunity to get an MCA degree to those who find it difficult or even impossible to pursue regular MCA courses at a university either due to their job commitments or certain other circumstances.

ii. To help the learners, study at their own pace, from their own chosen place.

iii.To provide students with an in-depth understanding of their chosen field of study, including current theories, research methodologies, and significant developments.

iv. To develop students' abilities to critically evaluate existing literature, arguments, and evidence within their field.

v. To encourage the integration of knowledge from various disciplines, promoting a more holistic understanding and innovative approaches to solving complex problems.

vi. To instill a strong sense of ethical responsibility and an understanding of the ethical implications of research and professional practice within their discipline.

2. Relevance of the Program with Suresh Gyan Vihar University, Jaipur Mission and Goals

Suresh Gyan Vihar University (SVGU) was established with a vision to become a University with a commitment to excellence in education, research, and innovation aimed towards human advancement.

The proposed program is highly relevant to the SVGU's mission i.e.

- Facilitate holistic education through knowledge sharing, skilling, research, and development.
- Integrate academic and research work towards the nation's development.
- Mentor students' physical, mental, emotional, secular, and spiritual attributes to become a valued human resource as it aims to provide quality education to those aspiring candidates who are deprived of higher education due to the limited number of intakes in the conventional mode of education in the Universities.

Moreover, to keep the quality intact the curriculum and syllabus have been designed at par with the conventional mode keeping in mind the specific needs and acceptability of the learners' ODL mode and in keeping with the aims and objectives of the University also ensuring the industry and future skills relevance.

Nature of Prospective Target Group of Learners

The curriculum of MCA is designed in such a way that it helps the students to become not only more employable but also encourages them to become entrepreneurs. Primarily the target group of learners will be:

- Those deprived of admission in the regular mode due to limited intake capacity.
- Those employed in various organizations who desire to pursue higher education as a passion or as a means for movement up the promotional ladder.
- Dropouts primarily due to social, financial and economic compulsions as well as demographic reasons.
- Population of any age and those living in remote areas where higher education institutes are not easily accessible.

3. Appropriateness of program to be conducted in ODL mode to acquire specific skills and competence

The degree would be of most value to students which can support the development of critical thinking, research skills, and subject-specific knowledge. In various fields such as education, business, social sciences, humanities, and public administration, it provides professionals with the opportunity to acquire advanced theoretical knowledge and practical skills that are directly applicable to their work environments.

4. Instructional Design

Curriculum Design

The curriculum is designed by experts in the field of Arts and has taken into account to include relevant topics that are contemporary and create environmental awareness. It is approved by the BoS (Board of Studies), the CIQA (Centre for Internal Quality Assurance), and the AC (Academic Council) of the university.

Program Structure and Credits Mapping: MCA

Semester	Course Code	Paper	Credit	Contact Hours	Internal	External	Total
1	MCA-101	Computer Fundamental	06	15	30	70	100
1	MCA-102	Introduction to Software	06	12	30	70	100
1	MCA-103	Data Structure through C	06	12	30	70	100
1	MCA-104	Elements of system analysis and design	06	15	30	70	100
1	MCA-105	Introduction to Database Management System	06	15	30	70	100
2	MCA-201	Introduction to computer Organization	06	15	30	70	100
2	MCA-202	Introduction to software Engineering	06	12	30	70	100

2	MCA-203	Computer Oriented Numerical method	06	12	30	70	100
2	MCA-204	Theory Of Computer Science	06	15	30	70	100
2	MCA-205	Practical based on MCA-201, MCA-202	06	12	30	70	100
3	MCA-301	Web Development Tools	06	15	30	70	100
3	MCA-302	Mobile Application Development	06	12	30	70	100
3	MCA-303	Object Oriented Technology	06	15	30	70	100
3	MCA-304	Cyber Security	06	12	30	70	100
3	MCA-305	Practical based on MCA-301, MCA-303	06	12	30	70	100
4	MCA-401	Software Development Project	06	12	30	70	100
Total Credits			**96**				

Contact Hours at campus mentioned above are other than the PCP (Personal Contact Program) conducted atcampus.

Programme Outcomes:

After the completion of programme the students are able to

PO1. Choose modern computing tools, skills and techniques necessary for innovative

software solutions

PO2. Relate practical skills to provide solutions to industry, society and business.

PO3. Enumerate the creative ideas with the emerging technologies in real time applications.

PO4. Interpret complex business scenarios and contemporary issues in emerging technologies.

PO5. Classify the computing systems through quantitative and qualitative technique.

PO6. Associate techniques necessary for innovative software solutions.

Programme Specific Outcomes:

Upon completing the MCA degree students have/capable of:

PSO1. Enumerate technical skills in computer application fields
PSO2. Relate the innovative ideas in required real-time applications.
PSO3. Integrate multi-disciplinary creativity in a modernized organization.

SYLLABUS
MCA (Semester - I)

Computer Fundamental (MCA-101)

Learning Outcomes:

This is a course on Computer Fundamentals and Programming that aims at exposing the students to understand how computer works, analytical skills to solve problems using computers. And to write computer programs using C.

Unit I

Functional units of computer, I/O devices, primary and secondary memories. Number systems: decimal, octal, binary and hexadecimal; Representation of integers, fixed and floating point numbers, Operator precedence, character representation; ASCII, Unicode.

Unit II

Programming Fundamentals with C - Algorithm, techniques of problem solving, flowcharting, stepwise refinement; Constants and variables; Data types: integer, character, real, data types; Arithmetic expressions, assignment statements, logical expressions. Control flow

Unit III

Arrays and structures. Pointers, dynamic memory allocations

Unit IV

Program Structures – functions, subroutines

Unit V

I/O operations, Program correctness; Debugging and testing of programs.

Reference:

• Balaguruswamy E. 2019. Programming with ANSI C. Tata McGraw Hill.
• Gottfried B. 2017. Programming with C, Schaum Outline Series. Tata McGraw Hill.
• Kanetkar Y. 1999. Let Us C. BPB Publ.

Introduction to Software (MCA-102)

Learning Outcomes

- Distinguish between Operating Systems software and Application Systems software.
- Describe commonly used operating systems.
- Identify the primary functions of an Operating System.
- Describe the "boot" process.
- Identify Desktop and Windows features.
- Use Utility programs.

Unit-I:

Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional Engineering Processes, Software Quality Attributes. Software Development Life Cycle (SDLC) Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.

Unit-II:
Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Decision Tables, SRS Document, IEEE Standards for SRS. Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.

Unit-III:
Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halestead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.

Unit-IV:
Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products. Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards.

Unit-V:
Software Maintenance and Software Project Management,Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software ReEngineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Estimation of Various Parameters such as Cost, Efforts, Schedule/Duration, Constructive Cost Models (COCOMO), Resource Allocation Models, Software Risk Analysis and Management.

References:
1. R. S. Pressman, Software Engineering: A Practitioners Approach, McGraw Hill.
2. Rajib Mall, Fundamentals of Software Engineering, PHI Publication.
3. K. K. Aggarwal and Yogesh Singh, Software Engineering, New Age International Publishers.
4. Pankaj Jalote, Software Engineering, Wiley.
5. Carlo Ghezzi, M. Jarayeri, D. Manodrioli, Fundamentals of Software Engineering, PHI Publication.
6. Ian Sommerville, Software Engineering, Addison Wesley.
7. Kassem Saleh,"Software Engineering", Cengage Learning.
8. Pfleeger, Software Engineering, Macmillan Publication.

Data Structure through C (MCA-103)

Learning Outcome:
- While studying the Data Structures course, the student shall be able to:

- Learn the concept of data structures through ADT including List, Stack, and Queues.
- Design and implement various data structure algorithms.
- Understand various techniques for representation of the data in the real world.
- Able to develop application using data structure algorithms

Unit:1

Definition of Algorithm and Data structure- Types of Data structure (linear and non- linear data structure) - Linear data structure: Array, Stack, and Queue - Linked List-doubly linked list and circular list - Recursive and non recursive algorithm.

Unit: 2
Binary Tree – notations, terminology, Representation, Binary tree Traversal and Application - Graph-Notations, Terminology-Representation, Traversal and Application. Performance analysis of an algorithm.-Tabular method.

Unit: 3
Min/Max heaps – Deaps – Leftist Heaps - Binomial Heaps – Fibonacci Heaps - Skew Heaps – Lazy-Binomial Heaps.

Unit: 4
Binary Search Trees – AVL Trees - Red-Black trees – Multi-way Search Trees - B-Trees– Splay Trees – Tries.

Unit:5
Segment Trees – k-d Trees - Point Quad Trees – MX-Quad Trees - R-Trees – TV Trees

Reference books:
Á E. Horowitz, S.Sahni and Dinesh Mehta, Fundamentals of Data structures in C++, University Press, 2007.
Á E. Horowitz, S. Sahni and S. Rajasekaran, Computer Algorithms/C++, Second Edition, University Press, 2007.
Á G. Brassard and P. Bratley, Algorithmics: Theory and Practice, Printice–Hall, 1988.
Á V.S. Subramanian, Principles of Multimedia Database systems, Morgan Kaufman, 1998

Elements of system analysis and design (MCA-104)

Learning Outcome:
- While studying the System Analysis and Design course, the student shall be able to:
- Gather data to analyze and specify the requirements of a system.
- Design system components and environments.
- Build general and detailed models that assist programmers in implementing a system.
- Design a database for storing data, a user interface for data input and output,
- and controls to protect the system and its data

Unit:1
Introduction – Definition of a System – Characteristics of a system – Elements of Systems Analysis – System development life cycle - Software crisis – Role of Systems Analyst – Project Selection : Project request – Managing Project selection – Preliminary investigation – Problem classification and definition - Feasibility study : Types of feasibility – Investigative study – Cost Benefit Analysis – Fact finding techniques - DFD –Data Dictionaries – HIPO – Decision tables and Decision Trees – Warnier

Orr Diagrams.

Unit: 2
Introduction – Design Methodologies – Structured Design – Modularization – Design process – Systems Specifications – Prototype design - Input design and control: Elements of Input data – Processing transaction data – Design guidelines – Input verifications and control – Layout of Terminal screen – Output System design – Output devices – Types of Output – Designing screen output/report - Form design – File and Database design – Types of file – File Organisation – File design – Database Design – Coding system – Types of Code.

Unit: 3
Task of System development – Selection of Hardware and Software – Benchmark testing - Software selection criteria – Quality Assurance – Levels – Maintenance Issues
- Levels of Test – Testing plan – Designing test data – System control.

Unit: 4
Documentation Characteristics – Types of Documentation – Need for documentation Tools – System Implementation - Conversion methods – Post Implementation Review – Review Plan – System Maintenance

Unit: 5
Attributes of a Good Analyst – Organisational Issues - Communicating with Computers – Ergonomics – Human problems in automated office - Multimedia: Introduction
– Components of Multimedia – Hardware and Software requirements.

Reference Books:

Á James. A. Senn, "Systems Analysis and Design, McGraw-Hill, Inc. Professional Book Group New York, 1989.
Á Elias M. Awad, "Systems Analysis and Design McGraw-Hill Professonal, 1985**.**

Introduction to Database management system (MCA-105)

Learning Outcomes:

While studying the Database Management Systems course, the student shall be able to:

- Model an application's data requirements using Entity Relationship Model and
- design Relational Database Schema based on the Entity Relationship Model.
- Develop SQL queries to retrieve data.
- Understand the need for database normalization
- Understand the strategies for query processing
- Gain knowledge on the Basics of Transactions, Need for Concurrency Control and Need for Recovery.

Unit: 1
Data Models – Entity Relationship Model – Entity Types – Entity Sets – Attributes – Keys – Relationship Types – Relationship Sets – Roles – Structural Constraints – Strong & Weak Entity Types – Enhanced Entity Relationship Model – Specialization
and Generalization – Constraints and Characterization of Specialization and Generalization Hierarchies.

Unit: 2
Relational Model Concepts – Super Key – Candidate Key – Primary Key – Alternate Key – Foreign Key – Relation Model Constraints and Relational Database Schemas –

Update Operations – Transactions and Dealing with Constraint Violations – Unary and Binary Relational Operations in Relational Algebra – Queries in Relational Algebra – Basic SQL – SQL Data Definition and Data Types – Specifying Constraints in SQL – Basic Retrieval Queries in SQL – Insert, Delete and Update Statements in SQL – Views in SQL – Nested Queries – Triggers.

Unit: 3
Mapping Entity Relationship Model to Relations – Mapping Enhance Entity Relationship Model to Relations – Database Normalization – Functional Dependencies – First Normal Form – Second Normal Form – Third Normal Form – Boyce-Codd Normal Form – Fourth Normal Form and Fifth Normal Form.

Unit: 4
Translating SQL Queries into Relational Algebra and Other Operators – Algorithms for External Sorting – Algorithms for SELECT Operation - Implementing the Join Operation – Algorithms for Project and Set Operations – Implementing Aggregate Operations and Different types of JOINs - Combining Operations Using Pipelining – Parallel Algorithms for Query Processing.

Unit: 5
Introduction – Transaction and System Concepts – Desirable Properties of Transactions – Characterizing Schedules Based on Recoverability / Serializability – Transaction Support in SQL-Two-Phase Locking Techniques for Concurrency Control – Concurrency Control Based on Timestamp Ordering - Multiversion Concurrency Control
Techniques – Optimization Techniques and Snapshot Isolation Concurrency ControlGranularity of Items and Multiple Granularity Locking – Using Locks for Concurrency
Control in Indexes - Recovery Concepts – NO-UNDO / REDO Recovery Based on Deferred Update – Recovery Techniques Based on Immediate Update.

Reference Books:
1. Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, Seventh Edition, Pearson Education, 2017.
Master of Computer Applications 27
2.Á Abraham Silberschatz, Henry F. Korth, S. Sudharshan, “Database System Concepts”, Sixth Edition, Tata McGraw Hill, 2014.
3.Á C. J. Date, A. Kannan, S. Swamynathan, “An Introduction to Database Systems”, Eighth Edition, Pearson Education, 2006.
4.Á Raghu Ramakrishnan, Johannes Gehrke, “Database Management Systems”, Fourth Edition, Tata McGraw Hill, 2010

Introduction to Computer Organization (MCA-201)

Learning Outcomes:

After completion of the Computer Architecture, the student can able to:

- Understand the generations and classifications of computer
- Differentiate the instruction format and types
- Utilize the concept of different algorithms
- Use different memories for computer architecture
- Learn the pipelining concepts used in society

Unit: 1
Basic of Computer, Von Neumann Architecture, Generation of Computer, Classification of Computers - Instruction Executions, Register Transfer and Micro operations: Register Transfer, Bus and Memory Transfers.-Three-State Bus Buffers, Memory Transfer, Micro-Operations, Register Transfer Micro-Operations - Arithmetic Micro-Operations, Logic Micro-Operations, Shift Micro-Operations.

Unit: 2
Stack Organization, Register Stack, Memory Stack, Reverse Polish Notation - Instruction Formats, Three- Address Instructions, Two – Address Instructions, One - Address
36 Master of Computer Applications Instructions, Zero - Address Instructions, RISC Instructions-Addressing Modes. RISC & CISC and their characteristics.

Unit: 3
Addition and Subtraction With Signed-Magnitude, Multiplication Algorithm, Booth Multiplication Algorithm - Array Multiplier, Division Algorithm, Hardware Algorithm, Divide Overflow, Floating-Point - Arithmetic Operations, Decimal Arithmetic Operations, BCD Adder, BCD Subtraction.

Unit: 4
Modes f Transfer, Priority Interrupt, DMA, Input-Output Processor (IOP), CPU-IOP Communication - Memory Organization: Memory Hierarchy, Main Memory - Auxiliary Memory, Cache Memory, Virtual Memory, Associative Memory.

Unit: 5
Control memory – Address sequencing – Design of Control unit. Pipelining: Parallel Processing, Pipelining - Arithmetic Pipeline, Instruction Pipeline - Multiprocessors: Characteristics of Multiprocessors, Interconnection Structure: Time-Shared Common Bus, Multi-Port Memory - Crossbar Switch, Multistage Switching Network, Hypercube Interconnection.

References:
Á M.Morris Mano, “Computer System Architecture” Prentice Hall, 1993
Á John. P. Hayes, “Computer System Architecture”.
Á C. Hamacher, Z. Vranesic, S.Zaky, “Computer Organization.
Á “Computer Architecture and parallel Processing “, Hwang K. Briggs.

Introduction to Software Engineering (MCA-202)

Learning Outcomes:
After completion of the Software Engineering Methodologies, the student can able to:
- Know the Fundamentals of Software Engineering, Umbrella Activities in Software Engineering and the need for Software Process Models.
- Acquire Knowledge on Requirements Engineering
- Learn Analysis and Design Methods.
- Gain Knowledge on Software Testing Strategies and Software Configuration
- Management.
- Learn the Need for Risk Management and gain Knowledge on Software Maintenance and Reengineering.

Unit: 1
The Nature of Software – Defining Software – Software Application Domains – Legacy Software – The Changing Nature of Software – WebApps – Mobile Applications – Cloud Computing – Product Line Software – Defining the Discipline – The Software Process

– The Process Framework – Umbrella Activities – Process Adaptation – Software Engineering Practice – The Essence of Practice – General Principles – Software Development Myths – Software Process Structure – A Generic Process Model – Defining a Framework Activity – Identifying a Task Set – Process Patterns – Process Assessment and Improvement – Prescriptive Process Models – The Waterfall Model – Incremental Process Models – Evolutionary Process Models – Concurrent Models – A Final Word on Evolutionary Processes – Specialized Process Models – Component-Based Development – The Formal Methods Model – Aspect-Oriented Software Development – The Unified Process – Phases of the Unified Process – Personal and Team Process Models – Personal Software Process – Team Software Process – Process Technology – Product and Process – Agility – Agility and the Cost of Change – Agile Process – Agility Principles – The Politics of Agile Development – Extreme Programming – The XP Process – Industrial XP – Other Agile Process Models – Scrum – Dynamic Systems Development Method – Agile Modeling – Agile Unified Process.

Unit: 2

Modeling – Principles That Guide Process, Practice and Each Framework Activity – Communication, Planning, Modeling, Construction and Deployment Principles – Requirements Engineering – Establishing the Groundwork – Identifying Stakeholders – Recognizing Multiple Viewpoints – Working toward Collaboration – Asking the First Questions – Nonfunctional Requirements – Traceability – Eliciting Requirements – Collaborative Requirements Gathering – Quality Function Deployment – Usage Scenarios – Elicitation Work Products – Agile Requirements Elicitation – Service-Oriented Methods – Developing Use Cases – Building the Analysis Model – Elements of the Analysis Model – Analysis Patterns – Agile Requirements Engineering – Requirements for Self-Adaptive Systems – Negotiating Requirements – Requirements Monitoring – Validating Requirements – Avoiding Common Mistakes – Requirements Analysis – Overall Objectives and Philosophy – Analysis Rules of Thumb – Domain Analysis – Requirements Modeling Approaches – Scenario-Based Modeling – Creating a Preliminary Use Case – Refining a Preliminary Use Case – Writing a Formal Use Case – UML Models That Supplement the Use Case – Developing an Activity Diagram – Swimlane Diagrams – Requirements Modeling: ClassBased Methods – Identifying Analysis Classes – Specifying Attributes – Defining Operations – Class-Responsibility-Collaborator Modeling – Associations and Dependencies – Analysis Packages – Requirements Modeling for Web and Mobile Apps.

Unit: 3

Software Design – Design within the Context of Software Engineering – The Design Process – Software Quality Guidelines and Attributes – The Evolution of Software Design – Design Concepts – Abstraction – Architecture – Patterns – Separation of Concerns – Modularity – Information Hiding – Functional Independence – Refinement – Aspects – Refactoring – Object-Oriented Design Concepts – Design Classes – Dependency Inversion – Design for Test – Design Model – Data Design Elements – Architectural Design Elements – Interface Design Elements – Component-Level Design Elements – Deployment-Level Design Elements – Designing Class-Based Components – Basic Design Principles – Component-Level Design Guidelines – Cohesion – Coupling – Software Architecture – Importance of Software Architecture – Architectural Descriptions – Architectural Decisions – Architectural Genres – Architectural Styles – A Brief Taxonomy of Architectural Styles – Architectural Patterns – Organization and Refinement – Architectural Considerations – Architectural Decisions – Architectural Design – Representing the System in Context – Defining Archetypes – Refining the Architecture into Components – Describing Instantiations of the System – Architectural Design for Web Apps and Mobile Apps – Assessing Alternative Architectural Designs – Architectural Description Languages – Architectural Reviews – Pattern-based Architecture

Review – Architecture Conformance Checking – Agility and Architecture.

Unit: 4
Software Testing Strategies – A Strategic Approach to Software Testing – Verification and Validation – Organizing for Software Testing – Software Testing Strategy – Criteria for Completion of Testing – Strategic Issues – Test Strategies for Conventional Software – Unit Testing – Integration Testing – Test Strategies for Object-Oriented Software – Unit Testing in the OO Context – Integration Testing in the OO Context – Test Strategies for WebApps – Test Strategies for MobileApps – Validation Testing – Validation-Test Criteria – Configuration Review – Alpha and Beta Testing – System Testing – Recovery Testing – Security Testing – Stress Testing – Performance Testing – Deployment Testing – The Art of Debugging – The Debugging Process – Psychological Considerations – Debugging Strategies – Correcting the Error – White-Box Testing – Basis Path Testing – Flow Graph Notation – Independent Program Paths – Deriving Test Cases – Graph Matrices – Control Structure Testing – Black-Box Testing – GraphBased Testing Methods – Equivalence Partitioning – Boundary Value Analysis – Orthogonal Array Testing – Model-Based Testing – Software Configuration Management – Elements of a Configuration Management System – Baselines – Software Configuration Items – Management of Dependencies and Changes.

Unit: 5
Risk Management – Reactive versus Proactive Risk Strategies – Software Risks – Risk Identification – Assessing Overall Project Risk – Risk Components and Drivers – Projection – Developing a Risk Table – Assessing Risk Impact – Risk Refinement – Risk Mitigation, Monitoring, and Management – The RMMM Plan – Maintenance and Reengineering – Software Maintenance – Software Supportability – Reengineering – Business Process Reengineering – Business Processes – A BPR Model – Software Reengineering – A Software Reengineering Process Model – Software Reengineering Activities – Reverse Engineering – Reverse Engineering to Understand Data – Reverse Engineering to Understand Processing – Reverse Engineering User Interfaces – Restructuring – Code Restructuring – Data Restructuring – Forward Engineering – Forward Engineering for Client-Server Architectures and Object-Oriented Architectures – The Economics of Reengineering.

References:
Á Ian Sommerville, “Software Engineering”, Tenth Edition, Pearson 2016.
Á Ian Sommerville, “Engineering Software Products”, Pearson 2019.

Computer oriented and numerical method (MCA-203)

Learning Outcomes:
- Obtain an intuitive and working understanding of numerical methods for the basic problems of numerical analysis.
- Gain experience in the implementation of numerical methods using a computer.
- Trace error in these methods and need to analyze and predict it.

Unit:1
Computer Arithmetic and Solution of Non-Linear Equations : Introduction – Floating Point Arithmetic and Errors: Floating point represent of Numbers – Sources of Errors – NonAssociativity of Arithmetic – Propagated Errors – Pitfalls in Computation. Solution of NonLinear equations: Bisection – Fixed point – Regula falsi – Newton’s Raphson – Secant method. Convergence criteria of Iterative methods .

Unit 2 :
Solution of simultaneous Linear Algebraic Equations and ordinary differential equations : Cramer's Rule - Gauss elimination method – Pivoting Strategies - Gauss Jordan method – Jacobi Iterative method – Gauss Seidal method –Comparison of Direct and Iterative methods.

Unit 3 :
Interpolation and Curve Fitting : Problem of Interpolation - Langranges method of Interpolation – Inverse Interpolation – Newton's interpolation formulae – Error of the Interpolating Polynomial - Interpolation at equally spaced points.

Unit 4
Forward and Backward differences – Newton's forward and backward difference formulas. Fitting of polynomials and other curve - Least square approximation of functions - linear and polynomial regressions.

Unit 4 :
Numerical differentiation and Integration : Differentiation based on polynomia fit - Numerical integration using Simpson,s rule and Gaussian quadratic formula - Numerical solution of differential equations of the form dy/dx=f(x,y) using Euler,s method and Runge-Kutta methods.

Reference Books :
1. Numerical methods for Scientific and Engineering Computation by M.K.Jain, S.R.K.Iyengar, R.K. Jain.
2. Elementary Numerical Analysis by Samuel D.Conte and Cart de Boor, McGraw Hill International Edition.
3. Numerical methods for Science and Engineering, PHI by R.G.Stanton
4. Computer based numerical algorithms by E.V.Krishnamoorthy
5. Introduction to Numerical Analysis by E.Atkinson

Theory of Computer Science (MCA-204)

Learning Outcomes:
Analyze, design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs. Problem-Solving. Identify problems and formulate solutions for systems and organizations while reconciling conflicting objectives and finding compromises. Communication

Unit 1 :
Set, Relations and Functions : Sets – Notation and description of sets – subsets – operations on sets – Properties of set operations – Relations : Representation of a relation – Operations on Relations – Equivalence Relation Partitions and Equivalence Classes. Functions : Definition – One to one – Onto functions – Special type of functions – Invertible and composition of functions.

Unit 2 :
Logic : Introduction – Connectives – Statements : Atomic – Compound – Well formed – Truth Table – Tautology – Tautological implications and equivalence of formulae – Replacement Process – Normal forms – Principal Normal forms – Theory of Inference – Quantifiers – Theory of Inference for Propositional and predicate calculus.

Unit 3 :
Finite Automata and Languages : Definition – Representation of FA – Languages

Accepted by FA - Non-deterministic Finite Automata – Regular Sets – Phase structure grammar – Context free grammar – Context free language – Finite Automata and regular languages – Turing Machines – Techniques for Turing Machine construction

Unit 4 :
Graph theory : Basic concepts – definition – paths – reach – ability and connectedness – matrix representation of graphs – trees.

Reference Books:
1. Discrete mathematical structures with applications to computer science by J.P.Tremblay and R.Manohar, McGraw Hill.
2. Discrete Mathematics by M.K.Venkatraman, N.Sridharan and N.Chandrasekaran.

Web development tool (MCA-301)

Learning Outcome:
The student will understand, apply, be able to evaluate various techniques for creating large and maintainable "static" websites including Templates and Static Site Generators. The student will learn Server side (back-end) fundamentals and apply them to the creation of dynamic websites and web apps.

Unit 1 :
Programming Concepts: Introduction – Problem solving Stages – Pseudocode – Algorithm – Flowchart – Translators – Machine, Assembly and Procedural Languages – Linkers – Loaders – Elements of a programming language – Graphical User Interface (GUI) – Operating system concepts - Process Management – Multiprogramming – Multitasking – Timesharing – CPU Scheduling – Deadlock avoidance - I/O Device Management – Memory management – Partition – Partition – Page management – Swapping - File Management

Unit 2 :
UNIX Operating System : Foundations of UNIX operating system – Features of UNIX – Structure of UNIX operating system – File System – Different types of files – Command format – Text Manipulation commands – Text Editor – Line editors : ed,ex line editors – Vi Screen editor – Sed – File permissions – Super user, owner and other user categories and their privileges – Communication between users – Super user privileges

Unit 3 :
Programming in Unix : Shell Programming – Command Interpreter – Environment variables – Parameter passing – Shell programming language constructs – operators – Expression evaluation – Support for C programming Code – read, echo, if, case – Loops: do, for loops – System Administration – Adding user accounts – Changing privileges – File system mounting and unmounting – Running background processes

Unit 4 :
Software Engineering : Software Life Cycle – Role of software engineer – Qualities of a software product – Principles of software engineering – Trends in Software Development – 4GL and Natural Languages – System Investigations – Control of System Investigations - Case Tools

Reference:
1. T.W. Pratt – Programming Languages, Design and Implementation – PHI.
2. R.G. Dromey – How to solve it by Computer – PHI.
3. Operating system Design and Implementation by Andrew S. Tanenbaum – PHI
4. Software Engineering, Pressman

Mobile Application Development (MCA-302)

Learning outcomes:

After completion of the Application Programming for Mobile Devices, the student will be able to:

- Describe the requirements for mobile applications
- Explain the challenges in mobile application design and development
- Develop design for mobile applications for specific requirements
- Implement the design using Android SDK
- Implement the design using Objective C and iOS
- Deploy mobile applications in Android and iPone marketplace for distribution

Unit: 1

Introduction to mobile applications – Embedded systems - Market and business drivers for mobile applications – Publishing and delivery of mobile applications – Requirements gathering and validation for mobile applications

Unit 2:

Introduction – Basics of embedded systems design – Embedded OS - Design constraints for mobile applications, both hardware and software related – Architecting mobile applications user interfaces for mobile applications – touch events and gestures – Achieving quality constraints – performance, usability, security, availability and modifiability.

Unit 3:

Designing applications with multimedia and web access capabilities – Integration with GPS and social media networking applications – Accessing applications hosted in a cloud computing environment – Design patterns for mobile applications.

Unit 4:

Introduction – Establishing the development environment – Android architecture – Activities and views – Interacting with UI – Persisting data using SQLite – Packaging and deployment, Interaction with server side applications – Using Google Maps, GPS and Wifi – Integration with social media applications.

Unit 5:

Introduction to Objective C – iOS features – UI implementation – Touch frameworks – Data persistence using Core Data and SQLite – Location aware applications using Core Location and Map Kit – Integrating calendar and address book with social media application – Using Wifi - iPhone marketplace.

Reference:

1. http://developer.android.com/develop/index.html
2. Jeff McWherter and Scott Gowell, "Professional Mobile Application Development", Wrox, 2012
3. Charlie Collins, Michael Galpin and Matthias Kappler, “Android in Practice”, DreamTech, 2012
4. James Dovey and Ash Furrow, “Beginning Objective C”, Apress, 2012
5. David Mark, Jack Nutting, Jeff LaMarche and Frederic Olsson, “Beginning iOS 6 Development: Exploring the iOS SDK”, Apress, 2013.

Object Oriented Technology(MCA-303)

Learning Outcomes:

On successful completion of this program the student will be able to:

- Identify information technology related problems,analyze them and design the system or provide solution to the problem.
- Apply current technical concepts and practices in the core information technologies of human computer interaction, information management, programming, networking and web systems and technologies.
- Use the system Analysis Design Paradigm to critically analyze a problem.

Unit -1 :

C++ Fundamentals : Object Oriented Programming - Programming Paradigms – Benefits and Concepts – Advanced Concepts – OOP languages – Overview of C++ - Structure of a C++ Program – Header Files - Keywords – Tokens and Identifiers – Compiling – Running C++ programs - Constants and Variables: Data Types – Integer – Float – Char – Double – Pointer – Variable and Constant declarations – Macro definitions – Reference variables – Complex variables – Type conversions – Type casting – Storage classes : auto, register, static, extern - Input and Output: Stream I/O – I/O Manipulators – Creating I/O manipulators – IOS flags – Stream buffer class hierarchy

Unit-2 :

Programming Constructs : Operators: Arithmetic – Relational – Logical Assignment – Pre and Post Increment & Decrement – Bitwise – Scope Resolution :: operator – ?(conditional) – Value operator – Member operator – Indirection operator – new and delete operator – Precedence rules – Control structures: if – else – if elseif ladder – switch case - Iterative constructs - Loops – for loop – while loop – do while loop – Initialisation – exit condition – increment/decrement for three loops compared – Nesting loops – Creating infinite loops – break and continue statements – goto statement and labels

Unit-3 :

Data Structures : Arrays: Single Dimensional arrays - Declaration – Initialization – Multi-Dimensional arrays – Declaration – Initialization – Addressing method – Subscripts – Character arrays – Initialization – Null Character – Multidimensional character arrays – Structures: – Declaration – Definition – Bitfields – Array of structures – Structure containing arrays – Pointer to structures – Structures versus unions – Ananymous unions

Unit – 4 :

Structured and Object Oriented Programming : Functions: Structured Programming – Function definition & declaration – Parameters – Arguments – Return Values – void – Call by value parameters – Call by reference parameters – Passing arrays – Passing structures – Passing a function to another function – Pointer to function – Recursive function – Classes : and Objects – Visibility Labels – private, public and protected – Data members – Member functions – Object declaration and accessing members – Passing objects to functions – Returning objects – Constructor function – Destructor function – friend functions – static data and function members – Inline functions versus macros – Overloading: Compile-Time Polymorphism – Function overloading – Rules for function overloading – Operator overloading – rules for operator overloading - Function templates – Class templates – Extensibility – Reusability – Inheritance – Run-Time Polymorphism – Virtual functions – Files: fstream header file - text and binary files creation and access – random access in files – storing objects in files – command-line arguments to main() function – Exception handling – Unified Modeling Language (UML) – Context Diagrams

Reference Books :

1. The C++ programming language, Bjarne Stroustrup, Pearson publications.
2. Object Oriented Programming in C++ by N.Barkakati, PHI.

Cyber Security (MCA-304)

Learning Outcomes:

- To understand necessary Approaches and Techniques to build protection mechanisms in order to secure computer networks.
- To know about the malicious software & firewalls Understand the fundamentals of networks security, security architecture, threats and vulnerabilities
- Apply the various Authentication schemes to simulate different applications.
- Understand various Security practices and System security standards

Unit: 1

Introduction, Computer Security, Threats, Harm, Vulnerabilities, Controls, Authentication, Access Control and Cryptography.Web attack: Browser Attacks, Web Attacks Targeting Users, Obtaining User or Website Data, Email Attacks.
Network Vulnerabilities: Overview of vulnerability scanning, Open Port / Service Identification, Banner /Version Check, Traffic Probe, Vulnerability Probe, Vulnerability Examples, OpenVAS, Metasploit. Networks Vulnerability Scanning (Netcat, Socat), Network Sniffers and Injection tools.

Unit: 2

Firewalls and Packet Filters: Firewall Basics, Packet Filter Vs Firewall, How a Firewall Protects a Network, Packet Characteristic to Filter, Stateless Vs Stateful Firewalls, Network Address Translation (NAT) and Port Forwarding.

Unit: 3

Scanning for web vulnerabilities tools: Nikto, W3af,HTTP utilities - Curl, OpenSSL and Stunnel.Application Inspection tools – Zed Attack Proxy, Sqlmap, DVWA, Webgoat.Password Cracking and Brute-Force Tools: John the Ripper, htcrack, Pwdump, HTC-Hydra.

Unit: 4

Cyber Crimes, Types of Cybercrime, Hacking, Attack vectors, Cyberspace and Criminal Behavior, Clarification of Terms, Traditional Problems Associated with Computer Crime, Introduction to Incident Response, Digital Forensics, Computer Language, Network Language, Realms of the Cyber world

Unit: 5

Internet crime and Act: A Brief History of the Internet, Recognizing and Defining Computer Crime, Contemporary Crimes, Computers as Targets, Contaminants and Destruction of Data, Indian IT ACT Page 3 of 23 2000. • Firewalls and Packet Filters, password Cracking, Keyloggers and Spyware, Virus and Warms, Trojan and backdoors, Steganography, DOS and DDOS attack, SQL injection, Buffer Overflow, Attack on wireless Networks.

Reference Books:

1. William Stallings, "Network Security Essentials Applications and Standards"Third Edition, Pearson Education, 2008.

5. Procedure for Admission, Curriculum Transaction and Evaluation

The proposed program in ODL mode will be conducted by CDOE-SGVU with the support of various

departments of the University. Eligibility criteria, course structure, detailed curriculum, duration of program and evaluation criteria shall be approved by Board of Studies and Academic Council, SGVU, Jaipur which are based on UGC guidelines for the program which comes under the purview of ODL and mode for award of Degree.

Details of Procedure for admission in which eligibility criteria for admission and fee structure of the course, Curriculum includes Program delivery, norms for delivery of courses in ODL mode, use of IT services to academic support services, course design academic calendar and Evaluation which includes Distribution of Marks in Continuous internal assessments, Minimum Passing criteria and system of Grading formats are given in detail as under.

Procedure for Admission

Students who will seek admission in MCA program to apply through its website www.sgvu.edu.in

Minimum Eligibility Criteria for Admission

The minimum eligibility criteria for admission in ODL MCA program is a pass in Bachelor's with computer science from any recognized University.

Program Fee and Financial Assistance Policy

Program fees for students for proposed MCA in various streams offered by CDOE-SGVU Jaipur is Rs. 52,000 Per year tuition fees and 3000 per year examination fees.

Curriculum Transactions

Program Delivery

The curriculum will be delivered through the Self Learning Materials (SLMs) supported by various learning resources including audio-video aids.

Academic Calendar

Sr no	Name of the Activity	Tentative months schedule(specify months) during Year			
		From (Month)	To (Month)	From (Month)	To (Month)
1	Admission	Jul	Sep	Jan	Feb
2	Assignment Submission (if any)	Oct	Nov	April	May
3	Evaluation of Assignment	Nov	Dec	May	June
4	Examination	Dec	Jan	June	Jul
5	Declaration of Result	Feb	Mar	Aug	Sep
6	Re–registration	Jan	Feb	Jul	Sep
7	Distribution of SLM	Jul	Sep	Jan	Feb

8	Contact Program (Counselling, Practical's, etc.)	Nov	Dec	May	June

Evaluation

The evaluation shall include two types of assessments-

1. Continuous Assessment in the form of assignments (30% Weightage)
2. End Semester Examination, which will be held at the SGVU campus (70% Weightage).

Minimum Passing percentage

The students are considered as passed in a course if they score 40% marks in the Continuous Evaluation (Internal Assessment) and end-semester Examinations (External Assessment).

Marks and Grades

Grades & Grade Points

a. At the end of the Semester / Year every student is assigned a 'Letter Grade' based on his/her performance over the semester in all courses for which he/she had registered.

b. The letter grade and grade point indicate the results of quantitative and qualitative assessment of the student's performance in a course.

c. There are seven letter grades: **A+, A, B+, B, C+, C, D, E (E1 for internal back and E2 for external back), F** that have grade points with values distributed on a 10-point scale.

6. Requirement of the Laboratory Support and Library Resources

Library Resources

CDOE-SGVU has excellent library with all the books required for the course learning and reference books for the course of MCA. Adequate online learning links and e-learning materials will also be provided to students which will be support students in their learning cycle.

7. Cost Estimate of the Program and the Provisions

The Estimate of Cost & Budget could be as follows (all figures on Annual basis):

1. Salaries: Rs. 10,00,000/- (Approx)
2. Travel: Rs. 30,000/- (Approx)
3. Seminars: Rs. 40,000/- (Approx)
4. SLM Preparation, Printing, Distribution: Rs. 3,00,000/- (Approx)
5. Library: 1,25,000/- (Approx)
6. Courier/Transportation: Rs. 50,000/- (Approx)
7. Infrastructure: Rs. 1,50,000/- (Approx)

8. Computer Labs & Leased Line: Rs. 1,00,000/- (Approx)

8. Quality assurance mechanism and expected Program Outcomes

The quality of the program depends on the course curriculum and syllabus which meets the requirement of the industry and creates the skillful learning in the students. The ultimate aim of MCA program in ODL Mode is to enhance skills of the learners as managers, entrepreneurs and seeing them excel in their profession and meeting global standards too by upgrading their career opportunities.

The CDOE, SGVU, Jaipur has constituted Centre for Internal Quality Assurance (CIQA) . The CIQA will do periodic assessment of the ODL learning course material and audio video tutorials and will assure that the quality of learning is maintained and time to time changes are made as per the requirement of the course. The CIQA will also access the quality of assignments, quizzes and end term assessment time to time and required changes will be assured by them to maintain the quality of the learning program. CIQA will assure that the learning is made a truly global experience for the learner along with inculcation of required skills in the learner as expected program outcome with CDOE, SGVU, Jaipur.

The university will work continuously for the betterment of processes, assessments, teaching methodology, e-learning material improvisation as per four quadrant approach and implementation of the same as per New Education Policy. The University is committed to deliver the best education in all the learning modes with adherence to NEP, UGC and other regulatory guidelines in truly Global sense. To monitor quality of Student Support Services provided to the learners.