|  |
| --- |
| **SCHEME OF EXAMINATION FOR MASTER OF COMPUTER APPLICATIONS****w. e. f. Academic Session 2016-17(CHOICE BASED CREDIT SYSTEM ( CBCS))** |
| **Paper Code** | **Nomenclature of Paper** | **Credits** | **Exam Time****(hrs.)** | **External Marks** | **Internal Marks** | **Total Marks** |
| **Max** | **Pass** | **Max** | **Pass** |
| **FIRST SEMESTER** |
| MCA-16-11 | Programming with C | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-12 | COMPUTER ORGANIZATIOn & ARCHITECTURE | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-13 | Software Engineering | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-14 | Operating systemS | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-15 | WEB TECHNOLOGIES | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-16 | S/W LAB – I BASED ON MCA-16-11 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-17 | S/W LAB – II BASED ON MCA-16-15 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-18 | SEMINAR | 1 | 1/2 |  |  | 50 | 20 | 50 |
|  | **TOTAL** | **26** |  | **525** |  | **225** |  | **750** |
| **SECOND SEMESTER** |
| MCA-16-21 | SYSTEM PROGRAMMING | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-22 | OBJECT ORIENTED PROGRAMMING USING C++ | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-23 | PRINCIPLES OF PROGRAMMING LANGUAGES | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-24 | Data StructureS | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-25 | Computer Oriented Statistical methods | 4 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-26 | S/W LAB – III BASED ON MCA-16-22 & MCA-16-24 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-27 | S/W LAB – IV BASED ON MCA-16-25 | 2.5 | 3 | 75 | 30 | 25 | 10 | 100 |
| MCA-16-28 | SEMINAR | 1 | 1/2 |  |  | 50 | 20 | 50 |
|  | **TOTAL** | **26** |  | **525** |  | **225** |  | **750** |

**Note 1:** Every student has to earn 2 credits by selecting an open elective paper from other department(s) of

Faculty of Sciences (Physical Sciences) of KUK during second semester (OE-201 to OE-209).

**Note 2:** Total Credits for the First Year will be 54 (26+26+2).

**Note 3:** Sessional Marks in each theory paper will be awarded by the concerned teacher on the basis of marks

obtained in one class test (of 15 Marks and 90 minutes’ duration) and evaluation of assignments (of 10 Marks).

**Note 4:** Sessional Marks in each practical paper will be awarded by the concerned teacher on the basis of

marks obtained in one practical exam (of 15 Marks and 90 minutes’ duration) and viva-voce (of 10 Marks).

**Note 5**: Size of Groups for all practicals should not be more than thirty students.

**MCA-16–11 Programming with C**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours**

**CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide the fundamental and in-depth coverage of programming aspects and implementation in C language.

**Learning Outcomes:**

At the end of this course students should be able to:

* Implement the algorithms in C language.
* Use different constructs of C language to help in implementation.
* Understand the static and dynamic use of memory.

**UNIT – I**

Overview of C: Structure & Memory Layout of C Program; Elements of C, Data types; Storage classes in C: auto, extern, register and static storage class; Header files: Using pre-defined and user-defined header files, Operators: Arithmetic, relational, logical, bitwise, unary, assignment and conditional operators, side effects, precedence & associativity of operators.

**UNIT – II**

Input/Output: Unformatted & formatted I/O function in C, low-level DOS character I/O.

Control statements: Sequencing, Selection: if statement, switch statement; Repetition: for, while, and do-while loop; break, continue, goto statements.

Functions: Definition, prototype, parameters passing techniques, recursion, built-in functions.

**UNIT – III**

Arrays: Definition, types, initialization, processing an array, passing arrays to functions, returning arrays from functions, String handling.

Pointers: Declaration, operations on pointers, pointers and arrays, dynamic memory allocation, pointers and functions, pointers and strings, void pointer.

**UNIT – IV**

Structure & Union: Definition, processing, Structure and pointers, passing structures to functions, use of union.

Data files: Opening and closing a file, I/O operations on files, Error handling during I/O operation, Random access to files, Text file and Binary files. Pre-processor commands and Macro definitions.

**Text Books:**

1. Forouzan Behrouz, “Computer Science: A Structured Programming Approach Using C”, 1st Ed., Cengage Learning

2. Balagurusamy E., “Programming in ANSI C”, 6th Ed., Tata McGraw-Hill.

**Reference Books:**

1. Gottfried, Byron S., “Programming with C”, 2nd Ed., Tata McGraw Hill.

2. Jeri R. Hanly & Elliot P. Koffman, “Problem Solving and Program Design in C”, 7th Ed., Pearson Education.

3. Yashwant Kanetker, “Let us C”, 13th Ed., BPB Publications.

4. Rajaraman, V., “Computer Programming in C”, 3rd Ed., PHI Learning, India.

**MCA-16–12 COMPUTER ORGANIZATIOn & ARCHITECTURE**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**

The objective of this course is to provide concepts of computer architecture and organization, and understand the key skills of constructing cost-effective computer systems. It focusses on design issues in the development of basic computer or components that satisfy design requirements.

**Learning Outcomes**

At the end of this course students should be able to:

* Design combinational and sequential circuits.
* Understand the basic structure and operation of a digital computer.
* Understand the operation of the arithmetic unit including the algorithms & implementation of fixed-point and floating-point addition, subtraction, multiplication & division.
* Know the different ways of communicating with I/O devices and standard I/O interfaces.
* Understand the hierarchical memory system including cache memories and virtual memory.

**UNIT – I**

Digital Logic Fundamentals: Boolean algebra–basic functions, manipulating Boolean functions, K-maps and Quine McCluskey procedures. Combination Logic–multiplexers, decoders, encoders, comparators, adders & subtractors, BCD-to-Seven segment decoder. Basic Sequential Circuits–Flip-flops (RS, JK, T-type and D-Type), Ripple counter, Shift Register.

**UNIT – II**

Basic Computer Organization: Generic computer organization – system bus, instruction cycle, timing diagram of memory read and write operations, CPU organization, memory subsystem organization and interfacing – types of memory, chip organization, memory subsystem configuration, multi-byte data organization, I/O subsystem organization and interfacing, memory subsystem configuration.

Register Transfer Language (RTL): different types of micro-operations, using RTL to specify digital systems – specification of digital components, simple systems, Modulo-6 counter.

**UNIT – III**

CPU Design: design and implementation of simple CPU-fetching, decoding & executing instruction, establishing required data paths, designing hardwired control unit.

Microsequencer Control Unit Design: microsequencer operations, microinstruction formats, design and implementation of a simple microsequencer, reducing number of microinstructions.

Computer Arithmetic: Hardware implementation of unsigned & signed (addition & subtraction, multiplication, booth’s algorithm, division). Floating-point numbers (IEEE 754 standard) – addition, subtraction, multiplication, division.

**UNIT – IV**

Memory Organization: Hierarchical memory system, associative memory, cache memory – associative, direct and set associative mappings, replacing & writing data in cache, cache performance, virtual memory - paging, segmentation, memory protection.

I/O Organization: Asynchronous data transfer - source and destination - initiated, handshaking, programmed I/O, interrupts, DMA, IOP, serial communication–UART, RS-232C standard, USB standard.

**Text Books:**

1. John D. Carpinelli, “Computer Systems Organization & Architecture”, 1st Ed., Pearson Education, 2001.
2. Stallings W., “Computer Organization and Architecture”, 9th Ed., Pearson Education, 2014.

**Reference Books:**

1. Rajaraman, V., Radhakrishanan, T. “An Introduction to Digital Computer Design”, 5th Ed.,

 PHI Learning.

1. Mano, M. Morris “Digital Logic and Computer Design”, 5th Ed., Pearson Education.
2. Tanenbaum A.S., Todd Austin, “Structured Computer Organization”, 6th Ed., PHI Learning.
3. Carl Hamacher, Zvonko Vranesic, Safwat Zaky, “Computer Organization”, 5th Ed., Tata McGraw Hill.

**MCA-16–13 Software Engineering**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives:**

The objective of this course is to educate the students about (i) the different models of software development, (ii) metrics used in software engineering and (iii) different quality standards.

**Learning Outcomes:**

At the end of this course students should be able to:

* Apply the different tools and techniques to the software development.
* Quantitatively evaluate the process and product.

**Unit- I**

Introduction to Software Engineering, System Engineering Vs Software Engineering, Software Evolution, Software Characteristics, Components, Crisis–Problems and Causes, Software Feasibility, Software Process Models – V-Model, Waterfall, Iterative Enhancement, Incremental, RAD, Prototyping, Spiral, Concurrent Development, Rational Unified Process & AGILE. Challenges in Software Engineering. Software Quality Standards: ISO 9001, SEI-CMM, CMMI.

**Unit – II**

Software Project Management – Planning, Scope Management, Cost Estimation – LOC, Function Point Analysis & COCOMO, Putnam Resource Allocation Model, Project Scheduling & Resource Management Gantt-Chart, PERT Chart, Histogram, Critical Path Analysis, Team Building and Organization Charts, Project Monitoring & Risk Management, Software Configuration Management, Software Quality Assurance, Project Monitoring& Techniques.

Software Requirement Analysis - Structured Analysis, Object Oriented Analysis and Data Modelling, Software Requirement Specification, DFDs, Data Dictionaries, Decision Trees, Decision Tables & Structured English, ER Diagrams.

**Unit-III**

Design and Implementation of Software – Basic Fundamentals, Design Methodology (Structured and Object Oriented), Design Approaches, UML & Design Patterns, User Interface Designing Tools & Techniques, Design Complexity, Monitoring and Control, Coding, Halstead’s Software Science, McCabe’s Cyclomatic Complexity

Software Reliability: Metric and Specification, Fault Avoidance and Tolerance, Exception Handling, Defensive Programming, Component Based Development.

**Unit-IV**

Software Testing – Fundamentals, Validation & Verification, White-Box and Black-Box Testing Techniques (Control Flow, Data Flow, Loop, Mutation, Load, Stress, Performance, Boundary Value, Equivalence Class, Decision Table, Cause Effect Graph Testing) Testing Strategies: Unit, Integration, Validation and System Testing, Alpha & Beta Testing, Debugging, Static Testing Strategies.

Software Re-engineering &Maintenance – Re-Engineering & Reverse Engineering, Maintenance Characteristics, Maintainability, Maintenance Tasks and Side Effects. CASE Tools.

**Text Books**:

1. Pressman S. Roger, “Software Engineering”, 7th Ed., Tata McGraw Hill.

2. Jalote Pankaj, “An Integrated Approach to Software Engineering”, 2nd Ed., Narosa Publ. House.

**Reference Books:**

1. K. K. Aggarwal, Yogesh Singh, “Software Engineering”, 3rd Ed., New Age International.

2. Sommerville, “Software Engineering”, 5th edition, Pearson Education.

3. Fairley Richard, “Software Engineering Concepts”, Tata Mc-Graw Hill Ed.

4. Rajib Mall, “Fundamentals of Software Engineering”, 3rd Ed., PHI Learning.

**MCA-16–14 Operating systemS**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this paper is to get the students familiar with different functions performed by operating systems.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the concept of Operating System.
* Understand the concept of process, file, memory and device management.
* Appreciate the need of distributed operating systems and their working.

**UNIT – I**

Introductory Concepts: Operating system functions, structure, types viz. distributed systems, special-purpose systems, open-source operating systems; Operating system services, system calls, system programs.

CPU Scheduling: Process concepts, process operations, inter-process communication, scheduling criteria, scheduling algorithms, Comparative study of scheduling algorithms, Multiple processor scheduling.

**UNIT – II**

Concurrent Processes: Critical section problem, Semaphores, Classical process co-ordination problems and their solutions, monitors, synchronization examples.

Deadlocks: Deadlock characterization, Deadlock prevention and avoidance, Deadlock detection and recovery.

**UNIT – III**

Memory Management: Swapping, Paging, Segmentation, Virtual memory concepts: Demand Paging, Page replacement Algorithms, Thrashing.

Storage Management: File concepts, File access and allocation methods, File-system mounting, sharing, protection, structure and implementation. Directory Systems: Structured Organizations, directory protection mechanisms, recovery. Disk scheduling.

**UNIT – IV**

Protection & Security: Goals & principles of protection, domains of protection, access matrix, access controls. Security: Security problem, threats, security tools, classification.

Distributed Systems: Types of network-based OS, Network structure and topologies, Communication structure & Protocol, design issues. Distributed File-system: Remote file access, File replication. Distributed synchronization: Mutual exclusion, Concurrency control, deadlock handling.

**Text Books:**

1.Silberschatz A., Galvin P. B., Gagne G., “Operating System Concepts”, 10th Ed., Wiley India Pvt. Ltd.

2. Godbole, A.S., “Operating Systems”, Tata McGraw-Hill, 2nd Ed., New Delhi.

3. Tanenbaum, A.S., “Operating System- Design and Implementation”, 3rd Ed., PHI Learning.

**Reference Books:**

1. Deitel, H.M., “Operating Systems”, 3rd Ed., Pearson Education.
2. Stallings William, “Operating System”, 8th Ed., PHI Learning.

**MCA-16–15 WEB TECHNOLOGIES**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide the fundamental and in-depth coverage of technologies used in the design and development of web based applications such as HTML/CSS/JavaScript/PHP for client-side and server-side programming.

**Learning Outcomes:**

At the end of this course students should be able to:

* Design web sites for various requirements.
* Design web applications with CSS.
* Develop web based applications by programming on both sides (client and server).

**UNIT – I**

Introduction to Web Engineering: Categories and Characteristics of Web Applications, Web Applications Vs Conventional Software, Need for an Engineering Approach.

Web Essentials: The Internet, Basic Internet Protocols, WWW, HTTP (Structure of Request and Response Messages), Web Browser and its functions, URL, Web Servers and their features, Defining Virtual Hosts, Secure Servers.

**UNIT – II**

MarkUp Languages: Introduction to HTML, Characteristics, XHTML Syntax and Semantics, Fundamental HTML Elements, Lists, Tables, Frames, Forms, XHTML Abstract Syntax, Creating HTML Pages.

Cascading Style Sheets: Features, Core Syntax, Types, Style Sheets and HTML, Style Rule Cascading and Inheritance, Text Properties, CSS Box Model, Normal Flow Box Layout, Positioning and other useful Style Properties.

**UNIT – III**

Client–Side Programming (JavaScript): Introduction, obtaining user inputs, memory concepts, Operators, Control Structures, looping constructs, break, continue statements, Programmer defined functions, Scoping rules, Recursion and iteration, Array declaration and allocation, passing arrays to function, Objects: String, Date, Boolean, Window, document; using cookies, Handling Events Using JavaScript, data validation using regular expressions.

**UNIT – IV**

Server-Side Programming (PHP): PHP, Installing and Configuring MySQL and PHP, Basic Security Guidelines, Variables, Data Types, Operators and Expressions, Constants, Flow Control Functions; Switching Flow, Loops, Code Blocks and Browser Output, Objects, Strings Processing, Form processing, Connecting to database, cookies and session management, dynamic contents.

**Text Books:**

1. Deitel H.M., Deitel P.J., “Internet & World wide Web: How to program”, 4th Ed., Pearson Education.
2. Kogent Learning, “Web Technologies: HTML, JavaScript, PHP, Java, JSP, XML, AJAX – Black Book”, Wiley India Pvt. Ltd.

**Reference Books:**

1. Jeffrey C. Jackson, “Web Technologies”, 1st Ed., Pearson Education, India.
2. Thomas Powell, “The Complete Reference HTML”, 5th Ed., Tata McGraw Hill, India.
3. William Pardi, “XML in Action”, 1st Ed., IT Professional, New York, USA.

**MCA-16–18 SEMInar**

**Maximum marks: 50 Time: 1/2 hours CREDITS: 1**

**Seminar**

Each student shall individually prepare and submit a seminar report within stipulated time. A panel consisting of two teachers (internal) should evaluate the seminar report and the presentation. Marks should be distributed considering report writing, presentation, technical content, depth of knowledge, brevity and references and their participation in seminar. The time allotted for presentation is 30 minutes.

**MCA-16–21 SYSTEM PROGRAMMING**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide the fundamental and in-depth coverage of system programming concepts like assemblers, linkers, loaders, compiler etc.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basics of stems.
* Understand various kind language processors.
* Understand various compiling and linking steps.

**UNIT – I**

System Software: Definition, Goals of System Software, Program Development and Production Environments, Software Portability, Programs as components, Quick and Dirty Programming, User-Centric and System-Centric view of System Software.

Language Processors:Types of Language Processors, Program Generation, Program Execution, Program Translation and Interpretation, Fundamentals of Language Processing, Symbol Tables.

**UNIT – II**

Assemblers: Elements of Assembly Language Programming, Pass Structure of Assemblers, Design of Two-pass assembler, Intermediate code forms, Program Listing and Error reporting, Organizational and Design issues in assemblers.

Macros and Macro Pre-processors:Macro Definition and Call, Macro expansion, Nested Macro calls, Design of a Macro pre-processor, Processing of Macro definitions, Use of Stack in expansion of macro calls, Design of a macro assembler

**UNIT – III**

Linkers and Loaders: Linking & Relocation, Design of a Linker, Self-Relocating, Dynamic Linking, Linking for program overlays, Loaders, Absolute and Relocating loaders.

Scanning and Parsing:Chomsky hierarchy of formal languages, Ambiguous grammars, Scanning, Parsing: Top-down and Bottom-up Parsing.

**UNIT – IV**

Compilers and Interpreters:Binding and Binding times, Data Structures of compilers, Scoping rules, Memory allocation, Static and dynamic memory allocation and deallocation, Recursion, Compilation of expressions, Postfix notations, Expression trees, Compilation of Control structures, Code Optimization, Local and Global optimization, Overview and benefits of interpretation, Pure and impure interpreters.

**Text books**:

1. Dhamdhere D.M, “System programming”, 2nd Ed., Tata McGraw-Hill.
2. Beck L. Leland, “System Software”, 3rd Ed., Pearson Education.

**Reference Books:**

1. Aho, Sethi, & Ullman, “Compilers Principles, Techniques and Tools”, 2nd Ed., Pearson Education.
2. Donovan J. John, “System Programming”, Tata McGraw Hill.

**MCA-16–22 OBJECT ORIENTED PROGRAMMING USING C++**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this course is to provide the fundamental and in-depth coverage of object oriented programming. It also focuses on implementation of object-oriented paradigms in C++.

**Learning Outcomes:**

At the end of this course students should be able to:

* Make the object oriented scenario for a real life problem.
* Know various kind of object-oriented features.
* Implement the programming aspects in C++.

**UNIT - I**

Introduction: Object-Oriented features of C++, Comparison of C with C++, Class and Objects, Inline functions, Static data members and member functions, Read-Only objects, Pointers, Dynamic memory allocation and deallocation, constructors and destructors, Dynamic objects, array of pointers to object, local and global class, nested and empty class, pre-processor directives, Header files and namespaces. Console I/O: Hierarchy of console stream classes, unformatted and formatted I/O operations, Manipulators.

**UNIT - II**

Compile-time Polymorphism: Operator Overloading-overloading unary and binary arithmetic and relational operators, overloading subscript, insertion, extraction, new and delete operators; function overloading

Friend Function and Friend Class: Friend function, overloading operators by friend function, friend class

Type Conversion: Basic type conversion, conversion between Objects and Basic Types, conversion between objects of different classes.

**UNIT - III**

Inheritance: Base and Derived Classes, Protected Members, Casting Base-Class Pointers to Derived-Class Pointers, Using Member Functions, Overriding Base–Class Members in a Derived Class, Public, Protected and Private Inheritance, Using Constructors and Destructors in derived Classes, Implicit Derived–Class Object to Base-Class Object Conversion, Composition Vs. Inheritance.

Virtual Functions & Derivations: Virtual functions and their needs, Pure virtual function, virtual destructor, virtual derivation, abstract class.

**UNIT - IV**

Generic Programming: Function Templates, Overloading Template Functions, Class Template, Class Templates and Non-Type Parameters.

Exception Handling: Try, Throw, Catch, throwing an Exception, catching an Exception, Re-throwing an Exception.

File Handling: Hierarchy of File Stream classes, Opening and Closing files, File modes, testing for errors, File pointers and their manipulations, ASCII & Binary files, Sequential and Random access files.

**Text Books:**

1. Bjarne Stroustrup, “The C++ Programming Language”, 4th Ed., Pearson Education.
2. Forouzan, Gilberg, “Computer Science: A Structured Programming Approach Using C++”, Cengage Learning.

**Reference Books:**

1. Herbert Shildt, “C++: The Complete Reference”, 4th Ed., Tata McGraw-Hill.
2. Balaguruswami, E., “Object Oriented Programming in C++”, 2nd Ed., Tata McGraw-Hill.
3. Robert Lafore, “Object Oriented Programming in C++”, 4th Ed., Techmedia SAMS.
4. Bhave M.P., Patekar S.A., “Object Oriented Programming with C++”, 2nd Ed., Pearson Education.

**MCA-16–23 PRINCIPLES OF PROGRAMMING LANGUAGES**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this paper is to make the students familiar with different elements of programming languages such as data types/operators/statements/control constructs and their implementation with the understanding that it will help them in becoming a better programmer.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the programming language hierarchy.
* Understand various programming paradigms.
* Understand the semantics of different language constructs.

**UNIT – I**

Preliminaries: History, Impact of Programming Paradigms, Role of Programming Languages, Good Language, Effects of Programming Environment, Translators and virtual architectures, Binding and Binding time, Language Syntax, Analysis of Program, Synthesis of Object program, Formal translation models: BNF Grammars, General parsing, Language translation, Recursive descent parsing.

**UNIT – II**

Formal languages and automata: The Chomsky hierarchy of formal languages, regular grammars, Regular expressions, Finite State Automata, Context-free grammars, Pushdown automata, Ambiguous grammars.

Language Semantics**:** Attribute grammars,Denotational semantics, Program verification and validation, Data objects, variables, constants, data types, declaration, type checking, type casting, type promotion, Enumerators, Composite data types.

**UNIT – III**

Object Orientated concepts**:** Structured data types, Abstract data types, Information hiding, Subprogram concepts, Good program design, Type definitions, Type equivalence, Inheritance, Derived classes, Abstract classes, Polymorphism, Inheritance and software reuse.

Sequence control**:** Implicit and explicit sequence control, Sequence control within arithmetic expressions, sequence control between statements, sequencing with non-arithmetic expressions, Subprogram Sequence control.

**UNIT – IV**

Miscellaneous topics:Parameter passing techniques, Static & Dynamic Scoping, Storage of variables, Static storage, Heap Storage management, Distributed Processing, Exceptions and Exception handlers, Co-routines, Scheduled subprograms, Parallel programming, Processor design, Hardware and Software architectures, Network Programming, Evolution of scripting languages, Applets, XML.

**Text Books:**

1. Pratt T.W., Zelkowitz M.V., Gopal T.V., “Programming Languages Design and Implementation”, 4th Ed., Pearson Education.
2. Sebesta W. Robert, “Concepts of Programming Languages”, 11th Ed., Pearson Education.

**Reference Books:**

1. Appleby Doris & Vande Kopple J. Julius, “Programming Languages-Paradigm and practice”, 2nd Ed., Tata McGraw Hill.
2. Sethi Ravi, “Programming Languages: Concepts & Constructs”, 2nd Ed., Pearson Education
3. Scott M., “Programming Language Pragmatics”, 4th Ed., Elsevier India.

**MCA-16–24 Data StructureS**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this paper is to make the students familiar with the commonly used data structures and understand their applications in real life problems.

**Learning Outcomes:**

At the end of this course students should be able to:

* Design various kind of data structures.
* Design the data structures according to the need of problem in hand.
* Understand various fundamental and advanced data structures.

**UNIT – I**

Introduction to Data Structures: Classification of Data Structures, Complexity of Algorithms, Asymptotic Notations, Abstract Data Types, Arrays, Representation of Arrays in memory, Operations on Array, Strings, Pointers, Sparse Matrices, Applications.

**UNIT – II**

Stacks & Queues: Representation of Stacks, Stack Operations, Applications, Queues, Operations on Queues, Circular Queues, Dequeue, Priority Queues, Applications.

Linked Lists: Introduction, Types, Operations (Insertion, Deletion, Traversal, Searching, Sorting), Applications, Dynamic Memory Management, Implementation of Linked Representations.

**UNIT – III**

Trees: Definition and Basic Terminologies, Representation of Trees, Binary Trees, Types of Tree, Representation of Binary Trees, Binary Tree Traversals, Threaded Binary Trees, Binary Search Trees and Operations, Minimum Spanning Tree, AVL Trees, Heap, m-way Search Trees, B-Trees, B+ Trees, Applications.

**UNIT – IV**

Graphs: Definitions and Basic Terminologies, Representation of Graphs, Graph Traversals, Shortest Path Problem, Applications.

Sorting and Searching: Recursive Binary Search, Types of Sorting, Implementation of Different Sorting Techniques: Selection Sort, Insertion Sort, Merge Sort, Radix Sort.

Hashing & Collision handling.

**Text Books:**

1. G.A.V Pai, “Data Structures and Algorithms”, 2nd Ed., Tata McGraw-Hill.
2. Drozdek, “Data Structure and Algorithms in C++”, 3rd Ed., Cengage Learning.

**Reference Books**:

1. Trembley, J.P. And Sorenson P.G., “An Introduction to Data Structures with Applications”, 2nd Ed.,

 Tata McGraw- Hill.

1. Seymour Lipschutz, “Data Structures”, Revised 1st Ed., Tata McGraw-Hill, Schaum’s Outlines, New

 Delhi.

1. Weiss, “Data Structures and Algorithm Analysis in C++”, 4th Ed., Pearson Education.
2. Goodrich, “Data Structures & Algorithms in C++”, 2nd Ed., Wiley India Pvt. Ltd.

**MCA-16–25 Computer Oriented Statistical methods**

**Maximum marks: 100 (External: 75, Internal: 25) Time: 3 hours CREDITS: 4**

Note: Examiner will be required to set NINE questions in all. Question Number 1 will consist of objective type/short-answer type questions covering the entire syllabus. In addition to question no. 1, the examiner is required to set eight more questions selecting two from each unit. Student will be required to attempt FIVE questions in all. Question Number 1 will be compulsory. In addition to compulsory question, student will have to attempt four more questions selecting one question from each Unit. All questions will carry equal marks.

**Objectives**:

The objective of this paper is to get the students acquainted with the statistical techniques required generally for analysis of the data and implementing those statistical methods in R language.

**Learning Outcomes:**

At the end of this course students should be able to:

* Understand the basics of computer oriented statistics.
* Understand various kind of distributions in statistics.
* Understand the hypothesis and significance of hypothesis.
* Implement the statistics in R language.

**UNIT – I**

R Language: Introduction, Objects, Data types, Operators, workspace, Looping in R, functions and packages, Handling data with R, Graphics with R, Statistical analyses with R: Generic functions, packages; Parametric and non-parametric tests with R.

**UNIT – II**

Statistics: Introduction, Origin & Development, Definition, Importance, Limitations; Frequency distributions: Graphical Representation; measures of central tendency, arithmetic mean, geometric mean, harmonic mean, weighted mean, median, mode; Measures of Dispersion, Skewness, Kurtosis

Probability: Introduction, types, Sample space, Event, Mathematical Notation, Laws of probability, types of events, Bayes Theorem.

**UNIT – III**

Random Variables & it types, Distribution functions & its types, Probability Density function, Mathematical Expectation, Moment Generating Function.

Theoretical Discrete Distributions: Binomial, Poisson, Geometric, Uniform.

Theoretical Continuous Distribution: Normal, Gamma, Beta, Central Limit Theorem, Crammer's Theorem

**UNIT – IV**

Curve fitting & Principles of Least Squares, Correlation: Karl Pearson, Rank; Regression.

Sampling: Introduction, types, parameter and statistic, Standard Error, Tests of Significance, Hypothesis, Critical Region and Level of significance.

Sampling Distributions: Chi-Square Distribution, T Distribution, F Distribution, Z Distribution; ANOVA

**Text Books:**

1. Gupta S. C., Kapoor V. K., “Fundamentals of Mathematical Statistics”, 11th Ed., Sultan Chand & Sons.
2. Gardener M., “Beginning R: The Statistical Programming Language”, 1st Ed., Wiley India Pvt. Ltd.

**Reference Books:**

1. Lander J.P. “R for Everyone”, 1st Ed., Pearson Education.
2. Gupta S. C., Kapoor V. K., “Fundamentals of Applied Statistics”, 11th Ed., Sultan Chand & Sons.
3. S.P.Gupta, “Statistical Methods”, 28th Ed., Sultan Chand & Sons.

**MCA-16–28 SEMInar**

**Maximum marks: 50 Time: 1/2 hours CREDITS: 1**

**Seminar**

Each student shall individually prepare and submit a seminar report within stipulated time. A panel consisting of two teachers (internal) should evaluate the seminar report and the presentation. Marks should be distributed considering report writing, presentation, technical content, depth of knowledge, brevity and references and their participation in seminar. The time allotted for presentation is 30 minutes.