UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Data Communication & Computer Networks

Course Code: MCA204

L-T Scheme: 3-1 **Course Credits: 4**

Introduction:

This course is to provide students with an overview of the concepts and fundamentals of data communication and computer networks. Topics to be covered include: data communication concepts and techniques in a layered network architecture, communications switching and routing, types of communication, network congestion, network topologies, network configuration and management, network model components, layered network models (OSI reference model, TCP/IP networking architecture) and their protocols, various types of networks (LAN, MAN, WAN and Wireless networks) and their protocols. The course is supplemented by a practical component covered in CS692 concurrently.

Objectives:

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

1. Build an understanding of the fundamental concepts of computer networking.
2. Familiarize the student with the basic taxonomy and terminology of the computer networking area.
3. Introduce the student to advanced networking concepts, preparing the student for entry Advanced courses in computer networking.
4. Allow the student to gain expertise in some specific areas of networking such as the design and maintenance of individual networks.

Learning Outcomes:

After completing this course the student must demonstrate the knowledge and ability to:

1. Independently understand basic computer network technology.
2. Understand and explain Data Communications System and its components.
3. Identify the different types of network topologies and protocols.
4. Enumerate the layers of the OSI model and TCP/IP. Explain the function(s) of each layer.
5. Identify the different types of network devices and their functions within a network
6. Understand and building the skills of subnetting and routing mechanisms.
7. Familiarity with the basic protocols of computer networks, and how they can be used to assist in network design and implementation.
8. Analyze the features and operations of various application layer protocols such as Http, DNS, and SMTP.

Application:

1. To configure and implement network topology.
2. To configure and implement local area network.
3. To design network and assign IP address
4. Connect Remote computers
5. Analyze the network.

Course Contents:

Unit-1:

Introduction; Data communications: components, data representation (ASCII,ISO etc.),direction of data flow(simplex, half duplex, full duplex); Networks: distributed processing, network criteria, physical structure (type of connection, topology), categories of network (LAN, MAN,WAN);Internet: brief

history, internet today; Protocols and standards; Reference models: OSI reference model, TCP/IP reference model, their comparative study.

Physical layer:
Overview of data(analog & digital), signal(analog &digital), transmission (analog & digital)& transmission media (guided & non-guided); TDM, FDM, WDM; Circuit switching: time division & space division switch, TDM bus; Telephone network;

Unit-2:
Data link layer:
Types of errors, framing(character and bit stuffing), error detection & correction methods; Flow control; Protocols: Stop & wait ARQ, Go-Back- N ARQ, Selective repeat ARQ, HDLC;

Medium access sub layer:
Point to point protocol, LCP, NCP, FDDI, token bus,token ring; Reservation, polling, concentration; Multiple access protocols: Pure ALOHA, Slotted ALOHA, CSMA, CSMA/CD, FDMA, TDMA, CDMA; Traditional Ethernet, fast Ethernet;

Unit-3:
Network layer:
Internetworking & devices: Repeaters, Hubs, Bridges, Switches, Router, Gateway; Addressing : Internet address, classful address, subnetting; Routing : techniques,static vs. dynamic routing , routing table for classful address; Routing algorithms: shortest path algorithm, flooding, distance vector routing, link state routing; Protocols: ARP, RARP, IP, ICMP, IPV6; Unicast and multicast routing protocols.

Transport layer:
Process to process delivery; UDP; TCP; Congestion control algorithm: Leaky bucket algorithm, Token bucket algorithm, choke packets; Quality of service: techniques to improve QoS.

Unit-4:
Application layer:
DNS; SMTP, SNMP, FTP, HTTP & WWW; Security: Cryptography, user authentication, security protocols in internet, Firewalls.

Modern topics:
ISDN services & ATM ; DSL technology, Cable modem, SONET. Wireless LAN: IEEE 802.11; Introduction to blue-tooth, VLAN's, Cellular telephony & Satellite network.

Text Books:
1. B. A. Forouzan – "Data Communications and Networking (3rd Ed.) " – TMH
2. A. S. Tanenbaum – "Computer Networks (4th Ed.)" – Pearson Education/PHI
3. W. Stallings – "Data and Computer Communications (5th Ed.)" – PHI/ Pearson Education
4. Zheng & Akhtar, Network for Computer Scientists & Engineers, OUP
5. Black, Data & Computer Communication, PHI
6. Miller, data Communication & Network, Vikas

Reference Books:
1. Kurose and Rose – " Computer Networking -A top down approach featuring the internet" – Pearson Education
2. Leon, Garica, Widjaja – "Communication Networks" – TMH

3. Walrand – "Communication Networks" – TMH.
4. Comer – "Internetworking with TCP/IP, vol. 1, 2, 3(4th Ed.)" – Pearson Education/PHI

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Information System Analysis & Design
Course Code: MCA205
L-T Scheme: 3-1 **Course Credits: 4**

Introduction:

The objectives of the course include the enabling of learner to identify the Software projects in an organization after studying various functionalities in the organization. The course covers requirements analysis and design techniques for information systems. Requirements analysis consists of two phases: a feasibility study and a requirements specification phase. For design, the topics covered include architectural design, database design, and user interface design.

Objectives:

The objectives of the course include the enabling of learner to identify the Software projects in an organization after studying various functionalities in the organization. Also, they should be able to structure various requirements, do the design and select the best method to develop the system. They should be able to implement and maintain the system. The learners should also get acquainted with different quality standards as well as learn about Management Information Systems.

Learning Outcomes:

Knowledge:

1. Different type of methodologies, analysis and design techniques.
2. Understand the information systems, feasibility study and a requirements specification phase.
3. Architectural design, database design, and user interface design.
4. Database design, user interface design.
5. Development technique, decision making for system.
6. Testing technique like unit, integration.

Application:

1. To develop, implement methodologies.
2. To develop, implement architectural design.

Course Contents:

Unit 1: **Overview of System analysis and design**

Introduction to System analysis and design. Development life cycle. Requirements determination, Logical design, Physical design, Program design. Risk and feasibility analysis, prototyping.

Unit 2: Information requirement analysis

Process modeling with physical and logical data flow diagrams. Data modeling with entity relationship diagrams. Normalization up to 3NF.

Unit 3: System design

Process descriptions, Input/output controls, Object modeling, Database design, User Interface design, Documentation, Data Dictionary.

Unit 4: Development methodologies

Top down, bottom up, structured chart, Decision table, decision tree, CASE productivity tools.

Unit 5: Testing

Unit, integration testing, System, Acceptance testing, decision tree

Unit 6: Case studies

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Test Case generation Case studies, Use of CASE tools by organizations, Definition of CASE Tools, Use of CASE tools by Organizations, Role of CASE Tools, Advantages of CASE Tools, Disadvantages of CASE Tools, Components of CASE, Types of CASE Tools, Classification of CASE Tools, Reverse and Forward Engineering, Visual and Emerging CASE tools, Traditional systems development and CASE based systems development, CASE environment, Emerging CASE Tools, Objected oriented CASE tools, Creating documentation and reports using CASE tools, Creating

Text Books

1. Elias M. Awad " System Analysis and Design" Galgotia Publications Pvt. Ltd.

2. System Analysis & Design, Parthasarathi, EPH

References

1. Information Systems: Analysis and Design,Ram Bansal 'Vigyacharya',New Age International
2. Analysis, Design & Implementation of Information System, Sharma, VIKAS

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Data Structures with C
Course Code: MCA201
L-T Scheme: 3-1 **Course Credits: 4**

Introduction:

This course examines data structures and algorithms basics. The Topics to be covered (tentatively) include:

- Abstract Data Type and Data Type
- Time and space analysis of algorithms
- Linear Data structures
- Non-linear Data structures
- Sorting, Searching and Hashing

Objectives:

In this course we will study the basic components of data structure and algorithm.Students are expected to be capable of understanding the data structures, their advantages and drawbacks, how to implement them in C, how their drawbacks can be overcome and what the applications are and where they can be used. The way different modules in the operating system interact and work together to provide the basic services of an operating system.

Learning Outcomes:

Knowledge:

1. To learn about the data structures/ methods/algorithms mentioned in the course with a comparative perspective so as to make use of the most appropriate data structure/ method/algorithm in a program to enhance the efficiency (i.e. reduce the run-time) or for better memory utilization, based on the priority of the implementation.
2. To understand at least the efficiency aspects of the graph and sorting algorithms covered in this course.
3. To convert an inefficient program into an efficient one using the knowledge gathered from this course.

Application:

1. To implement different types of linked list.
2. To implement graph algorithm for any network
3. To implement sorting and searching.

Course Contents:

Unit 1: Introduction-Data and data structure, Abstract Data Type and Data Type.Algorithm efficiency and analysis, time and space analysis ofalgorithms – order notations.

Unit 2: Linear Data structures–Array, Linked List, Stack, Queue and Recursion with their types, different operations and applications

Unit 3: Nonlinear Data structures–Graph, Trees, Minimum spanning tree with their types, different operations and applications.

Unit 4: Sorting, Searching and Hashing- Bubble sort and its optimizations, insertion sort, shell sort, selection sort, merge sort, quick sort, heap sort (concept of max heap, application – priority queue), radix sort. Sequential search, binary search, interpolation search. Hashing functions, collision resolution techniques.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Text Books

1. YashavantKanetkar, Abduln A.P.J. Kalam," Data Structure Through C",2nd edition, BPB Publications
2. Seymour Lipschutz, "Data Structures",Revised First edition,McGraw Hill Education.

References

1. Langsam,Augestein, Tenenbaum:Data Structures using Cand C++, 2nd Edn, 2000,
2. Horowitz and Sahani:FundamentalofDataStructuresin C,2nd Edn, 2008
3. Kruse, Tonso, Leung: Data Structures and ProgramDesign in C, 2000
4. Richard F.Gilberg&BehrouzForouzan: Data Structures, APseudocodeApproach withC, 2001.
5. Weiss: DataStructuresand AlgorithmAnalysisin C/C++, 3rd Edn, 2006

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Database Management System-I
Course Code: MCA202
L-T Scheme: 3-1 **Course Credits: 4**

Introduction

Database Management Systems (DBMS) consists of a set of interrelated data and a set of programs to access that data. They underpin any computer system and are therefore fundamental to any program of study in computer science. An understanding of DBMS is crucial in order to appreciate the limitations of data storage and application behavior and to identify why performance problems arise.
Students who complete this course are expected to develop the ability to design, implement and manipulate databases. Students will apply and build databases for various day to day real life scenarios and real life applications. The course will by and large be structured but will introduce open-ended data base problems.

Course Objectives:

- Ability to build normalized databases.
- Knowledge of Entity Relationship Modeling.
- Familiarity with SQL, embedded SQL and PLSQL.
- Familiarity with query processing and query optimization techniques.
- Understanding of transaction processing.
- Ability to handle recovery and concurrency issues.
- Familiarity with ODBC, JDBC.

Outcomes:

- Develop the ability to design, implement and manipulate databases.
- Introduce students to build database management systems.
- Apply DBMS concepts to various examples and real life applications.

Expected Student Background (Preconditions)

- Introduction to any programming language (Preferably, C)
- Data Structures

Course Contents:

Unit 1: Introduction to DBMS- Concept & overview of DBMS, Data Models & database Language,Database Administrator, Database Users, architecture of DBMS, Three levels of abstraction.

Unit 2: Entity Relationship Model – Basic concepts, Design Issues,Entity-RelationshipDiagram,Weak Entity Sets, Extended E-R features.

Relational Model-
Structure of relational Databases, Relational Algebra,Relational Algebra Operations,Views, Modifications of the Database.

Unit 3: SQL and Integrity Constraints:Concept of DDL, DML, DCL,Basic Structure, Set operations, Aggregate Functions,Null Values, Domain Constraints, Referential Integrity Constraints,assertions, views, Nested Sub queries.

Course Description

Unit 4: Relational Database Design: -Functional Dependency, Different anomalies in designing a Database,Normalization using functional dependencies,Decomposition, Boyce-Codd Normal Form, 3NF,Normalization using multi-valued dependencies, 4NF, 5NF.

Unit 5: Transaction: -Transaction concept, transaction model,serializability,transaction isolation level,Transaction atomicity and durability, transaction isolation and atomicity.

Concurrency control and recovery system:
Lock based protocol, dead lock handling, time stamp based and validation based protocol,failure classification, storage, recovery algorithm, recovery and atomicity,backup.

Unit 6: Internals of RDBMS:-Physical data structures, Query optimization: join algorithm,Statistics and cost based optimization.

Unit 7: File Organization & Index Structures:-File & Record Concept, Placing file records on Disk, Fixed and Variable sized Records,Types of Single-Level Index (primary secondary, clustering), Multilevel Indexes,Dynamic Multilevel Indexes using B tree and B+ tree.

Text Books:

1. Silberschatz, Korth and Sudarshan, "Database System Concepts", 6th Edition,McGraw Hill, 2010

2. Elmasri and Navathe, "Fundamentals of Database Systems", 6th Edition, Pearson, Addison-Wesley, 2010

References:

1. C.J. Date, "An Introduction to Database Systems", 8th Edition, Addison-Wesley, 2003

2. Ramakrishnan&Gherke, Database Management Systems, 2nd Edn., McGraw

3. Connolly and Begg, "Database Systems", 4th Edn., Addison-Wesley, 2005

4. Toby, Lightstone and Jagadish, "Database Modeling and Design", 5th Edn, Elsevier, 2011

5. Coronel and Rob, "Database Systems", 9th Edn.,Cengage, 2011
6. IEEE / ACM Transactions on Database Systems (TODS).
7. DBMS related Journals.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Object-Oriented Programming With C++

Course Code: MCA203
L-T-P Scheme: 3-1-0 **Course Credit: 4**

Introduction:
This course provides a comprehensive introduction to understand the underlying principles, techniques and approaches which constitute a coherent body of knowledge in C++.

Objectives:
The course presents basics of C++ programming including: Basics of C++ environment, Data representation, Control structures, Functions, Arrays, Pointers, Strings, and Classes that aims to:

- Understand object oriented programming and able to explain the difference between object oriented programming and procedural programming.
- Be able to program using C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
- Be able to build C++ classes using appropriate encapsulation and design principles.

Learning Outcomes:

- Be able to develop, design and implement simple computer programs.
- Understand functions and parameter passing.know how to achieve polymorphism at compile and run time with the concept of function overloading, operator overloading, virtual function
- Understand object-oriented design and programming and alsobe familiar with the concept of constructor destructor, inheritance
- Understand dynamic memory allocation and pointers.

Course Contents:

Module-I: Introduction to Object-oriented Programming concept
Procedure-oriented Programming, Object-oriented Programming Paradigm; Basic concepts of Object-oriented programming, Benefit of OOPs.

Module-II: Beginning with C++
What is C++? Application of C++,A simple C++ program, An example with class; Structure of C++ program, tokens, keywords, identifiers and constants, data types, reference variables, scope resolution operator.

Module-III: Functions in C++
Main function, function prototyping, call by reference,return by reference, Inline functions and friend functions, virtual function, Concept of Function overloading

Module-IV: Classes and Objects:
Specifying a class, defining member functions; A C++ program with class; Making an Outside Function inline; Static data members; static member functions; arrays of objects; Objects as function arguments.

Module V: Constructors and Destructors:
Constructors, default Constructors; Multiple constructors in a class; parameterized constructor; copy constructor; Destructor

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Module VI: Inheritance:
Defining Derived classes,single inheritance; multilevel inheritance, multiple inheritance; hierarchical, hybrid inheritance; virtual base classes, abstract classes; constructor in derived classes; Making a private member inheritable.

Module VII: Operator overloading:
Defining Operator overloading, rules for overloading operators; Overloading unary operators using member function; Overloading of unary operator with friend function; Overloading Binary operators using member function; Overloading Binary operators using friends, Examples; Type conversion.

Module VIII: Polymorphism:

Concept of polymorphism, runtime polymorphism, compile time polymorphism; Pointers, Pointers to objects; this pointer; Function overloading with an example(Program); Function overriding with a proper example; Virtual function; Pure Virtual function; Abstract class

Module IX: Exception Handling&Templates:
Introduction, Basics of Exception Handling; Exception Handling mechanism; Throwing and catching mechanism; Rethrowing an Exception; Introduction of Template; Class templates,; Function templates.

Text Books:

- Schildt, H., the Complete Reference C++, Tata McGraw Hill Education Pvt. Ltd.
- E.Balagurusamy; Object Oriented programming with C++; Tata McGraw Hill Education Pvt. Ltd.

References:

- Debasish Jana, C++ object oriented programming paradigm, PHI.
- D. Ravichandran, Programming with C++, Tata McGraw Hill Education Pvt. Ltd.
- Y.I. Shah and M.H. Thaker, Programming In C++, ISTE/EXCEL BOOKS.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Data Structure Lab
Course Code: MCA291
L-T-P scheme: 0-0-3 **Course Credit: 2**

Objectives:

1. Develop problem solving ability using Programming.
2. Develop ability to design and analyse algorithms.
3. Introduce students to data abstraction and fundamental data structures.
4. Develop ability to design and evaluate Abstract Data Types and data structures.
5. Apply data structure concepts to various examples and real life applications

Learning Outcomes:

The course will use hands on practice and applying the knowledge gained in theory course to different day to day real world applications.Upon the completion of data structure and algorithm practical course, the student will be able to:

- **Understand** and implement different type of data structure techniques
- **Analyze** the hashing method.
- **Implement** different type os sorting searching techniques.

Course Contents:

Exercises that must be done in this course are listed below:

Exercise No.1: Implementation of array operations
Exercise No. 2: Stacks and Queues: adding, deleting elements
Exercise No. 3: Circular Queue: Adding & deleting elements
Exercise No. 4: Merging Problem: Evaluation of expressions operations on multiple stacks & queues
Exercise No. 5: Implementation of linked lists: inserting, deleting, and inverting a linked list.
Exercise No. 6: Implementation of stacks & queues using linked lists, Polynomial addition, and Polynomial multiplication
Exercise No. 7: Sparse Matrices: Multiplication, addition.
Exercise No. 8: Recursive and Non-recursive traversal of Trees
Exercise No. 9: Threaded binary tree traversal. AVL tree implementation
Exercise No. 10: Application of Trees. Application of sorting and searching algorithms

Text Book:

1. YashavantKanetkar, Abduln A.P.J. Kalam," Data Structure Through C",2nd edition, BPB Publications
2. Seymour Lipschutz, "Data Structures",Revised First edition,McGraw Hill Education.

Recommended Systems/Software Requirements:

1. Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.
2. Turbo C or TC3 complier in Windows XP or Linux Operating System.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Database Lab
Course Code: MCA292
L-T-P Scheme: 0-0-3 **Course Credits: 2**

Objective:

At the end of the semester, the students should have clearly understood and implemented the following:

1. Stating a database design problem.
2. Preparing ER diagram
3. Finding the data fields to be used in the database.
4. Selecting fields for keys.
5. Normalizing the database including analysis of functional dependencies.
6. Installing and configuring the database server and the front end tools.
7. Designing database and writing applications for manipulation of data for a stand alone and shared database including concepts like concurrency control, transaction roll back, logging, report generation etc.
8. Get acquainted with SQL. In order to achieve the above objectives, it is expected that each students will chose one problem. The implementation shall being with the statement of the objectives to be achieved, preparing ER diagram, designing of database, normalization and finally manipulation of the database including generation of reports, views etc. The problem may first be implemented for a standalone system to be used by a single user. All the above steps may then be followed for development of a database application to be used by multiple users in a client server environment with access control. The application shall NOT use web techniques. One exercise may be assigned on creation of table, manipulation of data and report generation using SQL.

Learning Outcomes:

- Ability to build normalized databases.
- Knowledge of Entity Relationship Modelling.
- Familiarity with SQL, embedded SQL and PLSQL.
- Familiarity with query processing and query optimization techniques.
- Understanding of transaction processing.
- Ability to handle recovery and concurrency issues.
- Familiarity with ODBC, JDBC.

Course Contents:

Exercises that must be done in this course are listed below:

Exercise No.1:ER Model: An entity-relationship model (ERM) is an abstract and conceptual representation of data. Entity-relationship modeling is a database modeling method, used to produce a type of conceptual schema or semantic data model of a system

Exercise No. 2:EER Model: In computer science, the enhanced entity-relationship (EER) model is a high-level or conceptual data model incorporating extensions to the original entity-relationship (ER) model, used in the design of databases. It was developed by a need to reflect more precisely properties and constraints that are found in more complex databases.

Exercise No. 3: Relational Model: The relational model for database management is a database model based on first-order 4predicate logic, first formulated and proposed in 1969 by E.F. Codd. The model uses the concept of a mathematical relation, which looks somewhat like a table of values - as its basic building block, and has its theoretical basis in set theory and first-order predicate logic.

Exercise No. 4:1 NF: First normal form (1NF or Minimal Form) is a normal form used in database normalization. A relational database table that adheres to 1NF is one that meets a certain minimum set

of criteria. These criteria are basically concerned with ensuring that the table is a faithful representation of a relation and that it is free of repeating groups.

Exercise No. 5:2 NF: Second normal form (2NF) is a normal form used in database normalization. 2NF was originally defined by E.F. Codd in 1971. A table that is in first normal form (1NF) must

Exercise No. 6:3 NF: The Third normal form (3NF) is an important form of database normalization. 3NF is said to hold if and only if both of the following conditions hold:

• The relation R (table) is in second normal form (2NF)

• Every non-prime attribute of R is non-transitively dependent (i.e. directly dependent) on every candidate key of R.

Exercise No. 7:BCNF: A relation R is in Boyce-Codd normal form (BCNF) if and only if every determinant is a candidate key. 4The definition of BCNF addresses certain (rather unlikely) situations which 3NF does not handle.

Exercise No. 8:SQL-1: In this lab., we discuss basic SQL operations like creating a table, deleting a table, changing the schema of the table, primary key and foreign key constraints on a table and creating indexes on tables.

Exercise No. 9:SQL-2: Its scope includes efficient data insert, query, update and delete, schema creation and modification, and data access control. In this lab., we discuss SQL operations for populating the tables like inserting into a table, deleting values from a table, and updating the content of the tables.

References

1. “Database Systems: A Practical Approach to design, Implementation and Management”. Thomas Connolly, Carolyn Begg; Third Edition, Pearson Education.
2. "Fundamentals of Database Systems" Elmasri, Navathe, Pearson Education.
3. Bipin C Desai, ?An Introduction to Database Systems?, Galgotia. Publications Pvt Limited, 2001
4. “An Introduction to Database Systems”, C.J.Date, Pearson Education.
5. “A first course in Database Systems”, Jeffrey D. Ullman, Jennifer Windon, Pearson, Education.
6. “Data Management: databases and organization”, Richard T. Watson, Wiley.
7. “DataModeling Essentials”, Graeme C. Simxion, Dreamtech.
8. Introduction to Data Base Management, Naveen Prakash, Tata McGraw Hill
9. “Oracle 10g manuals”.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description

Title of Course: Object-Oriented Programming lab (C++)
Course Code: MCA293
L-T-P Scheme: 0-0-3 **Course Credit: 2**

Objectives:

The course presents basics of C++ programming including: Basics of C++ environment, Data representation, Control structures, Functions, Arrays, Pointers, Strings, and Classes that aims to:

- Understand object oriented programming and able to explain the difference between object oriented programming and procedural programming.
- Be able to program using more advanced C++ features such as composition of objects, operator overloads, dynamic memory allocation, inheritance and polymorphism, file I/O, exception handling, etc.
- Be able to build C++ classes using appropriate encapsulation and design principles.

Learning Outcome:

- Be able to develop, design and implement simple computer programs.
- Understand functions and parameter passing.
- Be able to do numeric (algebraic) and string-based computation.
- Understand object-oriented design and programming.
- Understand dynamic memory allocation and pointers.
- Be able to design, implement, and test relatively large C++ programs.

Lab Content:

Exercises that must be done in this course are listed below:

Experiment1:-Write a C++ programme to take 10 integer data from the user and find out the maximum minimum from that data.

Experiment 2:-Write a c++ program to generate the Fibonacci series by using class.

Experiment 3:-Write a program to calculate 1+x+x*x+x*x*x.....using loop.

Experiment 4:-Write a program in c++ to find the reverse of a number.

Experiment 5:-A shop required to store information about each item. Information will be item code, price and available quantities. User (sales person) will store information about each item and can display information about each item. Model the above problem with OOP.

Experiment6:-A cricket organization need to store information like name, number of innings, number of not out innings, total run scored and total wicket taken of each cricketer. After storing data, organization will analyze the data and want to come on the following conclusion: If a cricketer plays more or equal inning and is batting average is more than 35 then recognize him as a “BATSMAN”. if a cricketer plays more or equal to 50 innings’ and if taken more than 49 wickets then recognize him as a”BOWLER”.if one satisfies both condition then he will be “ALL ROUNDER”. Organization needs to display each information about each cricketer. Model above problem using OOPs.

Course Description

Experiment 7:-Create a class will two private integer data member, initialize them with constructor. Now display data members with the help of function which is not a member of that class.

Experiment 8:-Write a c++program to display the concept of function with default argument.

Experiment 9:-Create class 'fun' with one private float data member. initialize that data member with constructor. simillarly create another class magic with private data member. Initialize that data member with constructor. now using friend function check data member of which class is greater.

Experiment 10:-Create a class test with one private float data member initialize that data member with constructor similarly create another class testing with one private data member. Initialize that data member with constructor. Now using function swap the value of data member of the classes**.**

Experiment 11:-Write a C++ program to demonstrate the concept of single inheritance.

Experiment 12:-Write a C++ program to demonstrate the concept of multiple inheritance.

Experiment 13:-Write a C++ program to demonstrate the concept of MULTILEVEL inheritance.

Experiment 14:-Write a C++ program to demonstrate the concept of HYBRID inheritance.

Experiment15:-An application needs to swap two integer and two float values using functions. Approach the above problem using functions with same name.

Experiment16:-Write a program to calculate the number of objects created by your program.

Experiment 17:-Write a C++ program to achieve the following thing. A class contains 3 data member of type integer. Use ++ and –- operator in a way so that whenever we use ++ with the object of, all data member will incremented by one. Similarly, -- will work.

Experiment18:-Write a program to add two complex number using operators overloading.

Experiment19:-Write a C++ program to demonstrate the concept of Virtual Class.

Experiment 20:-Write a C++ program to show how Run Time Polymorphism is achieved in C++.

Text Books:

- Schildt, H., The Complete Reference C++, Tata McGraw Hill Education Pvt. Ltd.
- E.Balagurusamy; Object Oriented programming with C++; Tata McGraw Hill Education Pvt. Ltd.

Reference Books:

- Debasish Jana, C++ object oriented programming paradigm, PHI.
- D. Ravichandran, Programming with C++, Tata McGraw Hill Education Pvt. Ltd.
- Y.I. Shah and M.H. Thaker, Programming In C++, ISTE/EXCEL BOOKS.

Recommended Systems/Software Requirements:

1. Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.
2. Turbo C++ compiler in Windows XP/7 or Linux Operating System.

UNIVERSITY OF ENGINEERING & MANAGEMENT, JAIPUR

Course Description