**LESSON PLAN**

**Name of the Faculty : Mrs. Asha Kumari**

**Discipline : Computer Science & Engineering**

**Semester : 4th**

**Subject : Object Oriented Programming**

Lesson Plan Duration : 15 weeks (from January 2018 to April 2018)

\*\*Work Load(Lecture/Practical) per week(in hours)::Lectures:3, Practical-03

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Week** | **Theory** | | **Practical** | |
| **Lecture Day** | **Topic(including assignment/test)** | **Practical Day** | **Topic** |
| 1st | 1st | Unit-1; Introduction to C++, C++ Standard Library | 1st | **1**. Write a function called power ( ) that takes a double value for n and an int value for p, and returns the result as double value. Use a default argument of 2 for p, so that if this argument is omitted, the number will be squared. Write a main ( ) function that gets values from the user to test this function. |
| 2nd | Illustrative Simple C++ Programs, Header Files, Namespaces |
| 3rd | Application of object oriented programming. |
| 2nd | 4th | Object Oriented Concepts Introduction to Objects and Object Oriented Programming | 2nd | 2.A point on the two dimensional plane can be represented by two numbers: an X coordinate and a Y coordinate. Write a program that uses a structure called point to model a point. Define three points, and have the user input values to two of them. Then set the third point equal to the sum of the other two, and display the value of the new point. |
| 5th | Abstract Classes |
| 6th | Accessifier (public/ protected/ private), Class Scope and Accessing Class Members |
| 3rd | 7th | Controlling Access Function, Constant | 3rd | Create the equivalent of a four function calculator. |
| 8th | Class Member, Structure and Class |
| 9th | Test of Unit-1 |
| 4th | 10th | Unit-2:Friend Function and Friend Classes | 4th | A phone number, such as (212) 767-8900, can be thought of as having three parts: the area code (212), the exchange (767) and the number (8900). Write a program that uses a structure to store these three parts of a phone number separately. Call the structure phone. Create two structure variables of type phone. Initialize one, and have the user input a number for the other one. Then display both numbers. |
| 11th | This Pointer |
| 12th | Dynamic Memory Allocation and Deallocation (New and Delete) |
| 5th | 13th | Static Class Members | 5th | Create two classes DM and DB which store the value of distances. DM stores distances in metres and centimeters and DB in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB. Use a friend function to carry out the addition operation. |
| 14th | Constructors |
| 15th | Parameter Constructors and Copy Constructors |
| 6th | 16th | Deconstructors | 6th | Create a class rational which represents a numerical value by two double values- NUMERATOR & DENOMINATOR. Include the following public member Functions:  • constructor with no arguments (default).  • constructor with two arguments.  • void reduce( ) that reduces the rational number by eliminating the highest common factor between the numerator and denominator.  • Overload + operator to add two rational number.  • Overload >> operator to enable input through cin.  • Overload << operator to enable output through cout.  Write a main ( ) to test all the functions in the class. |
| 17th | Introduction of inheritance, Types of Inheritance |
| 18th | Overriding Base Class Members in a Derived Class |
| 7th | 19th | Public, Protected and Private Inheritance | 7th | Consider the following class definition  class father {  protected : int age;  public;  father (int x) {age = x;}virtual void iam ( )  { cout < < “I AM THE FATHER, my age is : ”<< age<< end1:}  };  Derive the two classes son and daughter from the above class and for each, define iam ( ) to write out similar but appropriate messages. |
| 20th | Effect of Constructors and Deconstructors of Base Class in Derived Classes. |
| 21st | Test of Unit-2 |
| 8th | 22nd | Unit-3:Polymorphism ,Pointer to Derived class | 8th | Write a program that creates a binary file by reading the data for the students from the terminal. |
| 23rd | Virtual Functions |
| 24th | Pure Virtual Function |
| 9th | 25th | Abstract Base Classes | 9th | A hospital wants to create a database regarding its indoor patients. The information to store include  a) Name of the patient  b) Date of admission  c) Disease  d) Date of discharge  Create a structure to store the date (year, month and date as its members). Create a base class to store the above information. The member function should include functions to enter information and display a list of all the patients in the database. Create a derived class to store the age of the patients. List the information about all the to store the age of the patients. List the information about all the pediatric patients (less than twelve years in age). |
| 26th | Static and Dynamic Binding |
| 27th | Virtual Deconstructors |
| 10th | 28th | Fundamentals of Operator Overloading | 10th | Make a class **Employee** with a name and salary. Make a class **Manager** inherit from **Employee**. Add an instance variable, named department, of type string. Supply a method to **to String** that prints the manager’s name, department and salary. Make a class **Executive** inherits from **Manager**. Supply a method **to String** that prints the string **“Executive”** followed by the information stored in the **Manager** superclass object. Supply a test program that tests these classes and methods |
| 29th | Rules for Operators Overloading |
| 30th | Implementation of Operator Overloading Like <<,>> |
| 11th | 31st | Implementation of Operator Overloading of Unary Operators | 11th | Imagine a tollbooth with a class called toll Booth. The two data items are a type unsigned int to hold the total number of cars, and a type double to hold the total amount of money collected. A constructor initializes both these to 0. A member function called payingCar ( ) increments the car total and adds 0.50 to the cash total. Another function, called nopayCar ( ), increments the car total but adds nothing to the cash total. Finally, a member function called displays the two totals. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a nonpaying car. Pushing the ESC kay should cause the program to print out the total cars and total cash and then exit. |
| 32nd | Implementation of Operator Overloading of Binary Operators |
| 33th | Test of Unit-3 |
| 12th | 34th | Unit-4Text Streams and binary stream Sequential and Random Access File | 12th | Write a function called reversit ( ) that reverses a string (an array of char). |
| 35th | Stream Input/ Output Classes |
| 36th | Stream Manipulators |
| 13th | 37th | Basics of C++ Exception Handling | 13th | Create some objects of the string class, and put them in a Deque-some at the head of the Deque and some at the tail. Display the contents of the Deque using the forEach ( ) function and a user written display function. Then search the Deque for a particular string, using the first That ( ) function and display any strings that match. Finally remove all the items from the Deque using the getLeft ( ) function and display each item. Notice the order in which the items are displayed: Using getLeft ( ), those inserted on the left (head) of the Deque are removed in “last in first out” order while those put on the right side are removed in “first in first out” order. The opposite would be true if getRight ( ) were used. |
| 38th | Try, Throw, Catch, multiple catch, Re-throwing an Exception |
| 39th | Exception specifications |
| 14th | 40th | Templates: Function Templates | 14th | Create a class account that stores customer name, account number and type of account. From this derive the classes  cur\_acct and sav\_acct to make them more specific to their requirements. Include necessary member functions in order to achieve the following tasks:  a) Accept deposit from a customer and update the balance.  b) Display the balance.  c) Compute and deposit interest.  d) Permit withdrawal and update the balance.  e) Check for the minimum balance, impose penalty, necessary and update the balance.  f) Do not use any constructors. Use member functions to initialize the class members. |
| 41th | Overloading Template Functions |
| 42nd | Class Template |
| 15th | 43rd | Class Templates | 15th | Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get\_data( ) to initialize baseclass data members and another member function display\_area( ) to compute and display the area of figures. Make display\_area ( ) as a virtual function and redefine this function in the derived classes to suit their requirements. Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively and display the area. |
| 44th | Non- Type Template arguments. |
| 45th | Test of Unit-4 |