Title : C++ programming
Faculty: Mr. Vishwa Kiran
Venue:
Duration: 3 Days

/**********Day1**********/

1. write a program to check the sizeof an empty structure in c and c++.
2. write a program to show the difference between **structure in c and** c++
    1. define both functions and data 2. provide access specifier
3. write a program to show the difference between **structure and class** in c++
4. write a program to create the following type of objects
    1. local 2. global 3. array of objects
5. write a c++ program to demonstrate the encapsulation feature.
6. Write a program to define plot class. For this program separate the interface and implementation part(i.e plot.h, plot.cpp, app.cpp). Use scope resolution operator
7. write a program create the following library by using c++(use encapsulation feature)
    1. stack
8. Write a program to define a class to store the information of an employee(emp id, name, designation, salary , phone number ..). Create an array of 5 objects. Initialize the data members through member function(or method). Ask the user enter the employee id and display the information of the employee.
9. Write a program to define a class
    1. with and with out a static data member 2. check the size of the object created for the above classes
10. Write a program to demonstrate the use of following member function with an appropriate examples
    1. static member function 2. const member function 3 friend function
11. Write a program to define a function, which receives
    1. objects as a parameter and invoke the methods through object
    2. address of an object and invoke the member function through pointer
12. Write a program to demonstrate that C++ is strongly typed (strict prototyped) programming language.
13. Write a program to overload
        add, sub, mul and div functions
14. Write a program define an employee class
    1. overload methods which is used to initialize the data members
15. Write a program to define an employee class. Define the method which takes default values if user invokes the method with out passing any parameters.

/********** Day2 **********/

1. Write a program to overload the following arithmetic operator for plot class
    1. +, -, *, /
    2. overload the above operator as non member and methods
2. Write a program program to overload the following relational operator for plot class
    1. <, >, < =, > =program , = =, !=
3. Check the commutative property when one of the operand is constant (p1 + 10 and 10 +p1)
4. overload unary operator for plot class
    1. pre increment, post increment, pre decrement, post decrement
    2. overload the above operator as methods and non member.
5. Write a program to differentiate
    1. malloc and new 2. free and delete
6. write a program to demonstrate the scenario of memory leakage
7. write a program to dynamically allocate memory for array of 10 user defined objects and release the memory (use new and delete).

Write a program to demonstrate the usage of “explicit” key word.

8. Write a program to demonstrate the concept of “Has a” and “Is a” relationship.
9. Write a program to demonstrate that inheritance and containership object code level re usability support object code level re usability
10. Write a program to derive a class, create an object of derived class.
    1. Check the size of a derived object
    2. check the order in which constructors are called
    3. define two functions with the same name in both base and derived class. Check which function is called when it is invoked through derived object

11. Derive a class, when an object of derived class is created it should call
    1. zero argument constructor of derived and one argument constructor of base class
    2. zero argument constructor of derived and two argument constructor of base class
    3. one argument constructor of derived and one argument constructor of base class
    4. one argument constructor of derived and two argument constructor of base class
    5. two argument constructor of derived and one argument constructor of base class
    6. one argument constructor of derived and two argument constructor of base class
12. Write a program to check the following purposes of inheritance
    1. using the existing feature
    2. to add the new feature
    3. to override the old feature
    4. to combine old and new feature
13. Write a program to demonstrate the use of
    1. public inheritance
    2. protected inheritance
    3. private inheritance
14. Write a program to demonstrate the single level inheritance
15. Write a program to demonstrate the multi level inheritance
    1. check the order in which constructor getting invoked
    2. check the size of the derived object
    3. check the order in which method are invoked (if function with a same name is defined in multiple classes)
    4.
16. write a program to demonstrate the multiple level inheritance
    1. check the order in which constructors are invoked
    2. check what happens if both the parent class have the methods with the same name (check with out invoking and invoking the methods)
    3. when an object of derived class is created it should call
        1. zero argument constructor of derived class, one argument of base1 and one argument of base2
        2. one argument constructor of derived class, one argument of base1 and one argument of base2
        3. two argument constructor of derived class, two argument of base1 and two argument of base2

/********** Day 3 **********/

1. write a program to demonstrate the need for virtual functions
2. Write a program to demonstrate that
    1. error code returned can be ignored and exception cannot be ignored
    2. exceptions can skip the levels of the call stack
3. Write a program to throw following type exceptions
    1. int
    2. float
    3. double
4. write a program to demonstrate that exceptions can pass as much information as you want from the code that finds the error to the code that handles it.
5. Write a program to handle the following types of error by throwing an exception
    1. divide by zero
    2. out of bound access
    3. unable to open a file
6. write a program to catch any type of exceptions (i.e int or float or double)
7. Write a program that support generic functions for adding two numbers
8. write a program to support generic class for a stack application