**Logo

Description automatically generated San Francisco Bay University**

**CS360L - Programming in C and C++ Lab**

**Lab Assignment #3**

**Due day: 6/7/2022**

**Instruction:**

1. **Push the answer sheets/source code to Github**
2. **Please follow the code style rule like programs on handout.**
3. **Overdue lab assignment submission can’t be accepted.**

**4. Take academic honesty and integrity seriously (Zero Tolerance of Cheating & Plagiarism)**

1. Analyze the following program and explain each statement and commented-down statements in red. Finally, run the program and type in appropriate inputs from standard input device to show the running results

*#include <stdio.h>*

*#include <iostream>*

*using namespace std;*

*class A {*

*public:*

*A(); //*

*A(int); //*

*A(const A&); //*

*//*

*~A(); //*

*public:*

*void operator=(const A& rhs); //*

*void Print(); //*

*void PrintC() const; //*

*//*

*int x; //*

*public:*

*//*

*int& X() { return x; }*

*};*

*A::A()*

*: x(0)*

*{*

*cout << "Hello from A::A() Default constructor" << endl;*

*}*

*A::A(int i)*

*: x(i)*

*{*

*cout << "Hello from A::A(int) constructor" << endl;*

*}*

*A::A(const A& a)*

*: x(a.x)*

*{*

*cout << "Hello from A::A(const A&) constructor" << endl;*

*}*

*A::~A()*

*{*

*cout << "Hello from A::A destructor" << endl;*

*}*

*void A::operator=(const A& rhs)*

*{*

*x = rhs.x;*

*cout << "Hello from A::operator=" << endl;*

*}*

*void A::Print()*

*{*

*cout << "A::Print(), x " << x << endl;*

*}*

*void A::PrintC() const*

*{*

*cout << "A::PrintC(), x " << x << endl;*

*}*

*void PassAByValue(A a)*

*{*

*cout << "PassAByValue, a.x " << a.x << endl;*

*a.x++; //*

*a.Print();*

*a.PrintC();*

*}*

*void PassAByReference(A& a)*

*{*

*cout << "PassAByReference, a.x " << a.x << endl;*

*a.x++; //*

*a.Print();*

*a.PrintC();*

*}*

*void PassAByConstReference(const A& a)*

*{*

*cout << "PassAByReference, a.x " << a.x << endl;*

1. *PrintC(); //*

*//a.Print(); // Call to "non-const" print function fails!*

*// Compiler error from above line. Why?*

*}*

*void PassAByPointer(A\* a)*

*{*

*cout << "PassAByPointer, a->x " << a->x << endl;*

*a->x++;*

*a->Print();*

*a->PrintC();*

*}*

*int main()*

*{*

*cout << "Creating a0"; getchar();*

*A a0; //*

*cout << "Creating a1"; getchar();*

*A a1(1); //*

*cout << "Creating a2"; getchar();*

*A a2(a0); //*

*cout << "Creating a3"; getchar();*

*A a3 = a0; //*

*cout << "Assigning a3 = a1"; getchar();*

*a3 = a1; //*

*// Call some of the "A" subroutines*

*cout << "PassAByValue(a1)"; getchar();*

*PassAByValue(a1); //*

*cout << "After PassAByValue(a1)" << endl;*

*a1.Print();*

*cout << "PassAByReference(a1)"; getchar();*

*PassAByReference(a1); //*

*cout << "After PassAByReference(a1)" << endl;*

*a1.Print();*

*cout << "PassAByConst(a1)"; getchar();*

*PassAByConstReference(a1); //*

*cout << "After PassAByConstReference(a1)" << endl;*

*a1.Print();*

*cout << "PassAByPointer(&a1)"; getchar();*

*PassAByPointer(&a1); //*

*cout << "After PassAByPointer(a1)" << endl;*

*a1.Print();*

*//*

*cout << "a1.X() = 10"; getchar();*

*a1.X() = 10;*

*a1.Print();*

*cout << "PassAByConstReference"; getchar();*

*PassAByConstReference(20);*

*// Why does the above compile? What does it do?*

*return 0;*

*}*

1. Write the program based on the following requirements
   1. Define a class called *student* that has the following data members:
      1. *int* student number
      2. *string* student name
      3. *double* student average
   2. The following member functions:
      1. Constructor that initialize the data members with default values.
      2. *set* and *get* functions for each data member
      3. *Print* function to print the values of data members.
   3. Define a class called *graduatestudent* that inherits data members and

functions from the class *student*, and then declare the following data members :

* + 1. *int level*
    2. *int year*
  1. Member functions:
     1. constructor
     2. *set* and *get* functions for each data member
     3. Print function.
  2. Define a class called *master* that inherits data members and functions from

*graduatestudent* class, and then declare the following data member:

* + 1. *int newid*
  1. Member function:
     1. constructor
     2. *set* and *get* function for the data member
     3. *Print* function
  2. Write a driver program(i.e. main function) that:
     1. Declare object of type student with suitable values then print it
     2. Declare object of type master with your information then print it.

1. Answer the questions after going through the following class:

*class Seminar{*

*int time;*

*public:*

*Seminar() //Function 1*

*{*

*time = 30;*

*cout << "Seminar starts now" << endl;*

*}*

*void lecture() //Function 2*

*{*

*cout << "Lectures in the seminar on" << endl;*

*}*

*Seminar(int duration) //Function 3*

*{*

*time = duration;*

*cout << "Seminar starts now" << endl;*

*}*

*~Seminar() //Function 4*

*{*

*cout << "Thanks" << endl;*

*}*

*};*

* 1. Write statements in C++ that would execute *Function 1* and *Function 3* of class Seminar.
  2. In Object Oriented Programming, what is *Function 4* referred as and when does it get invoked/called?
  3. In Object Oriented Programming, which concept is illustrated by *Function 1* and *Function 3* together?

1. Answer the questions after going through the following class:

*class Test{*

*char paper[20];*

*int marks;*

*public:*

*Test () // Function 1*

*{*

*strcpy (paper, "Computer");*

*marks = 0;*

*}*

*Test (char p[]) // Function 2*

*{*

*strcpy(paper, p);*

*marks = 0;*

*}*

*Test (int m) // Function 3*

*{*

*strcpy(paper,"Computer");*

*marks = m;*

*}*

*Test (char p[], int m) // Function 4*

*{*

*strcpy (paper, p);*

*marks = m;*

*}*

*};*

* 1. Write statements in C++ that would execute Fu*nction 1, Function 2, Function 3* and *Function 4* of class *Test*.
  2. Which feature of Object Oriented Programming is demonstrated using *Function 1, Function 2, Function 3* and *Function 4* together in the above class *Test*?

1. Consider the definition of the following class:

*class Sample{*

*private:*

*int x;*

*double y;*

*public :*

*Sample(); //Constructor 1*

*Sample(int); //Constructor 2*

*Sample(int, int); //Constructor 3*

*Sample(int, double); //Constructor 4*

*};*

* 1. Write the definition of the *constructor 1* so that the private member variables are initialized to *0*
  2. Write the definition of the *constructor 2* so that the private member variable *x* is initialized according to the value of the parameter, and the private member variable *y* is initialized to *0*
  3. Write the definition of the *constructors 3* and *4* so that the private  
     member variables are initialized according to the values of the parameters.