**Assignment**

1.Answer the questions (i) and (iii) 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;

}

};

i. Write statements in C++ that would execute Function 1 and Function 3 of class Seminar.

Seminar s1; // Execute funtion 1

Seminar s2(20); //Execute function 3

ii. In Object Oriented Programming, what is Function 4 referred as and when does it get invoked/called?

Destructor, it is invoked as soon as the scope of the object

gets over.

iii. In Object Oriented Programming, which concept is illustrated by Function 1 and Function 3 together?

Constructor Overloading (Polymorphism)

2.Answer the questions (i) and (ii) 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;

}

};

i. Write statements in C++ that would execute Function 1, Function 2, Function 3 and Function 4 of class Test.

Test t1; // Execute funtion 1

Test t2("English"); // Execute funtion 2

Test t3(50); // Execute funtion 3

Test t4("Maths",49); // Execute funtion 4

ii. Which feature of Object Oriented Programming is demonstrated using Function 1, Function 2, Function 3 and Function 4 together in the above class Test?

Constructor Overloading (Polymorphism)

3.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

};

i. Write the definition of the constructor 1 so that the private member variables are initialized to 0.

Sample :: Sample()

{

x = 0;

y = 0;

}

ii. 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.

Sample :: Sample(int a)

{

x = a;

y = 0;

}

iii. Write the definition of the constructors 3 and 4 so that the private  
member variables are initialized according to the values of the parameters.

Sample :: Sample(int a, int b)

{

x = a;

y = b;

}

Sample :: Sample(int a, double b)

{

x = a;

y = b;

}

4. Create a class called Box with a variable: width of type double. Inside the class define a constructor and a friend that prints the width value(printWidth). In the main() define a Box instance, set values and call printWidth.

5. Perform addition operation on complex data using class and object. The program should ask for real and imaginary part of two complex numbers, and display the real and imaginary parts of their sum.

#include<iostream>

using namespace std;

class Complex{

public:

int real;

int imag;

/\* Function to set the values of

\* real and imaginary part of each complex number

\*/

void setvalue()

{

cin>>real;

cin>>imag;

}

/\* Function to display the sum of two complex numbers \*/

void display()

{

cout<<real<<"+"<<imag<<"i"<<endl;

}

/\* Function to add two complex numbers \*/

void sum(Complex c1, Complex c2)

{

real=c1.real+c2.real;

imag=c1.imag+c2.imag;

}

};

int main()

{

Complex c1,c2,c3;

cout<<"Enter real and imaginary part of first complex number"<<endl;

c1.setvalue();

cout<<"Enter real and imaginary part of second complex number"<<endl;

c2.setvalue();

cout<<"Sum of two complex numbers is"<<endl;

c3.sum(c1,c2);

c3.display();

return 0;

} #include<iostream>

using namespace std;

class Complex{

public:

int real;

int imag;

/\* Function to set the values of

\* real and imaginary part of each complex number

\*/

void setvalue()

{

cin>>real;

cin>>imag;

}

/\* Function to display the sum of two complex numbers \*/

void display()

{

cout<<real<<"+"<<imag<<"i"<<endl;

}

/\* Function to add two complex numbers \*/

void sum(Complex c1, Complex c2)

{

real=c1.real+c2.real;

imag=c1.imag+c2.imag;

}

};

int main()

{

Complex c1,c2,c3;

cout<<"Enter real and imaginary part of first complex number"<<endl;

c1.setvalue();

cout<<"Enter real and imaginary part of second complex number"<<endl;

c2.setvalue();

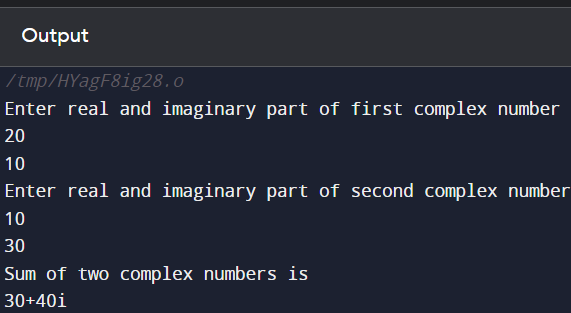
cout<<"Sum of two complex numbers is"<<endl;

c3.sum(c1,c2);

c3.display();

return 0;

}



6. Write a program in C++ having class string1 with members as

Int length;

Char \* buffer;

Implement this class using copy constructor, destructor , parameterized constructor and default constructor.

7. Write a program in C++ that counts the number of objects created in the class. Illustrate the program with the constructor and destructor both .

#include <iostream>

class MyClass {

private:

static int count;

public:

MyClass() {

//increment the count when an object is created

count++;

}

~MyClass() {

//decrement the count when an object is destroyed

count--;

}

//a function to get the current count of objects created

static int getCount() {

return count;

}

};

//initialize the static member variable to 0

int MyClass::count = 0;

//main function to create and destroy MyClass objects

int main() {

//create three MyClass objects

MyClass obj1, obj2, obj3;

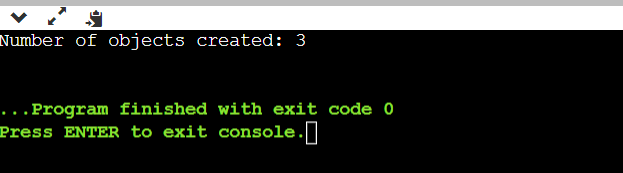
//print the current count of objects created

std::cout << "Number of objects created: " << MyClass::getCount() << std::endl;

//destroy the MyClass objects

return 0;

}



8. Write a program in C++ to calculate perimeter of all figures using the concept of friend function.