National University of Computer and Emerging Sciences, Lahore Campus



Course: **Object Oriented Programming** Course Code: **CS217** Program: **BS(Computer Science)** Semester: Fall 2019 60 Minutes Total Marks: 30 **Duration:** Paper Date: 8-Nov-2019 Weight 15 Section: ΑII Page(s): 4 Roll No: Exam: Sessional-II

- 1. Attempt all questions in the space provided in this sheet. You can use **rough** sheets but don't need to attach it here as it will not be marked.
- 2. Questions during exam are not allowed. Take reasonable assumptions where needed

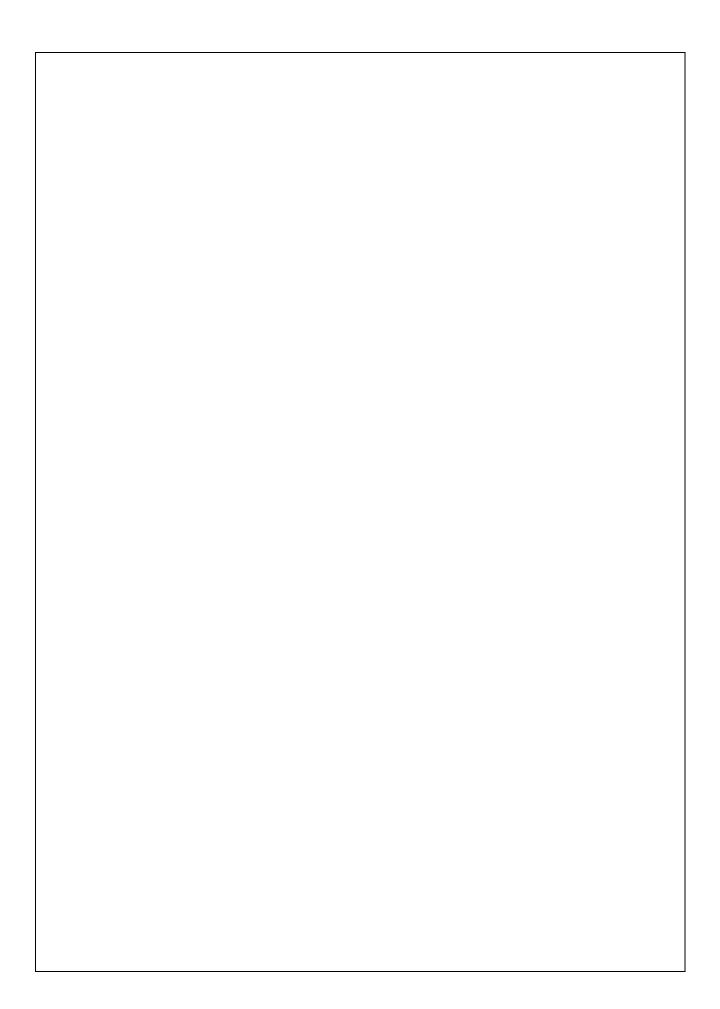
Question 1

Complex Number: [10 Points]

A complex number is a number that can be represented in the form of a+bi, where a and b are real numbers and i is the solution to the equation $x^2=-1$. Being a superset of real and imaginary sets, we can also perform arithmetic operations on complex numbers. For example, if you add 2+3i with 4+7i, it will result 6+10i. Similarly, if you multiply these two complex numbers it will result -13+26i.

Your task is to define a class **ComplexNumber** which makes the following *main* function to execute successfully. Moreover, overload all operators and functions that are essential for the following program.

```
int main()
{
      ComplexNumber C1(4.0, 7.0);
                                             //(4+7i)
      ComplexNumber C2(2.0, 3.0);
                                             //(2+3i)
      cout<<"Complex numbers are: "<< C1 << C2;</pre>
      cout<<"Addition of the two complex numbers is:"<< C1 + C2;</pre>
      cout<<"Multiplication of the two complex numbers is: " << C1 * C2;</pre>
      return 0;
}
class ComplexNumber
{
            float *real;
            float *imag;
public:
            ComplexNumber(float,float);//you can assume this is available to you
            //start your code here
```





Question 2

Syntax Errors: [20 Points]

Identify the syntax errors in the code below (by circling it) and then correct it (there)

```
#include <iostream>
#include <array>
#include <string>
#include<conio.h>
using namespace std;
class D {
      int y;
      void walk() {
             cout << "walk of D" << endl;</pre>
public:
      D(int y1 = 0) {
             y1 = y;
      }
class A {
public:
      int x;
      void print() {
             cout << "----A---- x:" << x << endl;
      A(int x1=0) {
             x = x1;
class B : A {
      Dx;
public:
      D getx() {
            return x;
      virtual void print() = 0;
      B(int x1, int y1):D(y1),A(x1){
```

```
};
class C : B {
public:
      int x;
      C(int x1 = 0, int x2 = 10, int x3 = 20):B(x1,x2) {
             x = x3;
      void print() {
             cout << "----C---- x:"<<x << endl;</pre>
             A::print();
             B::print();
      }
      void fun() {
             cout << "its fun" << endl;</pre>
      }
};
void main()
      B* p = new B;
      A* q = new A;
      q->print();
      q->A();
      B* ptr= new C;
      ptr->x = 35;
      ptr->print();
      ptr->getx().walk();
      C * p1 = dynamic_cast <C*>(ptr);
      (p1->fun()).fun();
      _getch();
      return 0;
}
```