National University of Computer and Emerging Sciences

School of Computing **Spring 2014** **Islamabad Campus**

CS103
Computer Programming

Saturday, April 5, 2014

Course Instructor(s)

Dr. Shahzad Rajput, Dr. Sibt ul Hussain and Mr. Naeem Ahmad

Serial No:

Sessional II

Total Time: 1 Hour

Total Marks: 100

_____

Signature of Invigilator

Student Name	Roll No	Section	Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. Use the back of the last page for rough work.
3. If you need more space write on the back side of the paper and clearly mark question and part number etc.
4. After asked to commence the exam, please verify that you have (**11**) different printed pages including this title page. There are total of (**3**) questions.
5. Use of calculator is strictly prohibited.
6. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
7. Use **proper indentation** while writing code and make sure that your code is legible. Failing to do so can cost you marks.

	I	II	III	**Total**
Total Marks	40	20	40	100
Marks Obtained				

Vetted By: _____ Vetter Signature: _____

Question I . . . . . (40 Marks)

(1) **(5 Marks)** What would be the output produced by executing the following C++ code? Identify and correct errors, and write output, if any.

```
#include <iostream>
using namespace std;
class GuessMe {
        int *p;
public:
        GuessMe(int x = 0) {
                p = new int;
                *p = x;
        }
        int GetX() {
                return *p;
        }
        void SetX(int x) {
                *p = x;
        }
        ~GuessMe() {
                delete p;
        }
};

int main() {
        GuessMe g1;
        g1.SetX(10);

        GuessMe g2(g1);
        cout << g2.GetX() << endl;
        return 0;
}
```

(2) **(5 Marks)** What would be the output produced by executing the following C++ code? Identify and correct errors, if any.

```
#include <iostream>
using namespace std;
class A {
public:
        virtual void print() {
                cout << "A" << endl;
        }
};

class B: public A {
};
```

```
12
13  class C: public B {
14  public:
15          void print() {
16                  cout << "C" << endl;
17          }
18  };
19
20  int main() {
21          C c;
22          B *b = &c;
23          b->print();
24          return 0;
25  }
```

(3) **(10 Marks)** The following C++ code has error(s) and you are not allowed to modify:

- Class A
- Display function in class B

What would you modify in class B to remove the error(s) such that the following output is produced:
num=5
gum=10

```
1   #include<iostream>
2   using namespace std;
3   class A {
4   private:
5           int num;
6   public:
7           A(int x = 0) :
8                           num(x) {
9           }
10          A(const A& a) :
11                          num(a.num) {
12          }
13          void Display() {
14                  cout << "num=" << num << endl;
15          }
16          void SetNum(int x) {
17                  num = x;
18          }
19  };
20
21  class B: public A {
22  private:
23          int gum;
24  public:
25          B(int x = 0, int y = 0) :
26                          A(x), gum(y) {
27          }
```

```
28          B(const B& b) {
29                  num = b.num;
30                  gum = b.gum;
31          }
32          void Display() {
33                  A::Display();
34                  cout << "gum=" << gum << endl;
35          }
36  };
37
38  int main() {
39          B b1(5, 10);
40          B b2(b1);
41          b2.Display();
42          return 0;
43  }
```

(4) **(5 Marks)** What would be the output produced by executing the following C++ code? Identify and correct errors, if any.

```
1   #include<iostream>
2   using namespace std;
3   class A {
4           int x;
5   public:
6           A(int val = 0) :
7                           x(val) {
8                   cout << "A  " << x << endl << flush;
9           }
10          A(const A& a) {
11                  x = a.x;
12                  cout << "B  " << x << endl << flush;
13          }
14          void SetX(int x) {
15                  this->x = x;
16          }
17          ~A() {
18                  cout << "D  " << x << endl << flush;
19          }
20  };
21
22  A f(A a) {
23          cout << " C " << endl << flush;
24          a.SetX(100);
25          return a;
26  }
27
```

```
int main() {
        A a(1);
        A b=f(a);
        b.SetX(-100);
        return 0;
}
```

(5) **(15 Marks)** What would be the output produced by executing the following C++ code? Identify and correct errors, if any.

```
#include <iostream>
#include<cassert>
using namespace std;
class Point3D
{
public:
        Point3D() {
                p[0] = p[1] = p[2] = 0;
        }
        Point3D::Point3D(int x_, int y_, int z_) {
                p[0] = x_; p[1] = y_; p[2] = z_;
        }
        Point3D operator*(const int & v) {
                Point3D v1;
                v1[0] = p[0] + v;   v1[1] = p[1] + v;  v1[2] = p[2] + v;
                return v1;
        }

        Point3D operator+(const Point3D &v) {
                Point3D v1;
                v1[0] = p[0] + v[0];  v1[1] = p[1] + v[1]; v1[2] = p[2] + v[2];
                return v1;
        }

        Point3D operator-(const Point3D &v) {
                Point3D v1;
                v1[0] = p[0] - v[0];  v1[1] = p[1] - v[1]; v1[2] = p[2] - v[2];
                return v1;
        }

        Point3D operator-() {
                Point3D v1;
                v1[0] = -p[0];         v1[1] = -p[1];         v1[2] = -p[2];
                return v1;
        }
        bool operator==(const Point3D &v) {
                return p[0] == v[0] && p[1] == v[1] && p[2] == v[2];
        }
        int operator[](const int & i) const {
```

```
40                  assert(i >= 0 && i <= 2); // check for index with-in range
41                  return p[i];
42          }
43          int & operator[](const int & i) {
44                  assert(i >= 0 && i <= 2); // check for index with-in range
45                  return p[i];
46          }
47  private:
48          int p[3];
49  };
50  ostream &operator<<(ostream &out, const Point3D&v) {
51          out << " X = " << v[0] << " Y = " << v[1] << " Z = " << v[2] << endl
52                          << flush;
53          return out;
54  }
55
56  int main() {
57          Point3D p1(10, 20, 30), p2(20, 30, 40);
58          cout << " P1 : " << p1 << " P2 : " << p2 << endl;
59
60          Point3D p3;
61          p3[0] = 5;
62          p3[1] = 5;
63          p3[2] = 5;
64
65          Point3D p4 = -p1 - p2 - p1 * !(p1 == p2);
66          cout << " P4 : " << p4 << endl;
67  }
```

Question II . . . . . (20 Marks)

Develop a class *Polynomial*. The internal representation of a *Polynomial* is an array of terms. Each term contains a coefficient and an exponent, e.g., the term $2x^4$ has the coefficient 2 and the exponent 4. Develop a complete class containing proper constructor(s) and destructor as well as *set* and *get* functions. The class should also provide the following overloaded operator capabilities:

1. Overload the addition operator (+) to add two *Polynomials*.
2. Overload the subtraction operator(-) to subtract two *Polynomials*.
3. Overload the assignment operator to assign one *Polynomials* to another.
4. Overload the addition assignment operator (+=), subtraction assignment operator (-=), and multiplication assignment operator (*=).

Question III . . . . . (40 Marks)

Design a *Ship* class that has the following members:

- A member variable for the name of the ship (a string)
- A member variable for the year that the ship was built (a string)
- A constructor and appropriate accessors and mutators
- A virtual *print* function that displays the ship's name and the year it was built.

Design a *CruiseShip* class that is derived from the Ship class. The *CruiseShip* class should have the following members:

- A member variable for the maximum number of passengers (an *int*)
- A constructor and appropriate accessors and mutators
- A *print* function that overrides the *print* function in the base class. The *CruiseShip* class's print function should display only the ship maximum number of passengers.

Design a *CargoShip* class that is derived from the Ship class. The *CargoShip* class should have the following members:

- A member variable for the cargo capacity in tonnage (an *int*).
- A constructor and appropriate accessors and mutators.
- A *print* function that overrides the print function in the base class. The *CargoShip* class's print function should display only the ship's name and the ship's cargo capacity.

Demonstrate the classes in a program that has an array of *Ship* pointers. The array elements should be initialized with the addresses of dynamically allocated *Ship*, *CruiseShip*, and *CargoShip* objects. The program should then step through the array, calling each object's *print* function.