National University of Computer and Emerging Sciences

School of Computing **Spring 2015** **Islamabad Campus**

CS103
Computer Programming

Saturday, April 25, 2015

Course Instructor(s)

Dr. Sibt ul Hussain, Dr. Usman Farrokh and Dr. Fareed Ahmad

Serial No:

Mid-II

Total Time: 1 Hour

Total Marks: 85

____

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 (**12**) 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.
8. **Please allocate your time properly according to the marks distribution.**
9. Write proper explanation of the error (or bug) where required, without proper explanation no marks will be awarded.

	I	II	III	**Total**
Total Marks	35	20	30	85
Marks Obtained				

Vetted By: ______________ Vetter Signature: ______________

Question I . . . . **(35 Marks)**

Please write proper explanation of the bug where required, without proper explanation no marks will be awarded, note there are no syntax errors (look for only logical ones) in the given set of codes.

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

```cpp
#include <iostream>
using namespace std;
class Number {
private:
        int n;
public:
        Number() : n(0) {
                cout << n;
        }

        Number( int nn )
        : n(nn)
        {
                cout << n;
        }

        Number(Number const& otherNum)
        : n(otherNum.n+1)
        {
                cout <<  n;
        }

        void display() { cout << n; }
        void increase() { n += 1; }
};
int main(){
        Number a, b(1), c(b);
        b.increase();
        c.display();
        b.display();
}
```

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

```cpp
#include <iostream>
using namespace std;
class Test{
private:
    int val;
public:
  Test(const Test& _other) : val(_other.val + 1)
  { }

  Test(int _val) : val(_val) { }

  int get_val() const{ return val;}

  const Test&  operator=(Test const& _other){
    val = _other.val;
    return *this;
  }
  Test  operator+(Test const& _other){
    Test t(val+ _other.val);
    return t;
  }
};

ostream& operator<<(ostream& stream, const Test & _val){
  stream << _val.get_val();
  return stream;
}

int main( ){
  Test a(1), b(2),c(a+b);
  a = b+c;
  cout << a << " " << b<<" "<<c;
}
```

(3) **(10 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 Point {
private:
        double x;
        double y;
public:
        Point(int x_=0,int y_=0) { x = x_; y = y_; }

        void setX(double newX) { x = newX; }
        void setY(double newY) { y = newY; }

        double getX() const { return x; }
        double getY() const { return y; }
        ostream & operator<< (ostream & out)
        {
                out<< x<<" "<< y;
                return out;
        }
        Point operator+(const int& x_)
        {

                return Point(50, 100);
        }
};

Point operator+(const int& x, const Point &p)
{
        return Point(p.getX()+x, p.getY()+x);
}
int main()
{
        Point p(4,5),p2;
        p2=5+p;
        p<<cout<<endl;
        p2<<cout<<endl;
}
```

(4) **(10 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 Memory {
        float capacity;
public:
        Memory(int cap = 1) {
                capacity = cap;
                cout << " Added Memory of Capacity= "
                                << capacity << " G " << endl;
        }
        ˜Memory() {
                cout << " Removed Memory of Capacity= "
                                << capacity << " G " << endl;
        }
};
```

```
class Core {
        float speed;
public:
        Core(float speed_ = 3.3) {
                speed = speed_;
                cout << " Added 1 Core of Speed= "
                                << speed << " GHz " << endl;
        }
        ~Core() {
                cout << " Removed 1 Core of Speed= "
                                << speed << " GHz " << endl;
        }
};
class Processor {
        const int ncores;
        Core cores[4];
public:
        Processor() :
                        ncores(4) {
                cout << " Added a Processor of  "
                                << ncores << " Cores " << endl;
        }
        ~Processor() {
                cout << " Removed a Processor of = "
                                << ncores << " cores " << endl;
        }
};
class Mobile {
        Memory m;
        Processor p;
public:
        Mobile() {
                cout << "  Building a Mobile  " << endl;
        }
        ~Mobile() {
                cout << "  Destroying a Mobile " << endl;
        }
};
int main() {
        Mobile m;
        cout << "  :) The End  " << endl;
}
```

(5) **(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 Mystery {
        int x, y;
public:
        Mystery(int x_ = 1, int y_ = 0) :
                        y(x + y_), x(x_) {

        }
        void print(ostream & out) const {
                out << "X =" << x << " Y=" << y;
        }
        Mystery & SetX(int x_) {
                x = x_;
                return *this;
        }
        Mystery & SetY(int y_) {
                y = y_;
                return *this;
        }
};
void operator<<(ostream & out, const Mystery & m) {
        m.print(out);
}
int main() {
        Mystery m1(5, 10), m2;
        m2.SetX(-1).SetY(500);
        cout << "M2=" << m2;
        cout << endl;
        cout << "M1=" << m1;
}
```

Question II . . . . (20 Marks)

Your goal is to write a program for creating a Phonebook. Your Phonebook should allow for storage of many contacts, where each contact should be represented by a person name and his phone number. In addition, to basic functionality your Phonebook should allow facility of performing following operations:

1. The union of two Phonebooks (Should be done by overloading + Operator)
2. The direct access of a contact's phone number via name (overloading the operator []). User should be able to both read and write the phone number via this operator.
3. Printing and addition of a contact using stream insertion and extraction operators.

Note: You are not required to write the main function.

Question III . . . . **(30 Marks)**

Your goal here is to write classes for creating a bouquet of flowers. To create the bouquet of flower your will need to write following two classes.

Design a class *Flower*. A "flower" is characterized by following attributes:

- a name
- a color
- a basic price per unit
- an indication whether the flower is perfumed or not
- and an indication to know whether the flower is on sale.

and with following behavior:

- a constructor initializing the attributes using parameters given in the order shown by the provided main(); a default constructor will not be necessary but the last two parameters will have false as default value;
- a price method returning the flower's price : the price will be the base price if the flower is not on sale; otherwise, the price will be half the base price;
- a bool parfume() method indicating whether the flower is perfumed or not;
- Overloaded stream insertion operator.
- an overloading of the == operator returning true if two flowers are identical, false otherwise. Two flowers are considered identical if they have the same name, the same color, and the two flowers are both either perfumed or not (neither the price nor the fact that the flower is on sale or not is involved in the comparison).

Next write a "Bouquet" class which will be modeled using a dynamic array of Flowers.

The Bouquet class offers the following methods :

- a method bool parfume() returning true if the bouquet is perfumed and false otherwise; a bouquet is perfumed if at least one of its flowers is perfumed;
- a method price without parameters returning the price of the bouquet of flowers; This is the sum of the prices of all its flowers; this sum is multiplied by two if the bouquet is perfumed;
- a stream insertion method, should display all information of bouquet with the total price.
- an overload of the += operator which allows adding a flower to the bouquet, the flower will always be added at the end.
- an overload of the -= operator taking as a parameter a flower and removing from the bouquet all the flowers identical to the latter (according to the definition of the == operator);
- an overloaded + operator according its usage in the provided main
- an overloaded - operator according to its usage in the provided main

```
int main() {
        // example of  Yellow oderless rose.
        Flower r1("Rose", "Yellow", 1.5);
        cout << r1 << endl;
        // example of  Yellow parfumed rose
        Flower r2("Rose", "Yellow", 3.0, true);
        // example of  parfumed Red rose on sale
        Flower r3("Rose", "Red", 2.0, true, true);
        Bouquet b1;
        b1 += r1; // add one Flower of r1 type
        b1 += r1; // add another Flower of r1
        b1 += r2;
```

```
    b1 += r3;
    cout << b1 << endl;

    b1 = b1 - r1; // Delete all the Flowers of type r1
    cout << b1 << endl;

    Bouquet b2;
    b2 = b1 + r1; // Add one Flower of type r1
    cout << b2 << endl;

    // Delete all the parfumed flowers from the bouquet.
    b2 -= r2;
    b2 -= r3;
    cout << b2;
    return 0;
}
```