National University of Computer and Emerging Sciences

School of Computing **Spring 2016** **Islamabad Campus**

CS103
Computer Programming

Monday, March 28, 2016

Course Instructor(s)

Sibt ul Hussain, Usman Farrokh and Hassan Mustafa

Serial No:

Sessional II

Total Time: 1 Hour

Total Marks: 90

_____

Signature of Invigilator

Student Name	Roll No	Section	Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

Instructions:

1. Answer all the questions on this paper. Read the question carefully, understand the question, and then answer it.
2. No additional sheet will be provided for rough work. Utilize the empty spaces for rough work.
3. 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.
4. Use of calculator is strictly prohibited.
5. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
6. Use **proper indentation** while writing code and make sure that your code is legible. Failing to do so can cost you marks. You can use comments or a little description to explain your code.
7. **Please allocate your time properly according to the marks distribution.**

	I	II	III	**Total**
Total Marks	46	30	14	90
Marks Obtained				

Vetted By: _____ Vetter Signature: _____

Question I . . . . . **(46 Marks)**

Please write proper explanation of the error (or bug) where required, without proper explanation no marks will be awarded.

Read the statement before each question carefully.

(a) **(12 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
#include <iostream>
using namespace std;
class Point{
        int x, y;
        public:
        Point(int a=0, int b=0){
                x=a;
                y=b;
                print();
        }
        void print(){
                cout<<" ("<<x<<","<<y<<") " <<endl;
        }
        ~Point(){
                cout<<"Point is going"<<endl;
        }
};
class Circle
{
        Point center;
        float radius;
        public:
        Circle():center(0,0){
                radius=0;
                cout<<"The basic circle"<<endl;
        }
        Circle(Point p):center(p){

        }
        Circle(const Circle & c):center(c.center), radius(c.radius){
                cout<<"The copied circle";
                center.print();
        }
        ~Circle(){
                cout<<"Circle is going"<<endl;
        }
};

int main() {
        Point p1;
        Circle c1;
        static Circle c2(p1);
        Circle c3(c2);
        return 0;
}
```

(b) **(6 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
#include <iostream>
using namespace std;
class Dummy
{
        float z;
        int x, y;
        public:
        Dummy(int x=0, int y=1):x(x+2),y(y+3){
                z = x + y + 1;
        }
        void print(){
                cout<< " X= "<< x
                  <<" Y = "<< y
                  <<" Z = "<< z;
        }
};

int main() {
        Dummy d(10);
        d.print();
        return 0;
}
```

(c) **(8 Marks)** What will be the output of following code. Explain the error or bug if there is any.

```
#include <string>
#include <iostream>
using namespace std;
class Volume {
        int volume;
        public:
        Volume(int l, int h, int w){
                volume = l*h*w;
        }
        Volume(int v){
                volume=v;
        }
        int getVolume() const {
                return volume;
        }
        Volume  operator ++(){
                int t = volume++;
                Volume temp(t);
                return *this;
        }
        Volume operator ++(int){
                Volume temp(0);
                ++*this;
                return temp;
        }
};
ostream& operator<<(ostream& out, const Volume& v){
        out<<"The volume is "<<v.getVolume()<<endl;
        return out;
}
int main(){
        Volume v(2,3,4);
        Volume v2(4,4,4);
        v2=++v;
        v2=v++;
        cout<<v2;
        cout<<v;
        v2=++v;
        cout<<v2;
        cout<<v;
        return 0;
}
```

(d) **(12 Marks) Identify and correct the errors in the following code so that the main() function could run. You cannot change the main() function!**

```
#include <string>
#include <iostream>
using namespace std;
class Employee{
          string name;
          const int nClients;
          string *Clients;
        Employee();
  public:
        Employee(char E[], int n){
                name = E;
                nClients = n;
                Clients = new string[nClients];
        }
        string & operator[](int i){
                return Clients[i];
        }
        int getnClients(){
                return nClients;
        }
        ~Employee(){
                delete [] Clients;
        }
};


void operator>>(istream in, Employee E){
        cout<<"Enter the names of "<<E.getnClients()<<" clients:"<<endl;
        for(int i = 0; i < E.getnClients() ;i++)
                in >> E[i];
}

void display(Employee E){
          for(int i = 0; i < E.getnClients() ;i++)
                  cout<<E[i]<<endl;
}

int main(){
Employee E1("Abassi", 2);
cin>>E1;
display(E1);
return 0;
}
```

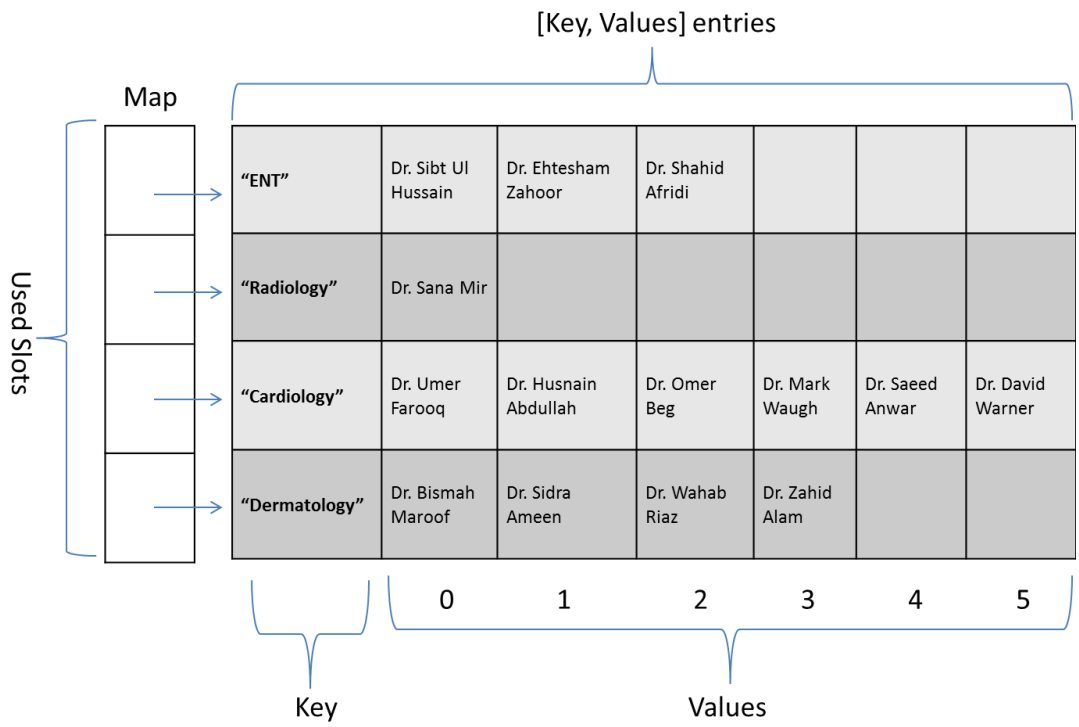
(e) **(8 Marks) Identify and correct the errors in the following code so that the main() function could run. You cannot change the main() function!**

```
#include <iostream>
using namespace std;
class Complex
{
        float real, imag;
        public:
        void setR(float x){
                real = x;
        }
        void setI(float x){
                imag = x;
        }
        float getR(){
                return real;
        }
        float getI(){
                return imag;
        }
        void display(){
                cout<<real<<" + i"<<imag;
        }
};

ostream & operator<<(ostream &o, const Complex &C){
        C.display();
}
int main() {
        Complex C;
        cout<<C.setR(3).setI(3);  //should print 3 + i3
        return 0;
}
```

Question II ..... **(30 Marks)**

Consider the following figure:



We need a map (or dictionary) to store data about medical practitioners. A map consists of multiple [key, values] entries, where extra space is dynamically allocated when needed. Typically multiple values may be mapped against one key. For simplicity, we assume that maximum six (06) values can be stored against one key (as shown in figure). Here key is the specialty of medical practitioner, and values are names of medical practitioners.

You are required to design and implement map class (and any other required classes) in C++, where following code should run without any syntax/logical errors:

```
void main()
{
        CMap map, map2, map3; // assume CMap is the name of map class

        // operator[] is used to get "entry" against given "key" from map and
        // can be used in following context with operator= to create/refresh an entry
        map["ENT"] = "Dr. Sibt Ul Hussain";
        map["Radiology"] = "Dr. Sana Mir";
        map2["Cardiology"] = "Dr. Umer Farooq";

        // operator+= with an "entry" is used to add new "value" against given "key".
        // If entry with given "key" does not exist then new entry is created in map
        map["ENT"] += "Dr. Ehtesham Zahoor";
        map["ENT"] += "Dr. Shahid Afridi";
        map2["Cardiology"] += "Dr. Husnain Abdullah";
        map2["Cardiology"] += "Dr. Omer Beg";
        map2["Cardiology"] += "Dr. Mark Waugh";
        map2["Cardiology"] += "Dr. Saeed Anwar";
```

```
19          map2["Cardiology"] += "Dr. David Warner";
20
21          // following statement creates new entry in map, and then add values to it
22          map2["Dermatology"] += "Dr. Bismah Maroof";
23          map2["Dermatology"] += "Dr. Sidra Ameen";
24          map2["Dermatology"] += "Dr. Wahab Riaz";
25          map2["Dermatology"] += "Dr. Zahid Alam";
26
27          // operator+ is also used to add two maps and return a new map.
28          // operator= is used to assign one map to other (deep copy).
29          map3 = map + map2; // map3 now looks like the map in figure above!
30
31          // operator-= is used to remove value from the map against key = ENT.
32          // If value does not exist, then nothing changes in the map. If a "key"
33          // has no remaining value against it, then the "key" must also be removed.
34          map["ENT"] -= "Dr. Ehtesham Zahoor";
35  }
```

Question III . . . . . (14 Marks)

Your goal in this question is to design a mobile application management system in other words an "App Store". In the application store you can store information about different type of applications such as *system* and *entertainment* applications. For all the types of applications you will store application id (integer), programmer id (integer), and date of publishing. System applications include number of threads they can use and number of hardware devices they can access. Entertainment applications have predefined storage limit and already defined features.

Apart from the applications' information, App Store also stores user information, i.e. his name and email. Furthermore, it also records for each user what applications he is currently using.

Now given this description, identify all the classes, their data members and their relationships (composition, aggregation, inheritance) for the complete system. I.e. implement the identified classes without their member functions.