

## Object Oriented Programming 2<sup>nd</sup> Midterm Examination

### Question1:

- a) In **main** program (given below), 4 objects (a, b, c, e) are created. Explain, which data elements includes each of these objects.
- b) Some statements, in **main** program are incorrect. List them and explain the reason of errors.
- c) If the incorrect lines of **main** are discarded, what will be written on the screen when the C++ program below is compiled and run?

```
#include <iostream.h>
class A{
    int i1;
protected:
    int i2;
public:
    A(){ cout <<"Function A1"<< endl; i1=0; i2=1;}
    void seti(int inp){ cout <<"Function A2"<< endl;
                        i1=inp; i2=0; }
    void f1(char *c){ cout <<"Function A3:"<< c << endl;}
    void print(){ cout <<"Ai1="<<i1<<" Ai2="<< i2 <<
endl;}
    ~A(){ cout <<"Function A4"<< endl;}
};
class B:public A{
    int i1;
protected:
    int i3;
public:
    B(){ cout <<"Function B1"<< endl; i1=0; i3=1; }
    void seti(int inp){ cout <<"Function B2"<< endl;
                        i1=inp; i3=3; }
    void f1(int i){ cout <<"Function B3:"<< i << endl;}
    void print(){ A::print();
                  cout <<"Bi1="<<i1<<" Bi3="<< i3 << endl;}
    ~B(){ cout <<"Function B4"<< endl;}
};
class C:private A{
    int i1;
public:
    C(){ cout <<"Function C1"<< endl;
        i1=0;}
    void print(){ A::print();
                  cout <<"Ci1="<<i1 << endl;}
    ~C(){ cout <<"Function C2"<< endl;}
};
class D{
public:
    D(){ cout <<"Function D1"<< endl;}
    void f1(char *c){ cout <<"Function D2:"<< c << endl;}
    ~D(){ cout <<"Function D3"<< endl;}
};
class E:public B,public D{
    int i1;
public:
    E(){ cout <<"Function E1"<< endl;
        i1=0;}
    ~E(){ cout <<"Function E2"<< endl;}
};
```

```
void main()
{
    A a;
    B b;
    C c;
    E e;
    a.i2=1;
    b.A::seti(3);
    a.seti(2);
    b.print();
    b.f1("INPUT1");
    c.A::seti(4);
    c.print();
    c.f1("INPUT2");
    e.seti(5);
    e.B::i3=7;
    e.print();
    e.f1("INPUT3");
}
```

### Question2:

A program to model the kinds of people one finds in a university will be written. The major categories are `students` and `teachers`. These classes are to be derived from a `person` class.

The `person` class includes a name. The `student` class includes the student's grade point average (GPA), and the `teacher` class includes the number of scholarly papers the teacher has published.

All the classes contain constructors to fill all data spaces. The `putData()` function prints the name and the GPA (of a student) or the number of published papers (of a teacher). They also contain a function called `isOutstanding()`, which makes it easy for the school administrators to create a list of outstanding students and teachers to give them some awards. Students with a GPA over 3.5 and teachers who have published more than 100 papers are considered outstanding. This function returns 1 if someone is outstanding or 0 if not.

In `main()`, define an array of pointers, which can point to students and teachers. The program first asks the kind of person and then lets the user enter the name. For students, the program also asks for the GPA; for teachers, it asks for the number of publications. The maximum number of pointers to objects, which can be kept in the array, is 100. When the user is finished, the program prints out the names and other data for all persons, noting those students and teachers who are outstanding.

Here's some sample interaction:

```
Enter student or teacher (s/t): s
  Enter name: Timmy
  Enter student's GPA: 1.2
  Enter another (y/n)? y
Enter student or teacher (s/t): s
  Enter name: Brenda
  Enter student's GPA: 3.9
  Enter another (y/n)? y
Enter student or teacher (s/t): t
  Enter name: Shipley
  Enter number of teacher's publications: 714
  Enter another (y/n)? n

Name = Timmy
  GPA = 1.2
Name = Brenda
  GPA = 3.9 (This person is outstanding)
Name = Shipley
  Publications = 714 (This person is outstanding)
```

- a) Write all necessary classes (including their methods).
- b) Write and explain the main program.