İ.T.Ü.
Computer Engineering Department
Dr. Feza BUZLUCA
Çiğdem ŞENGÜR

14.05.2001

Object Oriented Programming 2nd Midterm Examination

Question 1:

```
#include <iostream.h>
class A{
 protected:
  int *ip1;
 public:
  A(){
   cout <<"Function A1"<< endl;
   ip1= new int;
   *ip1=0;
  }
  A(const A& aa){
   cout <<"Function A2"<< endl;
   ip1=new int;
   *ip1=*aa.ip1;
  }
  void operator=(const A& aa){
    cout <<"Function A3"<< endl;
    *ip1=*aa.ip1;
  }
  virtual void f1(char *c)=0;
  virtual void f2(){
    cout <<"Ai1="<<*ip1<< endl;}
  void f3(int i){
    *ip1=i;
    cout <<"Ai1="<<*ip1 << endl;
  }
  ~A(){
    delete ip1;
    cout <<"Function A4"<< endl;}
};

class B:public A{
 protected:
  int *ip2;
 public:
  B(){
    cout <<"Function B1"<< endl;
    ip2= new int;
    *ip2=0;
  }
  void f1(char *c){
    cout <<"Function B2 "<<c<< endl;}
  virtual void f2(){
    cout <<"Bi1="<<*ip1<< endl;
    cout <<"Bi2="<<*ip2<< endl;
  }
  void f3(int i1){
    *ip2=i1;
    cout <<"Bi2="<<*ip2 << endl;
  }
  ~B(){
     delete ip2;
       cout <<"Function B3"<< endl;}

class C{
 public:
  C(){ cout <<"Function C1"<< endl;}
  void f1(char *c){
    cout <<"Function C2:"<< c << endl;}
  void operator=(const C &cc){
    cout <<"Function C3"<< endl;}
  ~C(){ cout <<"Function C4"<< endl;}
};

class D:public B{
  C c1;
  int i1;
 public:
  D(){
    cout <<"Function D1"<< endl;
    i1=0;
  }
  void f1(char *c){
    cout <<"D:"<< c << endl;}
  void f2(){
    cout <<"Function D2"<< endl;}
  void f3(){
    cout <<"Function D3"<< endl;}
  ~D(){ cout <<"Function D4"<< endl;}
};

void main()
{
  A a;
  B b;
  C c;
  D d1,d2;
  B b2=b;
  A* ap[4];
  ap[0]= new A;
  ap[1]=&b;
  ap[2]=&c;
  ap[3]=&d1;
  ap[3]->f1("Message 1");
  ap[3]->f2();
  ap[3]->f3(5);
  ap[1]->f3(2);
  B *bp=&d1;
  bp->f1("Message 2");
  bp->f3(5);
  A *ap2=new B;
  ap2->f3(3);
  delete ap2;
  d1.f3(5);
  d2=d1;
}
```

```
};
```

a) Some statements, in **main** program (given below), cause compile-time errors. List incorrect statements and explain the reasons of errors.

b) If the incorrect lines of **main** are discarded, what will be written on the screen when the C++ program below is compiled and run (ignore possible run-time errors).

c) Some of the classes are not written properly. Explain defects of classes, which may cause run-time errors. Make necessary modifications.

Question 2:

A linked list of geometrical objects will be created.
A **geometrical object** can be:

- **Square:** x,y location of left upper corner and side length.
- **Rectangle:** x,y location of left upper corner, long side length and short side length.
- **Circle:** x,y location of center and radius.

All classes will have constructors, which take necessary parameters to initialize objects. They will also have destructors, which will print the type and properties of the object being destructed, on the screen.

The user should be able to :

- Display the list. (display the type and properties of all objects.)
- Add objects to the **end** of the list**.** The method will take the address of the object as a parameter.
- Delete objects from the **head** of list.
- Print the area of all objects in the list. (The type and area of all the objects in the list should be printed out.)

a) Write all the necessary classes in C++, using the object-oriented approach. Classes must include all necessary methods.

b) Write a **main** program that creates a list of objects and invokes all the methods for list. The type of objects will be determined by the user. Also the user will provide all the other necessary parameters.