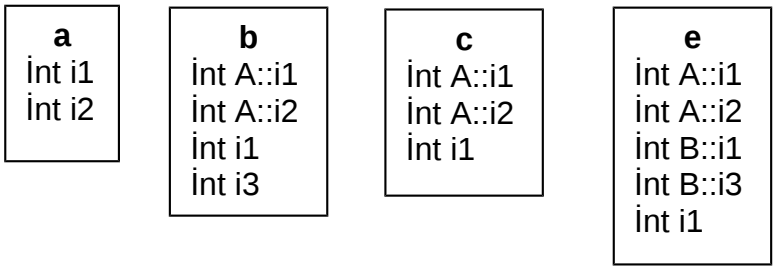
İ.T.Ü. Computer Engineering Department 08.05.2000
Dr. Feza BUZLUCA
Çiçek ÇAVDAR

Object Oriented Programming 2nd Midterm Examination Solutions

Question1:

a) Data elements of objects:



b) Incorrect statements in **main** program :

`a.i2=1;`	protected members are not accessible from objects of the class.
`b.f1("INPUT1");`	f1 in B takes a parameter of type int (f1 of A is overridden)
`c.A::seti(4);`	A is private base class of C. Public members of A are private in C.
`c.f1("INPUT2");`	A is private base class of C. f1 is not accessible for objects of C.
`e.B::i3=7;`	protected members are not accessible from objects of the class.
`e.f1("INPUT3");`	f1 is ambiguous in E. B::f1 or D::f1

c) If the incorrect lines of **main** are discarded, the output of program is:

```
Function A1   Constructor for a
Function A1   Constructor for b  A::
Function B1   Constructor for b  B::
Function A1   Constructor for c  A::
Function C1   Constructor for c  C::
Function A1   Constructor for e  A::
Function B1   Constructor for e  B::
Function D1   Constructor for e  D::
Function E1   Constructor for e  E::
Function A2   b.A::seti(3),  seti() in A is called for object b
Function A2   a.seti(2),  seti() in A is called for object a
Ai1=3 Ai2=0  b.print(), it calls A::print,  elements of b, derived from A are printed
Bi1=0 Bi3=1   b.print continues, elements of b are printed
Ai1=0 Ai2=1   c.print(), it calls A::print,  elements of c, derived from A are printed
Ci1=0         c.print continues, element of c is printed
Function B2   e.seti(5) calls the seti of B
Ai1=0 Ai2=1   e.print() calls print of B. It calls A::print,  elements of e, derived from A are printed
Bi1=5 Bi3=3   B::print() continues,  elements of e, derived from B are printed
Function E2   Destructor for e
Function D3   Destructor for e. Destr. of  E calls destr. of  D
Function B4   Destructor for e. Destr. of  E calls destr. of  B
Function A4   Destructor for e. Destr. of  B calls destr. of  A
Function C2   Destructor for c
Function A4   Destructor for c. Destr. of  C calls destr. of  A
Function B4   Destructor for b
Function A4   Destructor for b. Destr. of  B calls destr. of  A
Function A4   Destructor for a
```

Question2:
a)

```
#include <iostream>
using namespace std;
class person                    // person class
{
  protected:
     char name[40];
  public:
     person(){                        // Constructor
       cout << "  Enter name: ";
       cin >> name;
     }
     virtual void putData(){
       cout << endl << "Name = " << name;}
     virtual bool isOutstanding()=0;      // Pure virtual (abstract class)
};
class student : public person       // student class
{
     float gpa;               // grade point average
  public:
     student(){               // Constructor
       cout << "  Enter student's GPA: ";
       cin >> gpa;
     }
     void putData(){
       person::putData();
       cout << endl << "  GPA = " << gpa;
     }
     bool isOutstanding(){
       if (gpa > 3.5) return true;
          else return false;
     }
};
class teacher : public person       // teacher class
{
     int numPubs;             // number of papers published
  public:
     teacher(){               // Constructor
       cout << "  Enter number of teacher's publications: ";
       cin >> numPubs;
     }
     void putData(){
       person::putData();
       cout << endl << "  Publications = " << numPubs;
     }
     bool isOutstanding(){
       if(numPubs > 100) return true;
         else return false;
     }
};
```

b)

```
void main()
{
  person* persPtr[100];        // list of pointers to persons
  int n = 0;                   // number of persons on list
  char choice;                 // 's' or 't'
  do{
     cout << "Enter student or teacher (s/t): ";
     cin >> choice;
     if(choice=='s')                    // put new student
        persPtr[n++] = new student;     //    in array
     else                               // put new teacher
        persPtr[n++] = new teacher;     //    in array
     cout << "   Enter another (y/n)? ";  // do another person?
     cin >> choice;
  } while( (choice=='y') && (n<100) );   // cycle until not 'y'
  for(int j=0; j<n; j++)
  {                                      // print names of all
     persPtr[j]->putData();              // persons, and
     if (persPtr[j]->isOutstanding())
        cout << " (This person is outstanding)";  // say if outstanding
     delete persPtr[j];                  // delete object
   }
}     // end main()
```