İ.T.Ü.   02.04.2001
Computer Engineering Department
Dr. Feza BUZLUCA
Çiğdem ŞENGÜR
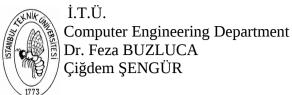
Object Oriented Programming 1ˢᵗ Midterm Examination

Question1:

a) This C++ program can be compiled without any errors, but class **B** can cause run-time errors. Explain this fact. Re-write class **B** to correct these errors. Modify only the functions, which may cause errors. Do not change other functions.

```cpp
#include<iostream.h>
class A{
    int ai;
 public:
    A();
    A(A&);
    ~A();
    void operator=(A);
    int fA() const {return ai;}
};
class B{
    A a;
    A *p;
 public:
    B(int);
    B(B&);
    ~B();
    A &f5(B);
};
A::A(){
    cout<<"Message 1" <<endl;
    ai=1;
}
A::A(A& a){
    cout<<"Message 2" <<endl;
    ai=a.ai;
}
A::~A(){
    cout<<"Message 3" <<endl;
}
void A::operator=(A a){
    cout<<"Message 4" <<endl;
    ai=a.ai;
}
```

```cpp
B::B(int i){
    cout<<"Message 5" <<endl;
    p=new A[i];
}
B::B(B& b){
    cout<<"Message 6" <<endl;
    p=b.p;
    a=b.a;
}
B::~B(){
    cout<<"Message 7" <<endl;
    delete[] p;
}
A &B::f5(B b){
    cout<<"Message 8" <<endl;
    if(a.fA()>b.a.fA()) a=b.a;
    return a;
}

void main(){
    A a;
    B b1(2),b2(1);
    B b3=b1;
    b3.f5(b2)=a;
    b2=b1;
}
```

b) What is written on the screen when the given program is compiled and run? Explain. Disregard any possible run-time errors.

c) Some methods of class **A** and class **B** are not written properly. They do not cause any errors but they are not efficient. Re-write these methods properly.

d) Modify class **A** for counting objects of this class. When an object of class A is created the number of existing objects will be displayed on the screen.

Question2:

Design a class **Polynomial** to model polynomials. Each object of this class may be of different degree. The coefficients will be held in the dynamic memory (heap) and pointed by a pointer.
Example:

$5x^3 + 2x + 1$ Degree=3, Coefficients: 1,2,0,5

Polynomial class will have a **constructor**, which creates a polynomial for a given degree. The constructor will take coefficients via a pointer argument. If the user does not give coefficients they all will have **1** as initial value.
The class will also have a **copy constructor** and a **destructor**.
Polynomial class will contain following operators:
* Assignment operator (can be chained like a=b=c).
* **+** operator to add two polynomials. Return value is a polynomial.
* **==** operator to compare two polynomials. If they are equal return value is 1, and 0 otherwise.
* **()** operator will calculate and return the value of the polynomial for a given integer or float value.
 * **[ ]** operator is used to access (for reading and writing) the **i**th coefficient of a polynomial.

a) Write the complete class Polynomial in C++, using the object-oriented approach.
b) Write a main program that creates polynomials and invokes all methods of the class Polynomial.