İ.T.Ü. Computer Engineering Department 12.06.2000
Dr. Feza BUZLUCA
Çiçek ÇAVDAR

Object Oriented Programming Final Examination

Question 1:

A function, a main program and the screen output of the program are given below.

a) Write all necessary classes to generate this output in C++.

```
#include <iostream.h>
#include <string.h>

/* class definitions must come here */

void function(A &obja)
{
    obja.f();
}
int A::count;
void main()
{
 A::init();
 A a1("A1",1);
 A *ap1=new A("AP1");
 A *ap2=new B("BB","AP2",2);
 A a2=*ap1;
 cout<<"---------------------"<<endl;
 delete ap1;
 cout<<"---------------------"<<endl;
 function(*ap2);
 cout<<"---------------------"<<endl;
 delete ap2;
 cout<<"---------------------"<<endl;
 B b1("BB1","B1",10);
 B b2=b1;
 cout<<"---------------------"<<endl;
 function(a2);
 function(b2);
}
```

Screen Output:

```
There are 1 objects
There are 2 objects
There are 3 objects
There are 4 objects
---------------------
There are 3 objects
---------------------
BB
AP2
2
---------------------
B has been removed
There are 2 objects
---------------------
There are 3 objects
There are 4 objects
---------------------
AP1
0
BB1
B1
10
B has been removed
There are 3 objects
B has been removed
There are 2 objects
There are 1 objects
There are 0 objects
```

b) If we change only the main program as follows, which statements would generate compile errors, why?

```
int A::count;
void main()
{
 B b1("BB1","B1",10);
 A a1("A1",1);
 A a2=b1;
 B b2=a1;
 b1.init();
 a1.init();
 B *bp1=new A("BP1",2);
 A *ap2=new B("BB","AP2",2);
 A *ap;
 ap=&b1;
 function(&a1);
 function(ap2);
 function(*ap);
}
```

Question 2:

Courses given in a university can be grouped in three categories.

Category 1:

Grading: Homework 50%, Final 50%

Category 2:

Grading: Midterm 30%, Homework 30%, Final 40%

Category 3:

Grading: 1st Midterm 25%, 2nd Midterm 25%, Final 50%

Each course has a code-number (10 characters) and a credit coefficient.

Class definitions of the courses include:

Constructors to set the code-number and the credit coefficient of the course. The initial values of notes are negative numbers (it means this note is not given).

A function to set notes. The notes are taken from the user via keyboard. Notes can be given only once. It is not allowed to change the notes. This function returns 1 if the notes can not be set, otherwise 0.

A function to calculate and return the average of the course. A negative return value means the notes are not given.

A function to print all data involved with the course.

To define students, a class will be designed. The student class includes the name of the student and an array, which contains pointers to the courses of that student. Each student has 10 courses. This class also includes the following functions:

A constructor sets the name of the student and asks the user for the code-numbers and credits of the courses.

A function to enter notes of a course. The code-number of the course is taken as an argument.

A function to print all data and the average of courses of the student.

A function to calculate and return the graded average of the student (Average of all courses according their credits).

An operator function < , to compare the graded average of two students.

s1<s2 returns 1 if the average of s1 is greater than s2, returns 2 if the average of s2 is greater than s1, returns 0 if the averages are equal.

a) Write classes to define courses in C++, with necessary explanations.

b) Write the student class in C++, with necessary explanations.

c) Overload the << operator for the student class. The same operator can be used to print the name and the graded average of a student onto screen or to write the same data into a text file.

Write necessary parts of a main program to show how to print a student on the screen and to store it into a file (student.dat) using the << operator.