İ.T.Ü. 03.04.2000
Computer Engineering Department
Dr. Feza BUZLUCA
Çiçek ÇAVDAR

Object Oriented Programming 1st Midterm Examination

Question1:

a) What is written on the screen when the C++ program below is compiled and run? Explain.

```
#include <iostream.h>
class Aclass{
  private:
   int i;
  public:
   Aclass(){ cout << "Function 1" << endl;}
   Aclass(Aclass &);
   void operator=(Aclass);
   Aclass operator+(Aclass);
   ~Aclass(){ cout << "Function 5" << endl;}
};

Aclass::Aclass(Aclass &in_c)
{
  cout << "Function 2" << endl;
  i=in_c.i;
}

void Aclass::operator=(Aclass in_c)
{
  cout << "Function 3" << endl;
  i=in_c.i;
}

Aclass Aclass::operator+(Aclass in_c)
{
  cout << "Function 4" << endl;
  Aclass temp;
  temp.i=i+in_c.i;
  return temp;
}

void main()
{
 cout << "Start" << endl;
 Aclass a,b,c;
 cout << "Operation" << endl;
 a=b+c;
 cout << "End" << endl;
}
```

b) Implementation of `Aclass` has some drawbacks. Explain, how can be this class improved and re-write it more efficiently.

c) What is the output of the improved program?

Question2:

a) Write a class **Array** that models integer arrays. Each object of this class can have different lengths. The class must contain a pointer to integer numbers, and a variable, which stores the number of available (not used) spaces in the array.

   Array class will have a constructor, which creates an array of a given size, a destructor and a copy constructor.
   It will also have an assignment operator (can be chained like a=b=c), and operator < to compare number of empty spaces in two different array objects. This operator returns the difference of available spaces. If the result is negative, it means that the first array has less empty spaces. For example **–6** means that the first object has 6 available spaces less than the second object.
   The methods of the class are:
   * write function will insert an element to a given position of the array. Position is a number between 0 and (length-1) .
   * read function will return the value of an element from the given position of the array. After read operation, it is assumed that this space is available for further writes.
   * empty function will empty the array.

b) Write a class **Stack** that models stacks. It will use an object of Array class to store various stack elements. It will have a constructor, which creates an empty stack with the size of a given value (argument), a destructor and a copy constructor.
   It will also have an assignment operator (can be chained) and the operator < . This operator will compare available spaces in two different stack objects and will return the difference. For example **–6** means that the first object has 6 available spaces less than the second object.
   Its methods are:
   push: returns 0 if the stack is full, return 1 otherwise.
   pop: returns the element pulled from the stack. If the stack is empty it returns 0.
   EmptyStack: empties the stack.

c) Write a main program that creates two stacks of size 15. First push some elements to stacks. Program will print out the elements of the stack whose empty space is smaller then the other. If the stacks are equal or empty then related message will be print to the screen.