

Question #5 - (50 points)

Read the following given very carefully before start writing the code:

For the following problem, assume class **Date** has already been implemented (**DO NOT** write the class **Date**, just use it). The class **Date** includes methods that check for month, day, year and a method *const* print function that displays the date (the date is printed in the following format: MM/DD/YYYY)

You are requested to create the interface (header file) and the implementation of a new class called **ElectricalMachine** that contains:

- An **int** instance variable named **kind** that specifies the type of the machine (the int is 1 if the machine is a **generator**, 2 if it is a **motor**, the default value is 1)
- A **boolean** instance variable named **status** that identifies whether the machine is on (**true**) or off (**false**), default is **false**.
- A **float** instance variable named **speed** that specifies the speed of the rotation of the machine (default is 0.0).
- An instance variable named **serialNum** (**char***) that specifies the serial number of the machine (default is "S69-4569").
- A **Date** instance variable named **manufacturingDate** that specifies the manufacturing date of the machine (default 01/01/2000).

The new class should include appropriate **constructor** (s), **destructor**, **accessor** and **mutator methods** (get and cascaded set) for the first four instance variables, and a **friend print** function that prints the kind, speed, serial number, manufacturing date, as well as the **status** of the machine if it is on or off.

The class should also include a counter, **count**, of type integer, that keeps track of the number of objects created (this counter must be incremented whenever an object is created and decremented whenever an object is deleted), and a method **getCount()** that returns the number of objects created so far.

Demonstrate the use of **Composition** by using the class **Date**.

Write a driver program implementing the following steps.

- 1- Instantiate two dynamic objects **em1**, and **em2**, and initialize them with values of your choice, call the friend print function to display all the instance variables, and then display the number of objects created.
- 2- Instantiate a third object **e3** (not necessarily dynamic) with default values, and then demonstrate the use of cascading through set functions invocations.
- 3- Delete the object **em1** and then display back the number of objects.

to be written to the standard output when the following program is run?

```
#include<iostream>
using namespace std;

class Q1{
public:
    Q1( int a):x(a){};

    void modifyX( Q1 q)
    {
        q.x = 25;
    }

    int getX() const
    {
        return x;
    }
private:
    int x;
};

int main()
{
    Q1 q(10);
    cout<<"X = "<<q.getX()<<endl;
    q.modifyX( q );
    cout<<"X = "<<q.getX()<<endl;
}
```

Question #3 –(5 points)

What is wrong with the following code? Explain how to correct it .

```
#include<iostream>
using namespace std;

class Q6{
    public:
        int getX() const
        {
            return x++;
        }
    private:
        int x;
};

int main()
{
    Q6 q;
    cout<<q.getX()<<endl;
}
```

Question #4 - (5 points)
What is wrong with the following code? Explain how to correct it.

```
#include<iostream>
using namespace std;
class Q9{
    friend void modifyX(Q9 &, int):
    public:
        Q9(int a = 10):x(a){}
        int getX() const{
            return x;
        }
    private:
        int x;
};
void modifyX(Q9 &q, int a)
{
    q.x = a;
}
int main()
{
    Q9 q;
    cout<<q.getX()<<endl;
    q.modifyX( q, 20);
    cout<<q.getX()<<endl;
}
```

Question #6 - (30 points)

Create a class called Rational for performing arithmetic with fractions. Write a program to test your class. Use integer variables to represent the private data of the class (the numerator and the denominator). Provide a constructor that enables an object of this class to be initialized when it is declared. The constructor should contain default values in case no initializes are provided. Provide public member and or friend functions that perform each of the following tasks:

- a) Overload the Stream Insertion and Stream Extraction operators.
- b) Overload the + operator so as two objects can be added when the user type $r3 = r1+r2$.

Provide a driver that performs the following:

- 1- Create two Rational objects Rational r1, and r2
- 2- Input their instance variables from the keyboard using the overloaded stream extraction operator
- 3- Add r1 and r2 and store their result into and new rational number r3 using the overloaded + operator
- 4- Display the results of r3 and ~~r4~~ using the overloaded stream extraction operator.