Classes, objects. Static, const, mutable members of a class

Static Members of a class have to be defined and initialized. They are defined only inside the body of the class.

-static functions do not allow the "*this"* pointer

-in static functions we cannot access non-static members

-creating, initialization and the access to these members do not depend on the existence of the objects.

Example:

```
class Date
{
    int day, month, year;
    static Date today; // declare the static data member
public:
    // …
    static void initToday();
};

Date Date::today; // create the static data member
```

Const Members -const data cannot be modified;

-the syntax: *const data_type data_name;*

- const functions cannot modify the state of an object

-when the function is defined outside the body of the class, then the suffix const is required

-the syntax: *function_type function_name(list_of_parameters) const;*

Example:

```
class Date
{
    int day, month, year;
    static Date today; // declare the static data member
public:
    // const member function
    int getDay() const
     {
```

```
        day = 0 ; // ERROR: we're in const function
        return day;
    }
    int getMonth() const
    {
        return month;
    }
    int getYear() const
    {
        return year;
    }
};

int main()
{
    Date d;
    cout << d.getDay() << d.getMonth() << d.getYear();
    return 0;
}
```

Mutable Members –can be always mofidied, even in const functions!

-it is useful to use mutable in the case when we want to modify a value in a const function

-syntax: *mutable data_type data_name;*

Exercises:

1. **Write the necessary C++ code such that for the following function main the program will work:**

```
void main()
   {
   cout<<"///////// -----the Math class------\\\\\\\"<<endl;
      Math ob1(3,4);
      ob1.times();
      ob1.divide();

      Math ob2,ob3(5,5);
```

```
        ob2.times();
        cout<<"Add "<<ob3.add()<<endl;
    }
```

2. **Create a C++ program which will work for the following function main:**

```
void main()
{
   Person p1;
   p1.get_pers(); //will print the name, the city, gender,
                  //the height and how old is the person p1
   Person p2("Ana Maria","Deva","F",1.63,20);
   p2.get_pers();

   p2.travels();  //the user introduces if the person travels
                   // (Y/N), if 'Y' then it will print where
                   // exactly the person goes and if 'N'
                   //then it will print a message
   cout<<"The first person is dealing with
"<<p1.is_dealing_with()<<endl;
   p2.eats(); //will print some dishes (depending on the
              // chosen option) breakfast, lunch or dinner.
}
```

3. **Repare (add and modify) the following C++ program such that when calling the variable nb_of_objects in the function main will print on the screen the number of objects (as instances of the class Object):**

3.3

```
class Object
{
public:
  static int nb_of_objects;
      Object(int);
      Object(const Obiect&);
};
int Object:: nb_of_objects=0;
Object::Object(int a)
```

```
{
     Object::nb_of_objects++;
}
Object::Object(const Object& b)
{
     cout<<"Object: copy constructor "<<endl;
     Object::nb_of_objects++;
}
```

Hint: test the following programs (3.1 and 3.2) such that you can see the utility of the keyword static:

3.1 Example of static member: the effect

```
#include <iostream>

using namespace std;
void printstatic( int c )
{
static int nStatic;   // The value of nStatic is retained
// between each call of the function
   nStatic += c;
   cout << "nStatic is " << nStatic << endl;
}

int main() {
   for ( int i = 0; i < 5; i++ )
      printstatic( i );
}
Output:
nStatic is 0
nStatic is 1
nStatic is 3
nStatic is 6
nStatic is 10
```

3.2 The use in classes:

```
#include <iostream>

using namespace std;
class CMyClass {
public:
```

```
   static int m_i;
};

int CMyClass::m_i = 0;

int main() {
   cout << CMyClass::m_i << endl;
   cout << CMyClass::m_i << endl;

   CMyClass::m_i = 1;
   cout << CMyClass::m_i << endl;
   cout << CMyClass::m_i << endl;
}
```

Output:

```
0
0
1
1
```

HOMEWORK

1. Create a class Car, which has a color, engine, horsepower, fuel, year of manufacturing, type, places, other... Take into account the facts that: a car has one single wheel, only 4 tires, can be driven by a single. Create the corresponding OOP in C++.

2. (Optional) Create a program which contains: a class Student which has name, year, group,..., a class Courses which contains the classes which can attend one student and a class Rooms which contains the rooms where the courses can take place. Note that two ore more courses cannot take place in the same room at the same time. Print the informations about at least 3 students (what are the courses they attend, in which room, …) Use constructors, destructor, ...

3. Given:

```
class Date
{
     int d,m,y;
public:
```

```
     int day() const { return d; }
     int month() const { return m; }
     int year() const;
     //......
};
inline int Date::year() const
{
     return y++; //error: attempt to change member value in
const function
}
inline int Date::year()
{                          // error: const missing in member
function type
     return y;
}
void f(Date& d, const Date& cd)
{
     int i=d.year();//ok
    d.add_year(1); //ok
    int j=cd.year(); //ok
     cd.add_year(1); //error: cannot change value of const
cd
}
```

Repare the errors!!

4. (Optional) For the following class String write the corresponding constructors, other methods and create some objects.

```
class Sring
{
     enum {DIM=100};
     int n;
     char s[DIM];
public:
     //constructor
   };
```

Read chapter 10 from the book „The C++ Programming Language", Bjarne Stroustrup 3rd Edition!!!!!!