The objective of this lab is to:

Understand and practice static data members, static member functions, this pointer, and copy constructor.

Instructions!

1. Please follow the dress code before coming to the lab. Keep your student identity cards with you.
2. This is an individual lab, you are strictly **NOT** allowed to discuss your solutions with your fellow colleagues, even not allowed asking how is he/she is doing, it may result in negative marking. You can **ONLY** discuss with your TAs or with me.
1. Strictly follow good coding conventions (commenting, meaningful variable and functions names, properly indented and modular code.
2. Save your work frequently. Make a habit of pressing **CTRL+S** after every line of code you write.
3. Beware of memory leaks and dangling pointers.
4. Create separate class header file (.h) and class definition file (.cpp) for each task.
5. Make member function `static` where appropriate.

Task 01: [10 Marks]

How many times is the copy constructor called in the following code:

```
class Widget
 {
        public:

        //Constructors and destructors exist here...

        Widget f(Widget u)
        {
               Widget v(u);
               Widget w = v;
               return w;
        }
};

int main()
{
   Widget x;
   Widget y = f(f(x));
}
```

Please note that you are strictly not allowed to run this code or similar on your compiler. Observe the code carefully and answer the question. Anyone caught writing this code on the compiler will be given a zero.

Task 02: [10 Marks]

Implement a *Time* class. Each object of this class will represent a specific time of day, storing the *hours*, *minutes*, and *seconds* as integers.
Include the following in the class:

- All constructors (default constructor, parametrized constructor, copy constructor)
- destructor
- Access and mutators for private data members
- a function *advance(int h, int m, int s)* to advance the current time of an existing object
- a function *reset(int h, int m, int s)* to reset the current time of an existing object

- A *print( )* function to print date in dd/mm/yyyy format.

Write a driver program to show the working of your class.

Note: *Implement proper checks for input validation i.e. months (between 1-12) and days (1-30 or 1-31 depending upon month) etc.*

Task 03: [10 Marks]

Implement a class *ObjectTracker* that can track how many objects of a class are present in memory. It should increment the object counter when a new instance of the object tracker is created. It should decrement the counter when any object is removed from the memory.

Implement all constructors, destructors, and necessary accessors and mutator functions. The user should be able to get the number of active objects of the class.