The objective of this lab is to:

Understand and practice concept of classes, objects, constructors, and destructors.

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.

Task 01: [15 Marks]

1. Design simple a class called *Date*. The class should store a date in three private integer data members: month, day, and year. Write a default constructor and appropriate mutator functions for all the data members. There should be three more member functions to print the date in the following forms:

   12/25/2012
   December 25, 2012
   25 December 2012

   Demonstrate the class by writing a complete program implementing it.
   Input Validation: Do not accept values for the day greater than 31 or less than 1. Do not accept values for the month greater than 12 or less than 1.

2. Write a class named Employee that has the following private member variables:

   Name: A *string* that holds the employee's name.
   idNumber: An *int* variable that holds the employee's ID number.
   Department: A *string* that holds the name of the department where the employee works.
   Position: A *string* that holds the employee's job title.

   The class should have the following constructors:
   - A constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee s name, employee s ID number, department, and position.
   - A constructor that accepts the following values as arguments and assigns them to the appropriate member variables: employee s name and ID number. The department and position fields should be assigned an empty string (" ").
   - A default constructor that assigns empty strings (" ") to the name, department, and position member variables, and 0 to the idNumber member variable.

   Write appropriate mutator functions that store values in these member variables and accessor functions that return the values in these member variables. Once you have written the class, write a separate program that creates five Employee objects to hold the following data.

Name	ID Number	Department	Position
Susan Meyers	**47899**	**Accounting**	**Vice President**
Mark Jones	**39119**	**IT**	**Programmer**
Jay Rogers	**81447**	**Manufacturing**	**Engineer**
Lionel Messi	**52362**	**HR**	**Manager**
Roger Federer	**54124**	**Finance**	**Auditor**

Task 02: [15 Marks]

Create a class *Counter* to act as a counter variable in a program. Implement the following class interface:

```
class Counter
{
private:
	int i;			// an integer to hold the value of a counter.

public:
	Counter();	// initialize counter to zero.
	Counter(int a_val);	// initialize counter with given parameter value.
	~ Counter();	// destructor should display message “Object is destroyed”.

	void setCounter(int a_intgr); // set counter value to given parameter value.
	void reset();		// reset the counter to 0.
	int getCounter () const; // return the counter value.
	void increment(); // increment counter by 1.
	void increment(int a_val); // increment counter by parameter value.

	void decrement();	// decrement counter by 1.
	void decrement(int a_val); // decrement counter by parameter value.

};
```

In *main( )* function take length and width from user and print a rectangle of the given size using *Counter* object in loops. Below is an example of how you should use the *Counter* object.

```
for (Counter c; c.getCounter() < 10; c.increment())
{
	. . .
}
```

Now in a separate function print all numbers in a user defined range. Call this function in *main( )* to demonstrate its working.