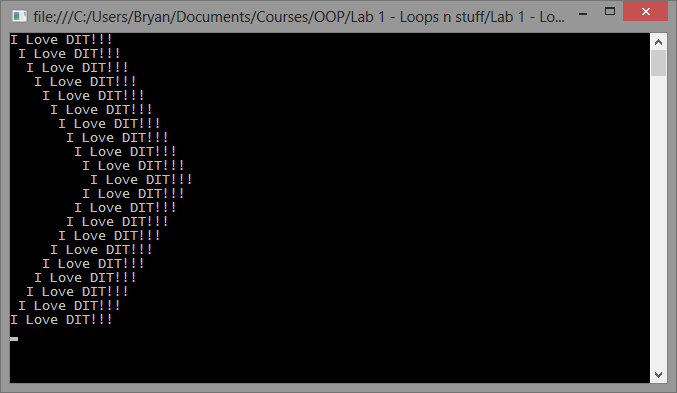
**Lab 1**

The purpose of this lab is just to get you back into programming by writing some simple programs in C# using Visual Studio.

**Program 1**

Write a C# program to generate the following:



**Program 2**

Write a while loop to print out the Fibonachi series up to 200. For anyone who has not seen (or read) the DiVinci code, it should look like this:

0

1

1

2

3

5

8

13

21

34

55

89

144

Notice that the first two numbers are 0 and 1 and every subsequent number is the sum of the previous two numbers.

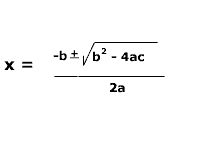
**Program 3**

Write a program to calculate the roots of a quadratic equation using the quadratic formula.

The roots of a quadratic equation in the form:

Ax2 + Bx + C

Are given by the quadratic formula



If you are not sure how to do this on pen and paper, read [this article which explains how to do it](http://www.purplemath.com/modules/quadform.htm). For practice, calculate the roots of the following formula on pen and paper FIRST:

**2*x*2 – 4*x* – 3 = 0**

**You should get the answer:**

2.581139

-0.581139

You probably all learned how to use this in secondary school :-), but if you need some revision, [**read this article first**](http://www.purplemath.com/modules/quadform.htm)**.**

**To calculate a square root in C#, use Math.Sqrt(number)**

**It returns a double and you must cast the return value to be a float:**

**(float) Math.Sqrt(number)**

Lab 2

The problem:

Google have developed a new algorithm for rating how funny a youtube video is by counting the number of times the letters “lol” appear in the comments of the video.

Create a new C# Console Program (store it on your U: Drive)

Write a c Sharp method:

static int lolCount(string s)

That returns the number of times that the substring "lol" appears in the string s.  The following code should be in your **Main** method:

  // Part A

            string s1 = "Wow that was so funny lol.";

            Console.WriteLine("The lolcount of " + s1 + " is " + lolCount(s1));

            string s2 = "lol!! I laughed so rhard! lol";

            Console.WriteLine("The lolcount of " + s2 + " is " + lolCount(s2));

            string s3 = "U r lame. lol lol lol";

            Console.WriteLine("The lolcount of " + s3 + " is " + lolCount(s3));

            string s4 = "lollollollollollollollol!!!!!";

            Console.WriteLine("The lolcount of " + s4 + " is " + lolCount(s4));

It should print:

The lolcount of Wow that was so funny lol. is 1

The lolcount of lol!! I laughed so rhard! lol is 2

The lolcount of U r lame. lol lol lol is 3

The lolcount of lollollollollollollollol!!!!! is 8

You can use the IndexOf method in your solution. Please note!!

You can pass a second parameter to IndexOf, which is the index to start searching from in the string. For example:

s.IndexOf("AAA", 0); \\ Start at the begining of the string

s.IndexOf("AAA", 5); \\ Start at the 5th character

Part B

In this lab we will create a program that can encrypt and decrypt strings using a simple algorithm.

The algorithm we will use is the following:

To encrypt a string, we reverse it and add 1 to the ASCII code for each character. For example, if we take the word:

bryan

When we encrypt the word it becomes:

obzsc

Notice that the “n” in bryan has become has “o” in the encrypted string and we have switched the order.

To decrypt a string we do the opposite. We reverse the order again, but this time subtracting 1 from the ASCII code for each character in the string.

To solve this lab, you should do the following:

1. Declare a string and assign a value to it.
2. Write a method to encrypt the string using the algorithm described above.
3. Call the method with the string you declared
4. Print the encrypted string out
5. Write a method to decrypt the string and make sure you get the original string back again.

Here is the code that needs to appear in your Main method:

     // Part B

            string s = "The desired result";

            Console.WriteLine("Original string: " + s);

            Console.WriteLine("Encrypted string: " + encrypt(s));

            string en = "umvtfs!efsjtfe!fiU";

            Console.WriteLine("Decrypted string: " + decrypt(en));

 This should print:

 Original string: The desired result

 Encrypted string: umvtfs!efsjtfe!fiU

 Decrypted string: The desired result

 Here are some encrypted strings for you to decrypt:

 /ztbF!tj!hojnnbshpsq!/hojnnbshpsq!fwpM!J

 "tlobU!///tj!fhofmmbiD!eobmfsJ!BOY!tsbfz!tjiu!gp!fnfiu!fiU

 tldpS!UJE

 fujsx!pu!fwbi!uoeje!vpz!fojm!fiu!tj!fupsx!sfwf!vpz!fepd!gp!fojm!utfc!fiU

Lab 3

Before attempting this lab, take the time to download the example program from the class this week and study it carefully

**Part 1:**

Create a struct Point, with private x and y fields of type float public properties that allow access to the private fields

Make a struct called Circle which has a public field of type Point to represent the centrepoint and a private field for the radius, (with a public property Radius)

Make a READONLY property for the Circumference and Area. Use a value of 3.14 for pi.

Write some code in your Main method to test out your struct

**Part 2:**

**Men** should drink no more than 21 units of alcohol per week. Between 16 and 21 units is considered the “warning zone” and in in excess of 21 units is “dangerous”. Below is a table giving the number of units of alcohol consumed by a group of men in the past week:

|  |  |
| --- | --- |
| Pat | 12 |
| Francois | 18 |
| Bryan | 10 |
| Derek | 25 |
| Mario | 35 |
| Shane | 12 |
| Millhouse | 0 |
| Stephen | 0 |
| Barry | 10 |
| Liam | 40 |

Write a C# program using Visual Studio that stores the information above in an array of structs and prints out a table of the men with a rating of their alcohol consumption formatted as follows:

Pat     12      SAFE  
Francois        18      WARNING  
Bryan   10      SAFE  
Derek   25      EXCESSIVE  
Mario   35      EXCESSIVE  
Shane   12      SAFE  
Millhouse       0       SAFE  
Stephen 0       SAFE  
Barry   10      SAFE  
Liam    40      EXCESSIVE

In your program DO NOT have the user have to type in the information using Console.ReadLine every time it is run, instead, hard code these values in your program. Also you should calculate the alcohol consumption category using if statements. Use structs, arrays, ToString, foreach in your solution. DO Not just hard code the printing of the table above.

Lab 4 was the lab test

**Lab 5**

Start a command shell in Windows and type:

u: <PRESS ENTER>

git clone <https://github.com/skooter500/OOP-Labs---Example-Programs>

Navigate to the folder that got created and open the Visual Studio solution file

DT228 Example Programs.sln

This will open up a Visual Studio solution with all of the example programs we have worked on so far in this course including lab solutions & the lab test solution. You can change the solution that you are working on, select the project and right click on it and choose Set As Startup project.

For this lab, I want you to start on the Expenses project we did in the lab test. Take the time to study the project carefully to make sure you understand what each line of the project does.

**Put comments in the code to explain what each line does.**

The aim of this lab is to modify the program in the following way:

Modify the program so that the searches are case insensitive. In other words, lab and LAB should both match Lab.

Now modify the program so that it initially prompts the user with the following:

Enter C to search by first 3 letters of the constituency or P to search by party or \quit" to quit"

If the user then presses C, the program should prompt the user as follows:

Enter the first three letters of a constituency or “quit” to quit

The user should then be able to enter the first 3 letters of a constituency and to print the expenses list for any matched constituency. If the user were to have presses P, the program should work as normal. Below is a sample interaction:

Enter C to search by first 3 letters of the constituency or P to search by party

or "quit" to quit

p

Enter the name of a party or "quit" to quit

pbp

Boyd Barrett, Richard PBP Dun Laoghaire 0 31865.51 31865.51

Collins, Joan PBP Dublin South-Central 1811.05 31865.51 30054.46

Total claimed for Party: pbp was 61919.97

Minimum claimed from Party: pbp was 30054.46 by Collins, Joan

Maximum claimed from Party: pbp was 31865.51 by Boyd Barrett, Richard

Enter C to search by first 3 letters of the constituency or P to search by party

or "quit" to quit

c

Enter the first three letters of a constituency or "quit" to quit

lou

Adams, Gerry SF Louth 0 48643.52 48643.52

Adams, Gerry SF Louth 0 48643.52 48643.52

Fitzpatrick, Peter FG Louth 0 47893.52 47893.52

Kirk, Seamus FF Louth 0 37490.62 37490.62

Nash, Gerald Lab Louth 0 45478.95 45478.95

O'Dowd, Fergus FG Louth 1182.45 17484.58 16302.13

Total claimed for Constituency: lou was 244452.3

Minimum claimed from Constituency: lou was 16302.13 by O'Dowd, Fergus

Maximum claimed from Constituency: lou was 48643.52 by Adams, Gerry

Enter C to search by first 3 letters of the constituency or P to search by party

or "quit" to quit

quit

Goodbye