SIT102 - Practical 1

Introduction to Computer Programming

Welcome to the SIT102 unit.

In this practical we will introduce the unit web site, discuss how the practicals will be conducted and start using Visual Studio to create your first program.

Your tutor is assigned to your tutorial to help your with your studies or if you are an Cloud Online student, please feel free to email call your unit chair.

Please ask your tutor any questions you may have as it is important that you keep up to date with your study.

CloudDeakin

To begin with, spend some time becoming familiar with CloudDeakin and how the unit material is organised.

Open *Discussion* and read any messages. Ensure you read the postings regularly as I will be posting answers to questions, posting notices regarding the unit and other important information as required.

You should be visiting CloudDeakin at least once a day to ensure you are up to date with the unit material.

I will try to email you regularly with any important information and also to inform you of any new material uploaded to CloudDeakin.

Walk through of Visual Studio

This semester we will be developing C# code using Visual Studio 2013 or 2015.

On campus students can use the version installed on the lab computers or if you need to use a copy of Visual Studio at home, you can download a version from the Microsoft website.

Go to http://www.microsoft.com/express/Downloads/ and download an Express edition of Visual C# 2015. You may have to download additional software in order to get the software running.

Alternatively you can purchase a copy or go to the library and search for textbooks on C# that have a CD attached as they sometimes provide an Express version.

You may use Visual Studio 2008 or 2010 if you have a copy.

If you have not already done so, read the textbook from page 1-11 as it provides some useful information regarding the background to the C# language.

Find and open Visual Studio on your computer. You will be using Visual Studio to write, compile, debug and test your programs, so it is important you become familiar with the environment.

Read and follow the 'Creating, compiling, and running your first program' section of chapter 1 (page 11 - 21). This will walk you through the use of Visual Studio to write, compile and run a small console application.

As with all software, the first time you use the software it can be very confusing and difficult. Visual Studio has many windows, new terms, functions and concepts. Most are not important to you at this stage but over time the environment will be less confusing and you will start to focus on the coding. It is important to get this first program running so if at first it does not work, back track and try again.

If you need help understanding the Visual Studio environment or getting your first program to work then ask your tutor or post a message to Student Talk.

It is rare that as a programmer you are able to write and compile a program without errors. The Visual Studio environment is equipped with error checking and reporting tools that are designed to help you discover and fix your errors.

The following code is the Hello World program you should enter into the Visual Studio project and then compile and run program.

Type the code in rather than copy and paste it as it important you use familiarise yourself with this process. Note the syntax highlighting that occurs.

You will also note the code is indented to reflect structure and depending on your Visual Studio setup, the indenting will be 2 to 4 spaces for each indent.

```
// This is a simple C# program.
// Call this program Example.cs.

using System; // compulsory library
class Example // define class Example
{
        // A C# program begins with a call to the FREE Main() with no return data and no input data
        static void Main()
{
                Console.WriteLine("C# gives you programming power.");
        } // end of Main program
} // end of Example class
```

Review the line by line description of the program from the textbook (pages 18-21) so you understand what all the code means. A more detailed description of the code will be provided in the lectures slides.

Program code must be correct for the code to compile. Introduce an error in the code by deleting a semi colon or removing a quote character then recompile the program and note the errors that are presented.

Fix the errors and recompile until the program is correct.

If you feel you can move forward, try the Small Variation of the program described in the textbook pages 21 - 37.

```
using System; // example pp22
class UseVars // challenge change to floating type and prompt user the dimension values
{
	static void Main()
	{
		// Declare variables
		int Length;	// this declares a variable
		int Width;	// this declares another variable
		int Area;	// this declares third variable

		// Assign Length the value of 9
		Length = 9;
		// this display the current value of Length
		Console.WriteLine("Length contains: " + Length);
		// Assign Width the value of 7
		Width = 7;
		// this display the current value of Width
		Console.WriteLine("Width contains: " + Width);
		// Assign area the product of Length and Width
		Area = Length * Width;
		// Display output
		Console.Write("Area contains Length * Width: ");
		Console.WriteLine(Area);
	}
}
```

Version C

```
using System; // example pp22
public class UseVars // challenge change to floating type and prompt user the dimension values
{
	public static void Main()
	{
		// Declare variables with Length = 9 and Width = 7
		int Length = 9,  Width = 7, Area;  // this declares the variable Area
		// Display current variable value
		Console.WriteLine("Length contains: " + Length);
		Console.WriteLine("Width contains: " + Width);

		// calculate the product of Length and Width and assign to Area
		Area = Length * Width;

		// Output the Results
		Console.WriteLine("Area contains Length * Width: " + Area);
	}
}
```

Version D

```
using System; // example pp22
public class UseVars // challenge change to floating type and prompt user the dimension values
{
	public static void Main()
	{
		// Declare variables Length = 9 and Width = 7 and area stores product of those two
		double Length = 9,  Width = 7, Area = Length * Width;
		// Output the Results
		Console.WriteLine("Length contains: " + Length +  "\nWidth contains: " + Width+
					"\nArea contains Length * Width: " + Area);
	}
}
```

Version HD: variable to store updated data – here we use fixed value no need for data variables

```
using System; // example pp22
public class UseVars // challenge change to floating type and prompt user the dimension values
{
	public static void Main()
	{	// Output the Results
		Console.WriteLine("Length contains: " + 9 +  "\nWidth contains: " + 7+
					"\nArea contains Length * Width: " + 9*7);
	}
}
```

```
using System; // example pp25
public class IntVsDouble
{
	public static void Main()
	{
		// Declare variables
		int iVar;		// this declares the variable iVar
		double dVar;	// this declares the variable dVar
		// Assign/store value to variable
		iVar = 10;
		dVar = 10.0F;  // what is F means?
		// calculate the division by 4
		iVar = iVar / 4;  // show short hand version
		dVar = dVar / 4.0; // show dVar / 4 explain why?
		// Output the Results
		Console.WriteLine(); // show \n do not need this
		Console.WriteLine("iVar after division: " + iVar);
		Console.WriteLine("dVar after division: " + dVar);
	}
}
```

Ver 1: declare and set variable

Ver 2: /=

Ver 3: can't have 10 /= 4 in declare => declare with 10/4 and 2 console (int div int # float div float)

```
using System; // example pp28 - 29
```

```
public class IfDemo
{
	public static void Main() // show cut-paste reduce typing and errors
	{
		int a = 2, b = 3, c;

		if ( a < b) Console.WriteLine("a is less than b”);
		if ( a == b) Console.WriteLine("you won’t see this”); // discuss = rather ==
		if ( a == 2) Console.WriteLine("a contains the value 2”);
		if ( a == 19) Console.WriteLine("you won’t see this”);
		if ( a == b-1) Console.WriteLine("a equal b -1”);

		c = a – b; // discuss int c = a –b; and do not need to declare c in the 1st line
		Console.WriteLine("c contains -1”);
		if ( c >= 0) Console.WriteLine("c is non-negative”);  // c > -1 is the better on
		if( c < 0 )   Console.WriteLine("c is negative”);

		Console.WriteLine();

		c = b – a;
		Console.WriteLine("c now contains 1”);
		if ( c >= 0) Console.WriteLine("c is non-negative”);  // c > -1 is the better on
		if( c < 0 )   Console.WriteLine("c is negative”);

		Console.WriteLine();
	}
}
```

Discuss else with if and >= with > -1 for positive or < 0 for negative

```
using System; // example pp34
public class FtoCTable
{
	public static void Main() // show cut-paste reduce typing and errors
	{
		for(int f = 0, counter = 0; f < 100; f++, counter++)
		{
			if(counter % 10 == 0) Console.WriteLine();
			Console.WriteLine(f + " degrees Fahrenheit is " + (5.0 /  9.0 * (f - 32.0))
					+ " degrees Celcius.”);
		}
	}
}
```

By the end of this week all students should try to complete the program to product the output of the test 1 by the Console.WriteLine() commands. Next week will complete declare data, get input & calculation for test 1.