COMPUTER PROGRAMMING LAB
CSE 4272

Lab 01

Prerequisites:

1. Download CodeBlocks from http://www.codeblocks.org/downloads/26
2. Make sure you are downloading "**codeblocks-17.12mingw-setup.exe**" exactly this version for windows.
3. After downloading, go through normal installation procedures.

Code & Run a program:

1. Open CodeBlocks
2. Create an empty file using **CTRL** + **SHIFT** + **N**
3. Before you start coding it is always a good practice to save the file so that you get color codes to help with your coding
4. To save the file, use **CTRL** + **S**
5. While saving the file, do ensure that your file name is saved with a **.c** extension (i.e. test.c)
6. After creating a .c file and saving it, you can start coding in the text editor.
7. Write down the following program code (write it down exactly as it appears, do not miss any space or semicolon and also don't add any extra spaces)

```
#include<stdio.h>

int main()
{
    printf("Hello World");
    return 0;
}
```

8. After writing down the code you can build (compile) and run (execute) the code using F9
9. A console will be opened with the output (result) of your program

What you have compiled is the traditional "Hello World" program in C. This program uses (that is, it 'includes') code from the C-language 'standard input/output library, `stdio`, using this statement:

```
#include <stdio.h>
```

The code that starts with the name `main` is the ‘main function’ – in other words, it is the first bit of code that runs when the program runs. The function name is followed by a pair of parentheses.

The `main` function is preceded by `int`. This shows that the function returns an integer (a full number) when it finishes running. The number `0` is returned in the last line of the function:

```
return 0;
```

This return value is unlikely to be of any significance in your programs and, for the time being at any rate, you can ignore it. By tradition, a value of `0` just means that the program ran without any errors. Any other value might indicate an ‘error code’.

Cautions:

1. Every statement must end with a semicolon ;
2. You cannot rerun the code unless you have closed your previous console output

Things to be discussed:

- `printf()` and `puts()` function
- Variable
- Arithmetic Operators
- scanf() function