

Programming 1: Tutorial 3

Introduction

For this tutorial you are going to produce a few basic programs. You can either over write previous question solutions with new ones or you can have a project for each of the exercises – don't forget to make them blank applications.

Hopefully you remember the basic layout of a C++ program?

If not here is a refresher, basic C++ programs like the ones you have been asked to do here have the following layout:

```
#include <iostream>
// include statements up here

int main()
{
    // block of code between the { and } brackets

    return 0;
}
```

It may also help if you go through a design stage for each exercise where you plan and puzzle through the logic before actually coding.

There is also a sample code at the end of the tutorial that demonstrates assignment.

Exercise 1: Data Types Byte Size

For this I want you to write a program that prints out the size in bytes of all the basic data types.

The program output should be a table of the form:

Type	Size
short int	2
int	4

To do this you will need to use the **sizeof** operator. This returns the size in bytes, of a type or an expression and can be used in the following way with data types to determine there size in bytes.

Examples: **sizeof(int)** **sizeof(long)** **sizeof(float)**

You don't need to define any variables in this program, as you can use **sizeof(type)** expressions directly in the stream output statements (cout).

Although you can assign the value return from the **sizeof(type)** expression to a variable if you desire.

The **"\t"** creates a tab, **"\n"** creates a new line.

You can reduce the amount of typing you have to do, by copying and pasting existing output statements and modifying the copies to work for other types.

Exercise 2: Arithmetic

There are a few things that the C/C++ programmer has to be aware of when using arithmetic.

Write a program that performs the following:

2 + 2
4 / 2
4 * 2
5 - 2
2 / 3

And prints the results to the screen

It is important to note that you can perform an arithmetic operation on the same line as the cout << statement.

Exercise 3: New spin on the Old

For this exercise we will build on last weeks how "old are you" exercise.

The program should ask how old you are, as per the previous exercise, it should then print out how many days you have lived and print that to the screen.

There are roughly 365.242199 days in a year.

Linking Error

To start of with this week we are going to look at the second type of error – Linking Errors

Linking Errors

- Error is typically in the form of the declaration, implementation or call of a function
- Error may also result from not including the correct header file
- Sometimes the compiler fails to link and the error is not your doing
- Error is detected by the linker (after an object file has been produced)
- Linker cannot correct error so an executable is not produced
- Linker prints error message but continues link analysis

The Most Common Linker Error

The most linker error is likely to result from selecting the wrong project at start up. This error comes about either because Visual Studio bases the projects dependencies and libraries initially on what project you select so selecting the wrong one can result in your code not compiling and a linker error being produced, or because when you create it not blank it defines a `main()` for you in a file so when you code your own you generate an error.

If you obtain the following error, or one similar:

```
fatal error C1010: unexpected end of file while looking for precompiled header directive
```

Copy your code and start a new project with the correct settings, paste your and recompile. If you are still getting the error its time to look for another solution.

To see an example of a linker error - start any Win32 Application, rather than a Win32 Console application. Copy one of you working exercises, or just write a basic hello world program, and you should receive a linker error like this:

```
LNK2001: unresolved external symbol _WinMain
```

This website provides some handy hints on errors and why they are being generated: <http://averia.unm.edu/VisualCPPErrorMessages.html>

Sample Programs

Assignment

```
// A short program with examples of assignment statements
#include <iostream.h>

int main()
{
    //Data declarations
    int i, j, k, m, n, surprise;

    float pi, c, z;

    //Assignment Statements
    i = 1;           // i is assigned the value 1

    j = 2;           // j is assigned the value 2

    k = 13/4;        // The result of 13/4 is 3 which is assigned to k

    i = i + j;       // The result of i + j is 3 which is assigned
                    // to i (the old value of i is overwritten.
    m = i * k;       // The result of i * k is 9 which is assigned
                    // to m
    n = m % j;       // The remainder of m divided by j is 1 which
                    // is assigned to n
    pi = 3.142;     // The value 3.142 is assigned to the float pi

    z = pi + 1;     // The value of pi + 1 is 4.142 which is
                    // assigned to z
    surprise = z;   // What happens when a floating point number is
                    // assigned to an integer identifier?

    return 0;
}
```

If you wish to check the output of the variables, add `cout` statements.