Anglia Ruskin University

Software Implementation MOD002702

Module Leader: Dr. George Wilson

George.Wilson@anglia.ac.uk

Room: COM307

Module Tutor: SAMs (Trinidad) staff

Anglia Ruskin University

Programming language: C#

Textbooks:

Miles, R. 2010. C# Programming. University of Hull. free

Sharp, J., A. 2012. Microsoft Visual C# 2012 step-by-step. Redmond, Washington. McGraw-Hill. comprehensive

Assessment:

100% coursework (one multi-phased element)
i.e 10% in-class exercises, 90% major assignment

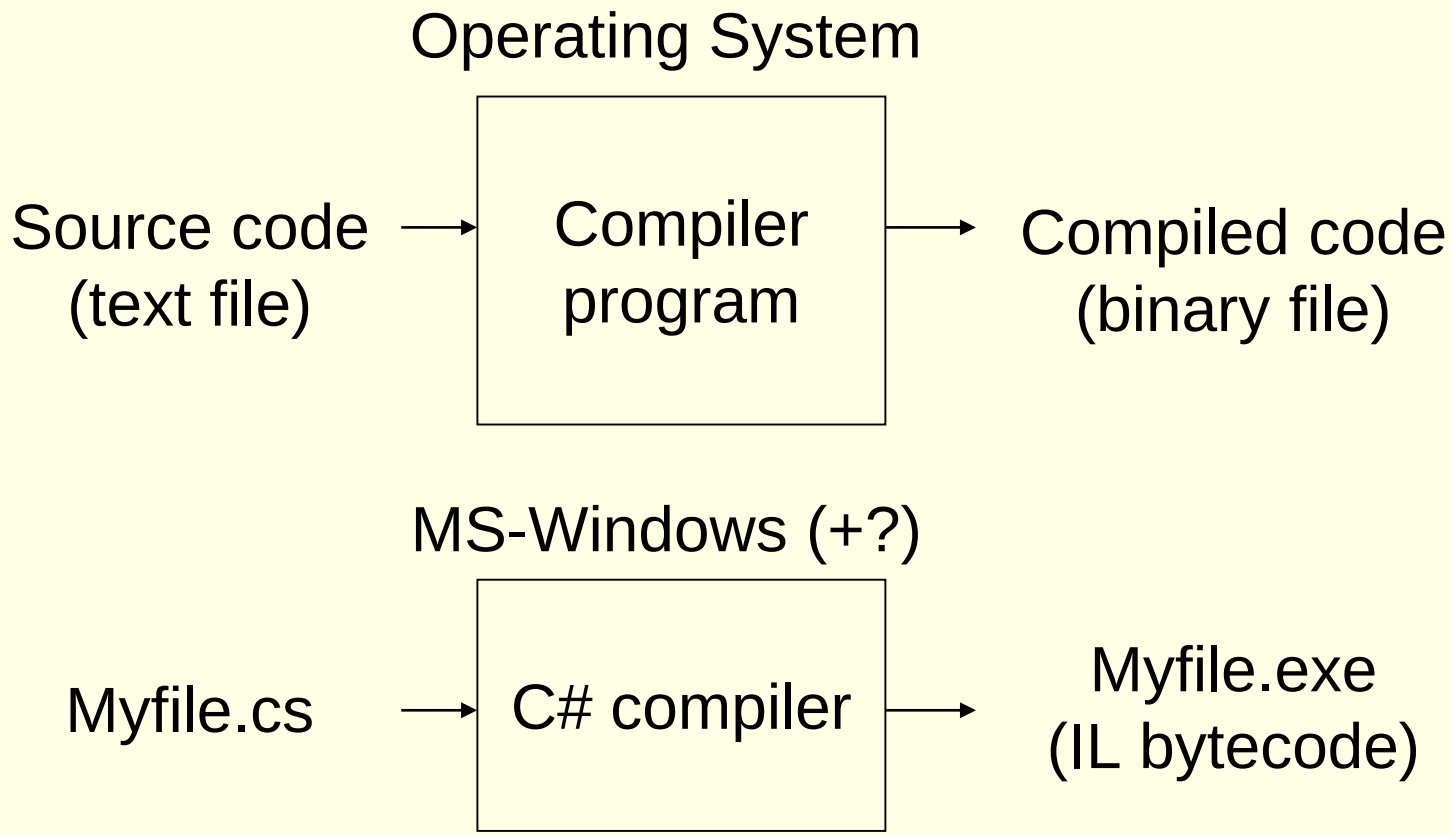
Anglia Ruskin University

Why study C#?

- Object-oriented (modern way of coding)
- Strongly typed (enforces good practice)
- Large standard library (support for networking, databases, GUIs)
- Relatively easy to transfer to other languages (eg Java, C/C++)
- Part of .NET framework

Anglia Ruskin University

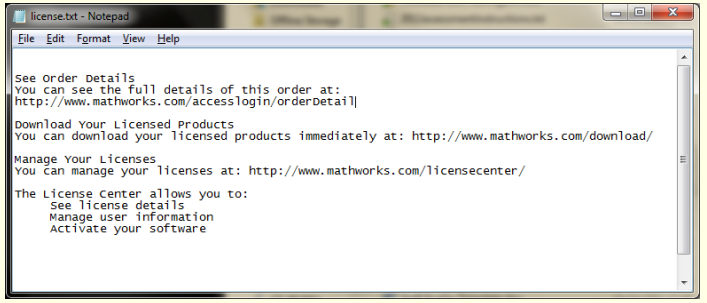
The compilation process

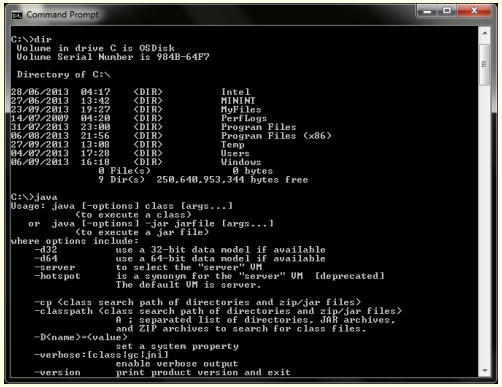


Anglia Ruskin University

To develop programs; we normally need at least>

1. Text editor eg Notepad
2. Compiler – a program to convert source code text into binary instructions
3. Console window – issue text commands eg to compile a program





Anglia Ruskin University

To simplify program creation programmer's usually use an

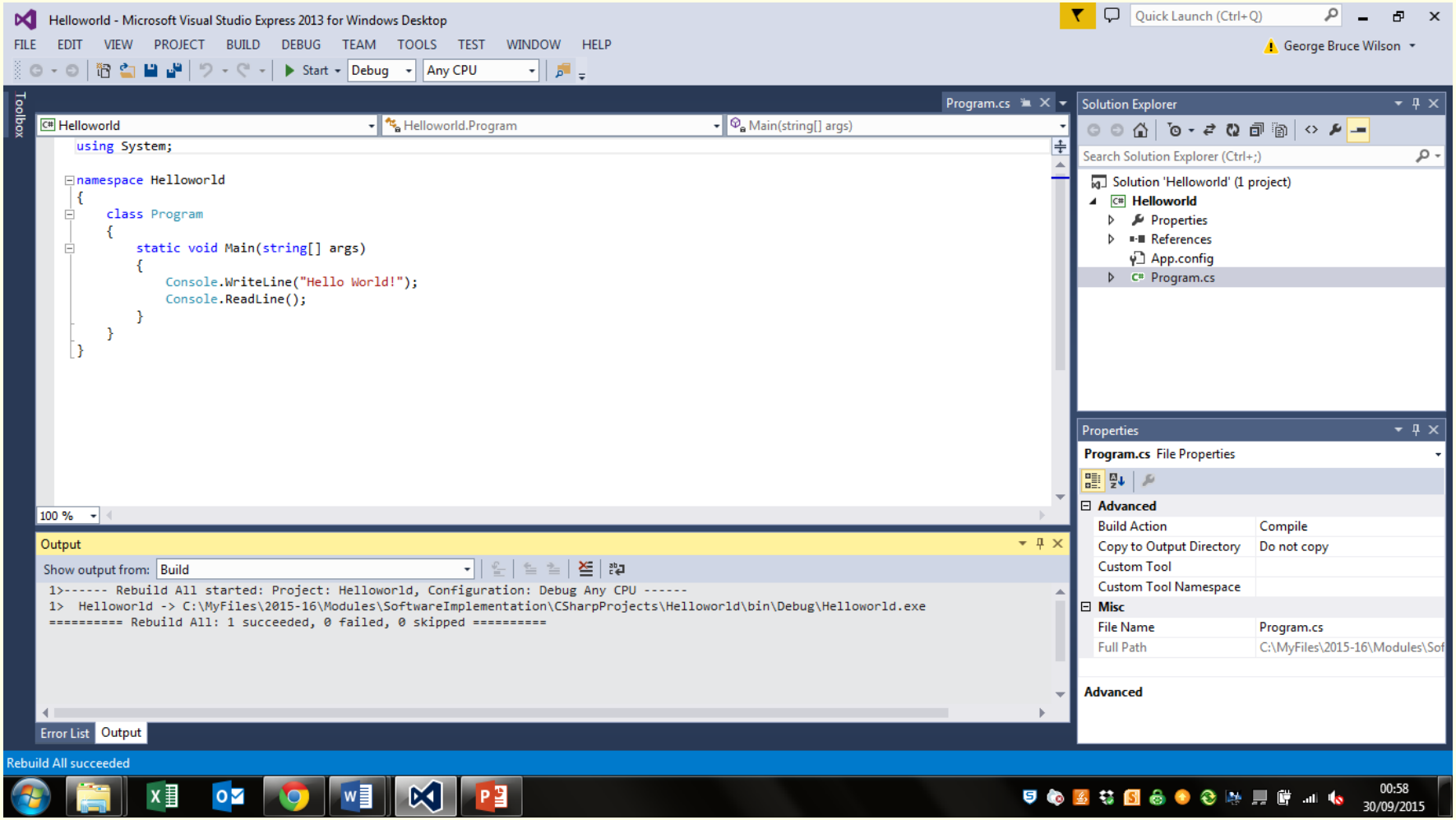
Integrated Development Environment (IDE)

An IDE is a utility to aid program development

- ***Editor***
- ***Link to compiler***
- ***Syntax highlighting and 'pop-up prompts'***
- ***Debugger***
- ***Aids file management***

C# - IDEs include Visual Studio, XNA, Mono

Anglia Ruskin University



Anglia Ruskin University

C# data types

A ***type*** specifies the *kind of data*

- **Primitive types** - basic data
- Complex(reference) types - classes

Examples of primitive types:

32.67104 104 ‘k’ TRUE

Anglia Ruskin University

Some common C# primitive data types

Type	Storage and meaning
int	32-bit integer (-2147483648 to +2147483647)
double	64-bit floating point
bool	8-bits (1-byte) :true or false
char	16-bit character (unicode)

Arithmetical operator symbols: usual rules of arithmetic apply>

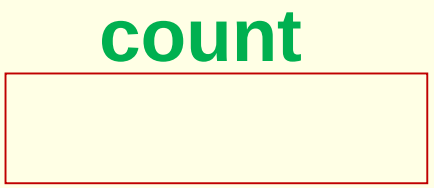
brackets **( )** multiply ***** divide **/** add **+** subtract **-**

Anglia Ruskin University

The differences between type - variable - value

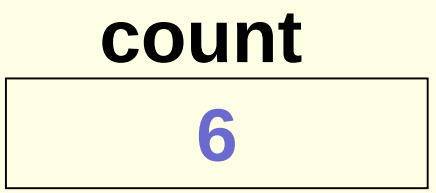
type variable

Declaration: **int count;**



Assignment: **count = 6;**

value



Combined declaration and assignment

int count = 6;

Semi-colon defines the end of a statement

Anglia Ruskin University

A simple C# program

```
using System;                  // ensures availability of resources

namespace TConv       // ensures common area of memory
{
   class TempConv
   {
      static void Main(string[ ] args)     // where execution starts
      {
         double fahrTemp, celTemp;
         string input;                          // string is a 'class' type
         Console.WriteLine("Enter a temperature in degrees F >");
         input = Console.ReadLine();
         fahrTemp = double.Parse( input );    // converts string to double
         celTemp = (fahrTemp - 32.0) * 5.0/9.0;    // conversion
         Console.WriteLine("The temperature in degrees C is " + celTemp);

      }  // brackets define blocks of code and are indented
   }
}
```

Anglia Ruskin University

Conclusions

- **A program is first written in a text file (source code)**
- **Program development undertaken using an IDE**
- **C# programs start at Main()**
- **Data represented as type - variable - value**
- **C# programs are based on units or ‘classes’, with content organised into blocks using curly brackets { }**
- **Source code ‘compiled’ into binary/bytecode (executable code)**