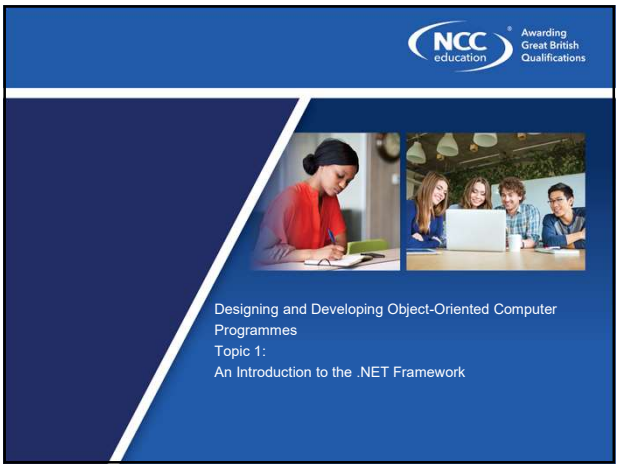
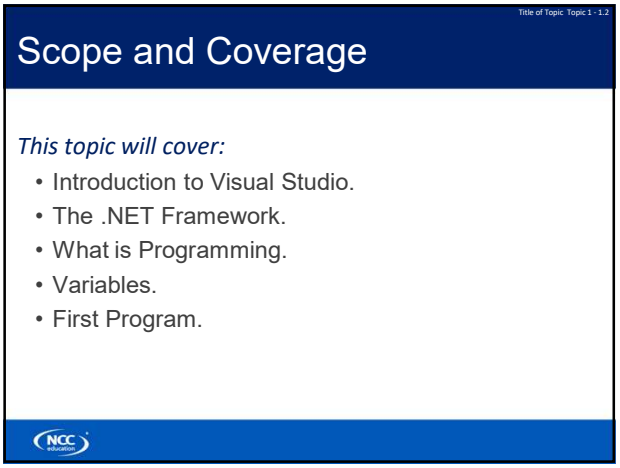
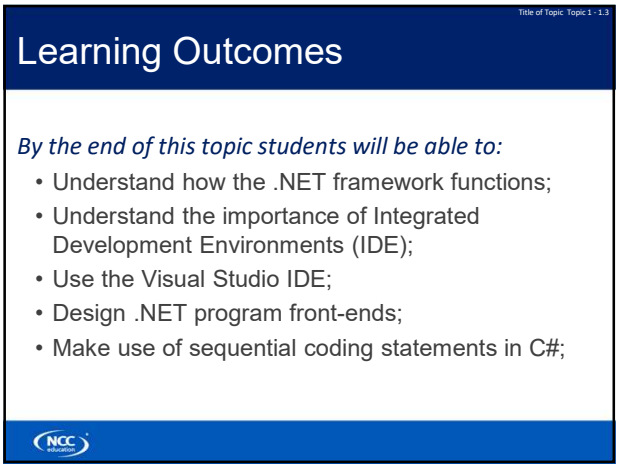

NCC education
Awarding Great British Qualifications
Designing and Developing Object-Oriented Computer Programmes
Topic 1:
An Introduction to the .NET Framework

1


Title of Topic Topic 1 - 1.2
Scope and Coverage
This topic will cover:
• Introduction to Visual Studio.
• The .NET Framework.
• What is Programming.
• Variables.
• First Program.

2


Title of Topic Topic 1 - 1.3
Learning Outcomes
By the end of this topic students will be able to:
• Understand how the .NET framework functions;
• Understand the importance of Integrated Development Environments (IDE);
• Use the Visual Studio IDE;
• Design .NET program front-ends;
• Make use of sequential coding statements in C#;

3

Title of Topic Topic 1 - 1.4

Introducing Visual Studio

- In this course we will be looking at how to build object oriented, event driven programs using the C# programming language.
- C# is part of Microsoft's .NET framework, and is a very popular language that is in considerable demand.
- We will be making use of Visual Studio as an Integrated Development Environment, or IDE.

4

Title of Topic Topic 1 - 1.5

IDE

- An IDE is a software package that combines all of the separate tools into a single application so that everything is contained within a unified package.
- This makes it easier for software to be developed since there's no need to constantly shift between multiple different tools.
- Visual Studio is one such IDE, and one of the most powerful available.

5

Title of Topic Topic 1 - 1.6

The .NET Framework

- C# is a programming language that is part of the .NET framework. This is a large and powerful programming framework that is aimed primarily, but not entirely, at Windows based platforms.
- The .NET framework is made up of five key parts:
  - The Common Language Runtime (CLR)
  - The base class library
  - Common language specifications
  - ADO .NET
  - ASP .NET

6

Title of Topic Topic 1 - 1.7

Rules and Syntax

- Every programming language is made up of a set of rules as to how various parts should go together, and a set of special words.
- When we put these together, we are working with the syntax of our language.
- C# is known as a C-type language, beause it makes use of a syntax that is derived from the C programming language.
- As such, when you know how to code in C# you should find it easy to move into PHP, Javascript, Java, or any number of other languages.

7

Title of Topic Topic 1 - 1.8

Source Code

- What we write is known as **source code**, and it's a set of instructions in a format that is relatively easy for humans to understand.
- It's not provided in a form that makes sense to the computer– it's something that's supposed to be easy for us to work with.
- All we have on our computers when we save a piece of source code is a text-file, not a computer program.

8

Title of Topic Topic 1 - 1.9

Compilers

- In order to turn our source code into a computer program we go through a process of **compilation**. The .NET framework takes the code we have written and turns it into a special kind of computer code that is understood by the Common Language Runtime.
- Every language, no matter what we use, **compiles** down into exactly the same computer code. This is known as **intermediate language** (IL).

9

Title of Topic Topic 1 - 1.10

Common Language Runtime

- On any computer where the .NET framework is installed there is a piece of software running called the Common Language Runtime (CLR).
- The CLR is a virtual machine – when we run our intermediate language it's not run on the computer itself. Instead, we provide instructions and requests to the CLR, and the CLR sends requests on to the operating system as required.

10

Title of Topic Topic 1 - 1.11

Machine Code

- When a program was compiled it ended up as machine code – a series of 1s and 0s that were understood by the electronics of the machine itself.
- Every different kind of processor and every different kind of operating system had its own way of representing and executing machine code.
- If you wanted to have one application that ran on lots of different systems, you'd need to compile it on each.
- **Porting** was the process of turning code written for one system into working code for another.

11

Title of Topic Topic 1 - 1.12

Virtual Machine

- The use of a **virtual machine** means that for each platform where the code is supposed to work all that's needed is a version of the Common Language Runtime, and Microsoft writes that for us.
- We don't write code for the computer we're on – we write it for the CLR, and Microsoft makes a CLR available for all supported platforms.
- When we write our code for .NET, it will run anywhere we can get access to a CLR.

12

Title of Topic Topic 1 - 1.13

For the Future

- .NET then is a powerful tool, and we're only going to scratch the surface during this course.
- However, what you learn here is going to open up a lot of doors for the future – any piece of software you need to write, you'll be able to do it with C# and .NET as your skills develop.

13

Title of Topic Topic 1 - 1.14

What is Programming?

- Programming is a difficult task, unlikely almost anything you may have learned before – it's like learning how to do mathematics in a foreign language, it needs you to learn new syntax combined with a formally exact vocabulary.
- When you learn how to speak a different language, people can often make out meaning even when you make mistakes. A computer cannot do that. And worse, it does exactly what you tell it to do.

14

Title of Topic Topic 1 - 1.15

Instructions - 1

- Consider the following instructions.
  1. Get two eggs from the fridge
  2. Get some milk from the fridge
  3. Get some butter from the fridge
  4. Break the eggs into a jug
  5. Whisk the eggs until they are well mixed.
  6. Heat some milk and some butter in a pot.
  7. Add the egg mixture.
  8. Stir until scrambled eggs are made.

15

Title of Topic Topic 1 - 1.16

Instructions - 2

- The instructions are pretty simple, but your assistant falters at the first instruction. He, or she, stands hitting the fridge door. You need to amend your instructions a little:
  1. Open the fridge.
  2. Get two eggs from the fridge
  .....

16

Title of Topic Topic 1 - 1.17

Instructions - 3

- You set your assistant back to work and they open the fridge. They get two eggs out. Then they stand there with a confused look on their face. "What does 'some' mean?", they ask. We need to be more specific:
  1. Open the fridge.
  2. Get two eggs from the fridge
  3. Get two tablespoons of milk from the fridge
  ……..
- And so on, until you have a completely correct, entirely unambiguous list of instructions that anyone can follow.

17

Title of Topic Topic 1 - 1.18

Variables - 1

- Most of what we do in C# is going to involve the use of a **variable.**
- Think of a variable as a little box with a name, into which we put some information so it's available for later use.
- Within C#, variables have a name, which we use to refer to it, and a type which tells C# what kind of data might be stored within. We create a variable like so:

```
<Type> <name>;
```

18

Title of Topic Topic 1 - 1.19

Variables - 2

- Some of the types we'll see a lot are known as primitive data types and make up the basic building blocks of the language;
- int Whole numbers. Short for integer
- double Real numbers, making use of decimal points.
- Boolean A true or false value.
- String A sequence of alphanumeric characters

19

Title of Topic Topic 1 - 1.20

Variables - 3

- If we want to create a variable that holds a whole number, and we want that variable to be called num, we'd use the following line of code:
- int num;
- If we want to put something into that variable, we use what is known as an assignment:
- num = 10;

20

Title of Topic Topic 1 - 1.21

Variables - 4

- We can do simple sums using variables. For that, we use the set of what are known as arithmetic operators – plus, minus, division (represented by /) and multiplication (represented by *). For example:
- int num;
- int ans;
- num = 10;
- ans = num * 20;

21

Title of Topic Topic 1 - 1.22

Variables - 5

- There are a few other terms we should define before we go forward too, so that you're familiar with them when you encounter them:

Term	Description
Syntax	The formal parts of a language that define what is valid when writing code. For example, the order of words we use to create a variable is defined by the language **syntax**.
Syntax error	An error that is the result of us not obeying the syntax of the language. For example, if we declared a variable like so: `num is an int;` That would be a syntax error, and our program would not **compile**.
Compile	The act of turning the human readable code in the IDE into something the computer can run and execute upon.
Runtime error	An error that occurs when the program is running – usually caused by incorrect logic in the program code we provide.
Widget or Control	A specific user interface element we place on our application. These include buttons, text fields, and combo boxes amongst other things.
GUI	A Graphical User Interface – that's what our users will use to interact with the programs we write.

22

Title of Topic Topic 1 - 1.23

Let's do some Programming - 1

- Start up Visual Studio, and you'll be presented with your programming environment.
- We'll be creating a **windows form application** and doing some simple things with it.
- At the top left of this interface is your **form**.
- This is the window of the application you'll be creating.
- Along the right hand side at the top is the **solution explorer**, which lets us look through all the files that are part of the project.

23

Title of Topic Topic 1 - 1.24

Let's do some Programming - 2

- Underneath that is the **property explorer**.
- We'll be using that a lot as we go through this course.
- There is a **toolbox** tab at the top left of the interface. Clicking this will open the toolbox of components we can select.
- We want to expand **common controls** which will reveal the standard GUI components we'll be using.

24

Title of Topic Topic 1 - 1.25

Let's do some Programming - 3

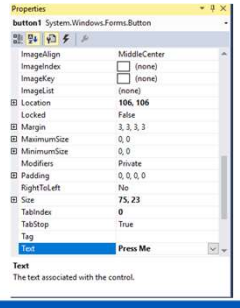
- Double click on **Button**, and it'll place a button on your application.
- This is how we begin with writing a C# program.
- We can drag and drop this button to where we want it to go, but more importantly we can also change the way it looks and behaves by changing its **properties**.

25

Title of Topic Topic 1 - 1.26

Let's do some Programming - 4

- Drag the button into the centre of your form, and then find the **Text** property.
- Change it to **press me.**



26

Title of Topic Topic 1 - 1.27

Let's do some Programming - 5

- Scroll up to the (name) property and change it to cmdPress.
- This won't change anything in the application, but it changes the name we use when we want to refer to this button in code.
- We'll usually do this by giving it a name that begins with a three letter prefix that tells us what kind of control it is.
- We don't have to, but it's good practise because it means we'll know in the code what kind of control we're working with.

27

Title of Topic Topic 1 - 1.28

Our first C# Program - 1

- What we've drawn here on the form is a **button control**.
- Controls are the building blocks of C# programs.
- We've set its **text** property to change how it looks, but there are many more properties that we haven't touched.

28

Title of Topic Topic 1 - 1.29

Our first C# Program - 2

- C# is an **event driven language**, which means that when we write code we associate it with **events** that occur in the lifetime of a program.
- When we drag a window around the display, we're creating events.
- When we move the mouse across it, we create events.
- When we click on a button, we create an event.

29

Title of Topic Topic 1 - 1.30

Our first C# Program - 3

- Most of the events that happen we're not interested in.
- Some of them we are. For example, we want to know when someone clicks that button we added.

30

Title of Topic Topic 1 - 1.31

Our first C# Program - 4

- Double click on the button within the IDE and it'll bring up the code editor for this program.
- Don't let this intimidate you – it's easier than you think because Visual Studio does most of the work for us here.

31

Title of Topic Topic 1 - 1.32

Our first C# Program - 5

```
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void cmdPress_Click(object sender, EventArgs e)
        {

        }
    }
}
```

32

Title of Topic Topic 1 - 1.33

Our first C# Program - 6

- There is a part of this code that contains the **stub** where we'll put the code relating to our button:

```
private void cmdPress_Click(object sender, EventArgs e)
{

}
```

- This tells C# that we have a control called cmdPress, and we want to handle its **click** event – that is, when someone clicks on it.

33

Title of Topic Topic 1 - 1.34

Our first C# Program - 7

- Add in the following code:

```
private void cmdPress_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello World");
}
```

- Be sure to do this exactly – C# is not forgiving of errors, and it is expects that you'll obey the **syntax** of the language.

34

Title of Topic Topic 1 - 1.35

Our first C# Program - 8

- Now, click on the play symbol at the top of Visual Studio and press the button when your application starts up.
- You'll see it displays a message box containing the text we entered.
- Press the stop button, and it'll stop and return you to the IDE.

35

Title of Topic Topic 1 - 1.36

Our first C# Program - 9

- Note you can switch between the code view and the **design view** by selecting the tab at the top of the screen.
- Making use of the Toolbox as before, add in a TextBox control.
- Call it txtName, and position it above the button on your form.

36

Title of Topic Topic 1 - 1.37

Our first C# Program - 10

- Now adjust the code so it says the following:

```
private void cmdPress_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hello " + txtName.Text);
}
```

- Run the program again, and type something into the text box.
- Press the button, and you'll see it appears in the messagebox that appears:

37

Title of Topic Topic 1 - 1.38

Our first C# Program - 11

- And if you wanted to put something **into** the textbox when you press the button:

```
private void cmdPress_Click(object sender, EventArgs e)
{
    txtName.Text = "My content in here";
}
```

- This is the crux of software development in Visual Studio.
- We add controls to a form, and then in the code that we write we make use of the properties of those controls to make things happen.

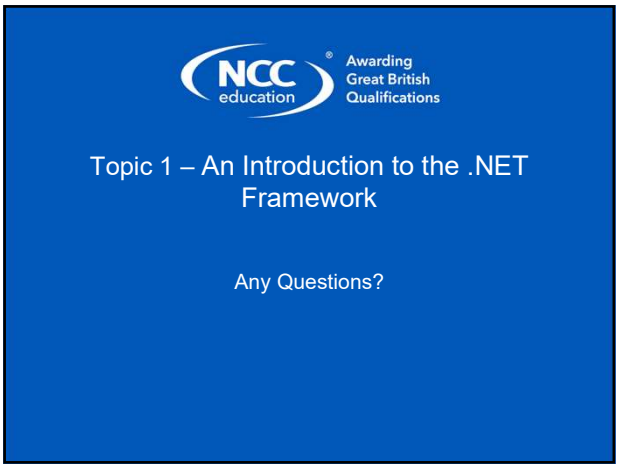
38

Title of Topic Topic 1 - 1.39

Our first C# Program - 12

- We will have to wait until we've covered some more of C#'s design before we can talk about what most of this code does, but we'll get to it.
- For now though, congratulations on completing your first C# program!

39


NCC education
Awarding
Great British
Qualifications
Topic 1 – An Introduction to the .NET Framework
Any Questions?

40