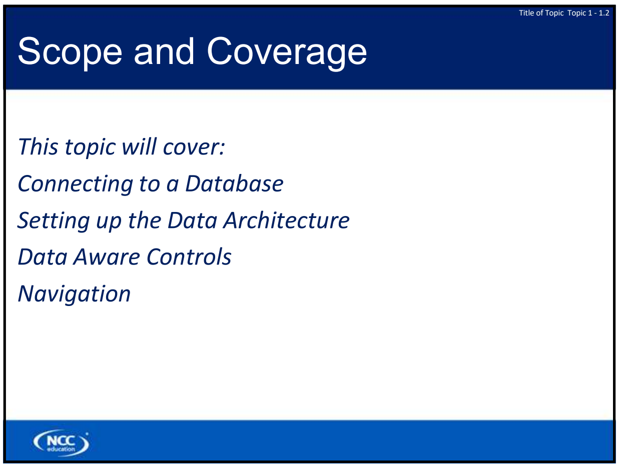
Designing and Developing Object-Oriented Computer Programmes
Topic 12:
Databases in .NET

1



Title of Topic Topic 1 - 1.2

Scope and Coverage

This topic will cover:

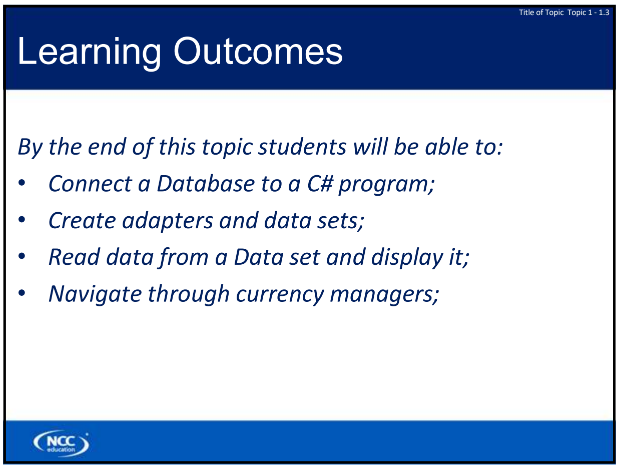
Connecting to a Database

Setting up the Data Architecture

Data Aware Controls

Navigation

2



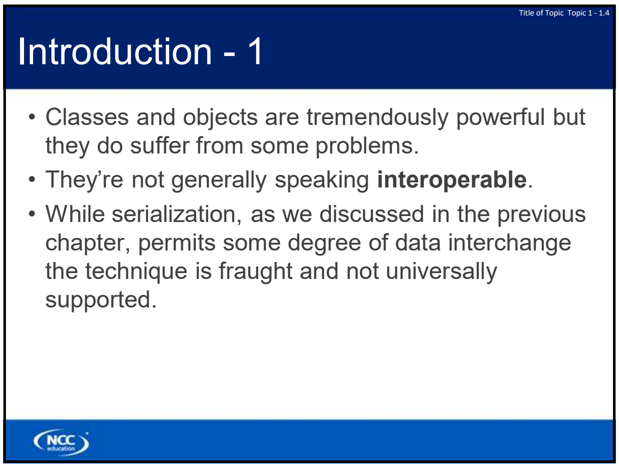
Title of Topic Topic 1 - 1.3

Learning Outcomes

By the end of this topic students will be able to:

- *Connect a Database to a C# program;*
- *Create adapters and data sets;*
- *Read data from a Data set and display it;*
- *Navigate through currency managers;*

3

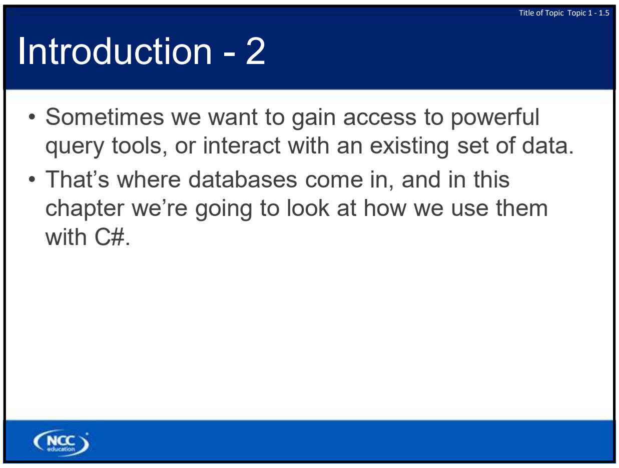


Title of Topic Topic 1 - 1.4

Introduction - 1

- Classes and objects are tremendously powerful but they do suffer from some problems.
- They're not generally speaking **interoperable**.
- While serialization, as we discussed in the previous chapter, permits some degree of data interchange the technique is fraught and not universally supported.

4

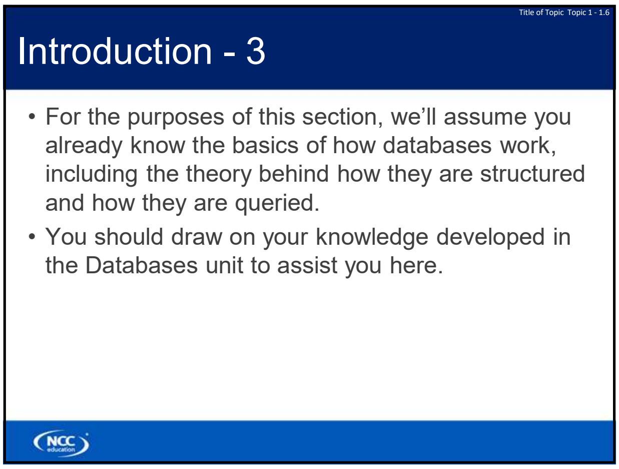


Title of Topic Topic 1 - 1.5

Introduction - 2

- Sometimes we want to gain access to powerful query tools, or interact with an existing set of data.
- That's where databases come in, and in this chapter we're going to look at how we use them with C#.

5



Title of Topic Topic 1 - 1.6

Introduction - 3

- For the purposes of this section, we'll assume you already know the basics of how databases work, including the theory behind how they are structured and how they are queried.
- You should draw on your knowledge developed in the Databases unit to assist you here.

6

Title of Topic Topic 1 - 1.7

Connecting to a Database - 1

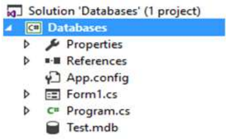
- A database is an external file, and so to make use of it we need to hook it up to our program.
- We do this by choosing **add > existing item** and navigating to where it is stored on our system.
- We will likely have to change the file-type to **data files** to get it to show up.

7

Title of Topic Topic 1 - 1.8

Connecting to a Database - 2

- Once we've added it, we'll have it appear as part of our solution explorer, like so:



Solution 'Databases' (1 project)
Databases
Properties
References
App.config
Form1.cs
Program.cs
Test.mdb

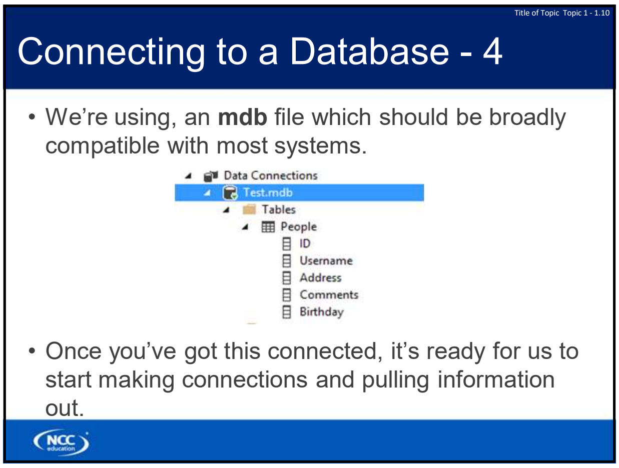
8

Title of Topic Topic 1 - 1.9

Connecting to a Database - 3

- This will also reveal the server explorer, which can be used to examine the structure of the database you just connected.
- Click into it to make sure the connection works – if it doesn't, you may need to make sure you have the appropriate database drivers and packages installed on your system.

9



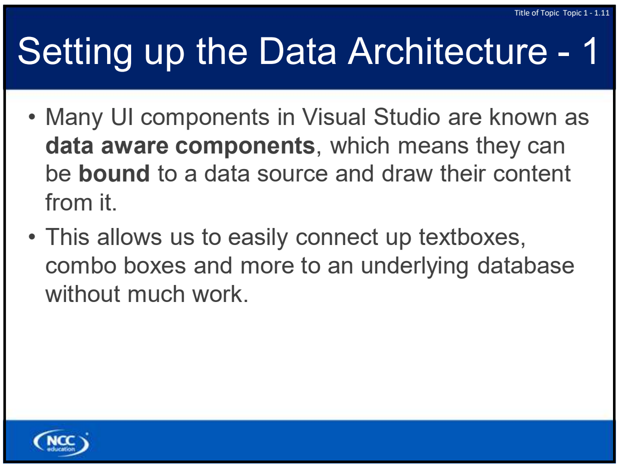
Title of Topic Topic 1 - 1.10

Connecting to a Database - 4

- We’re using, an **mdb** file which should be broadly compatible with most systems.

- Once you’ve got this connected, it’s ready for us to start making connections and pulling information out.

10

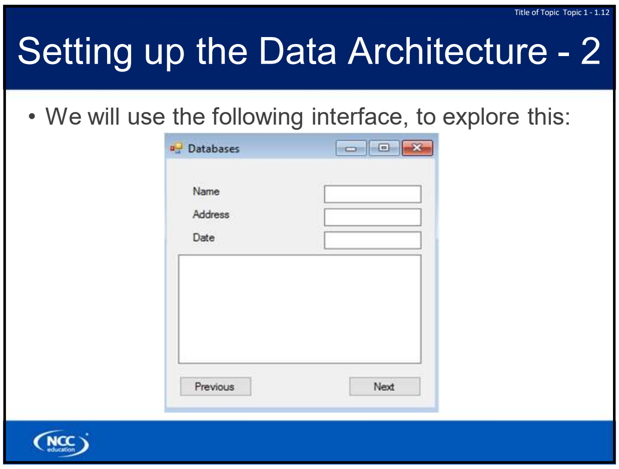


Title of Topic Topic 1 - 1.11

Setting up the Data Architecture - 1

- Many UI components in Visual Studio are known as **data aware components**, which means they can be **bound** to a data source and draw their content from it.
- This allows us to easily connect up textboxes, combo boxes and more to an underlying database without much work.

11



Title of Topic Topic 1 - 1.12

Setting up the Data Architecture - 2

- We will use the following interface, to explore this:

12

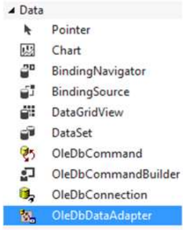
Setting up the Data Architecture - 3

- First of all, we need to create a **data adapter** which sits between our program and the database.
- This is used to provide a layer that means our code doesn’t need to change if different databases are plugged in.
- For this, we need to use the **Data** tab in our toolbox and add an **OleDBDataAdapter**.

13

Setting up the Data Architecture - 4

- If that’s not there, right click on the toolbox, choose ‘choose items’ and add the four OleDB components you’ll find.



14

Setting up the Data Architecture - 5

- The data adapter goes onto your form in the same way as any control, but it’ll be invisible.
- When you drag it onto the form, you’ll go through a series of dialogs that step you through the process.
- You’ll pick the database we added (test.mdb), choose ‘Use SQL Statements’, and then provide the following simple SQL statement:

```
SELECT * FROM PEOPLE
```

15

Title of Topic Topic 1 - 1.16

Setting up the Data Architecture - 6

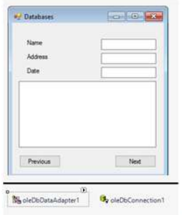
- You can do whatever you like with the SQL here if you know how it works, but this basic statement will just pull every record from the People table.
- You can see how you might be able to do more interesting things with a bit of SQL magic.

16

Title of Topic Topic 1 - 1.17

Setting up the Data Architecture - 7

- The fully configured adapter we drag across will now appear in the 'tray' of our application – a separate part of the builder that hides the 'invisible' things from the main form:



17

Title of Topic Topic 1 - 1.18

Setting up the Data Architecture - 8

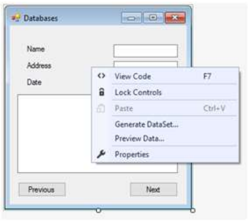
- If we want to draw a different set of data from a different query we'd use a second adapter, but they'll all be connected to the same data.
- That in turn is done through an object called a **Dataset** which represents a snapshot of a specific set of the database – content from tables, queries or SQL statements. To get our dataset, we right click on our form and choose **generate dataset:**

18

Title of Topic Topic 1 - 1.19

Setting up the Data Architecture - 9

- To get our dataset, we right click on our form and choose **generate dataset:**

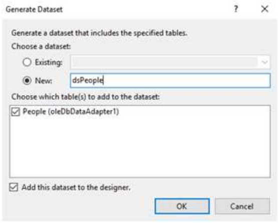


19

Title of Topic Topic 1 - 1.20

Setting up the Data Architecture -10

- This will bring up the wizard for generating a dataset – we'll give it a name (dsPeople) and then click 'ok':


Generate Dataset

Generate a dataset that includes the specified tables.

Choose a dataset:

○ Existing:

● New: dsPeople

Choose which table(s) to add to the dataset:

☑ People (oleDbDataAdapter1)

☑ Add this dataset to the designer.

20

Title of Topic Topic 1 - 1.21

Setting up the Data Architecture -11

- And then **finally** we've made the connection from our application to the database, and we can start hooking up our data aware controls.

21

Title of Topic Topic 1 - 1.22

Data Aware Controls - 1

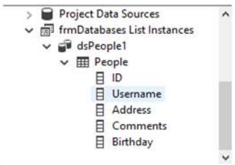
- We need to decide which controls are going to display specific parts of our dataset.
- We've already done that though, so all we need to do is **bind** the right control to the right part of the data.

22

Title of Topic Topic 1 - 1.23

Data Aware Controls - 2

- Each data aware control has a property called (DataBindings), and that in turn has a sub property called Text.
- By exploring the drop down menu it gives us, we can drill down into a single field of a single table in our project:



- Project Data Sources
- frmDatabases List Instances
  - dsPeople1
    - People
      - ID
      - Username
      - Address
      - Comments
      - Birthday

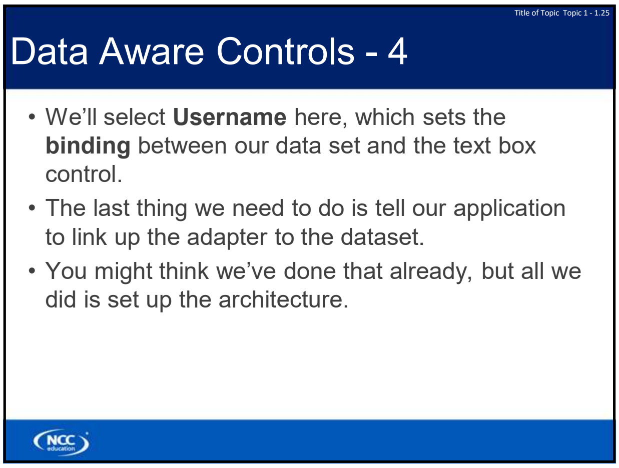
23

Title of Topic Topic 1 - 1.24

Data Aware Controls - 3

- Note that you'll want the data sets that are associated with the List Instances of our project – those are the ones that are properly configured.

24

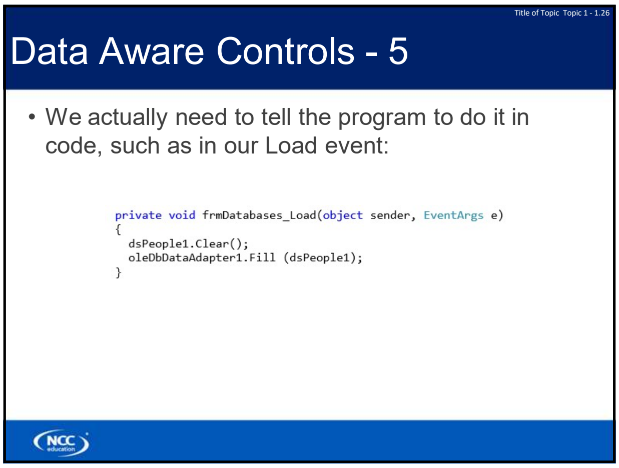


Title of Topic Topic 1 - 1.25

Data Aware Controls - 4

- We'll select **Username** here, which sets the **binding** between our data set and the text box control.
- The last thing we need to do is tell our application to link up the adapter to the dataset.
- You might think we've done that already, but all we did is set up the architecture.

25



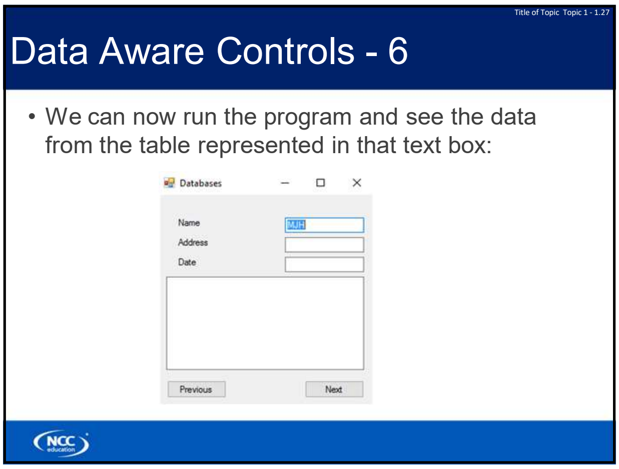
Title of Topic Topic 1 - 1.26

Data Aware Controls - 5

- We actually need to tell the program to do it in code, such as in our Load event:

```
private void frmDatabases_Load(object sender, EventArgs e)
{
  dsPeople1.Clear();
  oleDbDataAdapter1.Fill (dsPeople1);
}
```

26



Title of Topic Topic 1 - 1.27

Data Aware Controls - 6

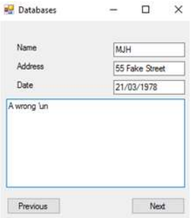
- We can now run the program and see the data from the table represented in that text box:

Name MJH

27

Data Aware Controls - 7

- We then go through the other text boxes and bind them to the appropriate locations in the database to display a full record for the data set we've created:



Name MJH
Address 55 Fake Street
Date 21/03/1978
A wrong 'un

28

Data Aware Controls - 8

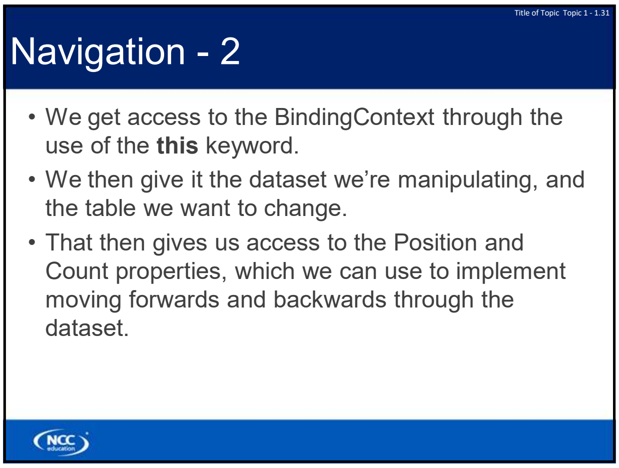
- If we change the underlying database and rerun our program, we'll get a newly filled adapter full of the data that's just changed.
- As you can imagine, this is very powerful if we're looking to share data between applications.
- But this is a passive view of data already there – we also need to know how to change it and navigate.

29

Navigation - 1

- Each of the current records being shown in the database is handled by a system called a **CurrencyManager**.
- Its job is to keep track of which controls are pointing to which fields of which records.
- You don't need to worry about the details of this – all we need to know is that every form in C# has a **BindingContext** object that keeps track of all these CurrencyManagers, and that gives us the tool we need to move things around.

30

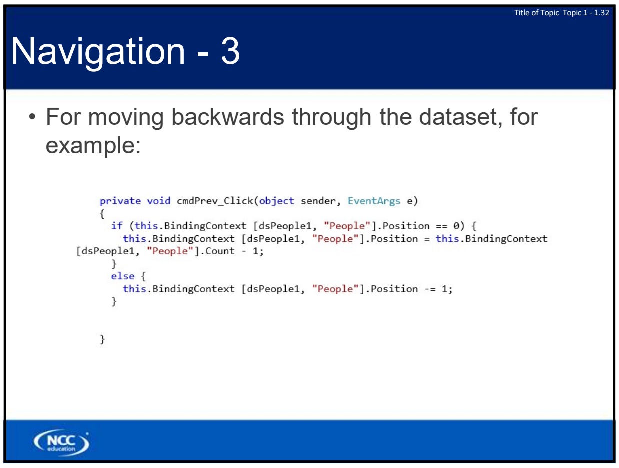


Title of Topic Topic 1 - 1.31

Navigation - 2

- We get access to the BindingContext through the use of the **this** keyword.
- We then give it the dataset we're manipulating, and the table we want to change.
- That then gives us access to the Position and Count properties, which we can use to implement moving forwards and backwards through the dataset.

31



Title of Topic Topic 1 - 1.32

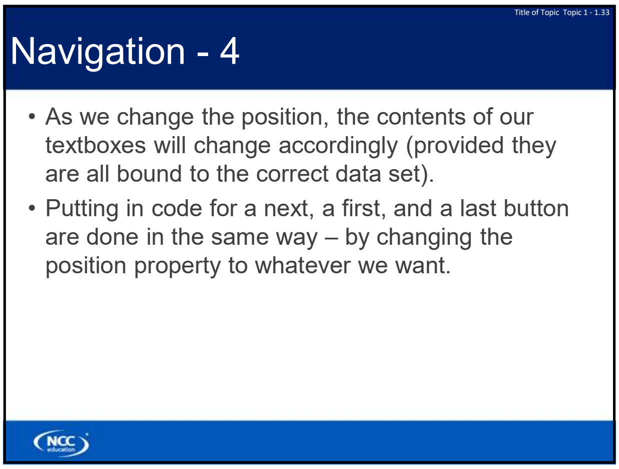
Navigation - 3

- For moving backwards through the dataset, for example:

```
    private void cmdPrev_Click(object sender, EventArgs e)
    {
      if (this.BindingContext [dsPeople1, "People"].Position == 0) {
        this.BindingContext [dsPeople1, "People"].Position = this.BindingContext
[dsPeople1, "People"].Count - 1;
      }
      else {
        this.BindingContext [dsPeople1, "People"].Position -= 1;
      }


    }
```

32

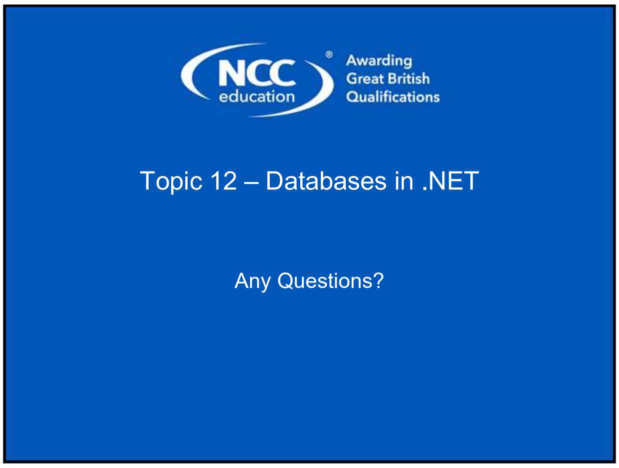


Title of Topic Topic 1 - 1.33

Navigation - 4

- As we change the position, the contents of our textboxes will change accordingly (provided they are all bound to the correct data set).
- Putting in code for a next, a first, and a last button are done in the same way – by changing the position property to whatever we want.

33



NCC education Awarding Great British Qualifications

Topic 12 – Databases in .NET

Any Questions?

34