INF 154
LECTURE 1
2021

- Introduction to Programming
- How to use Visual Studio
- Your first C# programs

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

After this lecture you should …

1. Be able to use the Visual Studio Interactive Development Environment to create a simple Windows Forms Based program in C#
2. Know how to create, run and save a C# project
3. Understand what a Control, a Property, an Event and an Event handler are
4. Know how to use the following C# controls e.g button, form etc

WHAT IS PROGRAMMING?

- Programming is the **skill to write code** that **a computer can follow**, **to execute a plan you made**, **to solve a problem**.
- Computer programming is a **way of giving computers instructions** about what they should do next.
- These **instructions are known as code**, and **computer programmers write code** to **solve problems** or **perform a task.**

There are therefore two aspects to programming:

1) **Making a plan to solve a problem**
2) **Translating the plan into code** that a computer can understand

For 1) you need computational thinking which includes algorithmic thinking

For 2) you need software and you need to learn the syntax/programming language.

WHAT IS PROGRAMMING?

For 1) you need computational thinking which includes algorithmic thinking

Computational Thinking

- The computational thinking process includes four key concepts:
  - **Decomposition**: Break the problem down into smaller, more manageable parts.
  - **Pattern Recognition**: Analyze data and identify similarities and connections among its different parts.
  - **Abstraction**: Identify the most relevant information needed to solve the problem and eliminate the extraneous/irrelevant details.
  - **Algorithmic Thinking**: Develop a step-by-step process to solve the problem so that the work is replicable by humans or computers.

For 2) you need software and you need to learn the syntax/programming language.

Software

- Visual Studio (Integrated Development Environment)

Programming Language

- C#

Why C#?

- One of the newer object-oriented programming languages
- Has links with C and C++
- Has the graphical user interface (GUI) features for rapid application development (RAD), such as Visual Basic and Delphi
- Has the added power of C++
- Has object-oriented class libraries and syntax similar to Java
- Can incorporate Web standards like HTML, XML and SOAP into programs

Why C#?

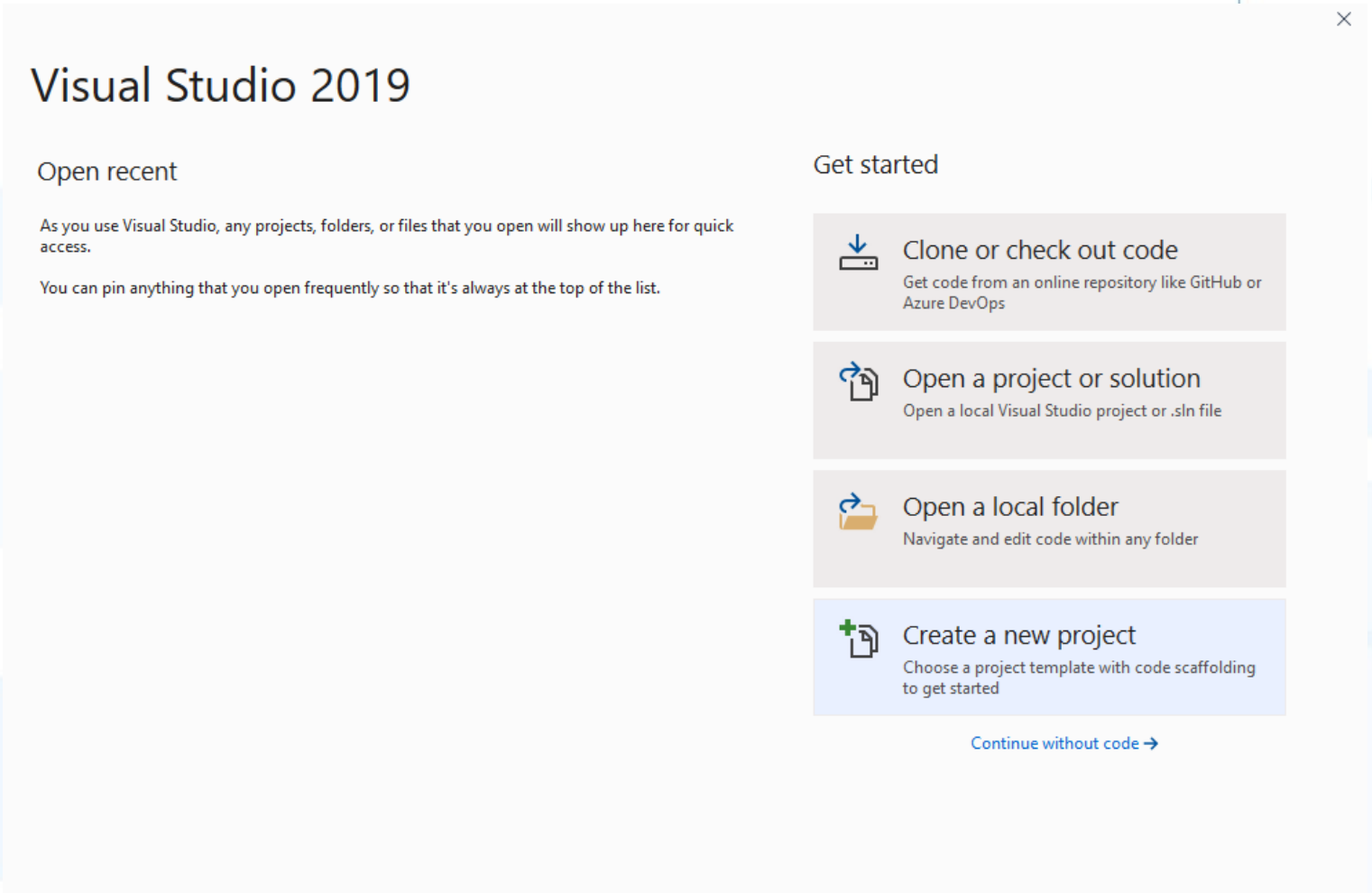
Can be used to develop different types of applications:

- Software components
- Mobile applications
- Dynamic web pages
- Database access components
- Windows desktop applications (with GUIs)
- Web services
- Console-based applications
- Class libraries and stand-alone components (.DLLs), smart device applications, and services

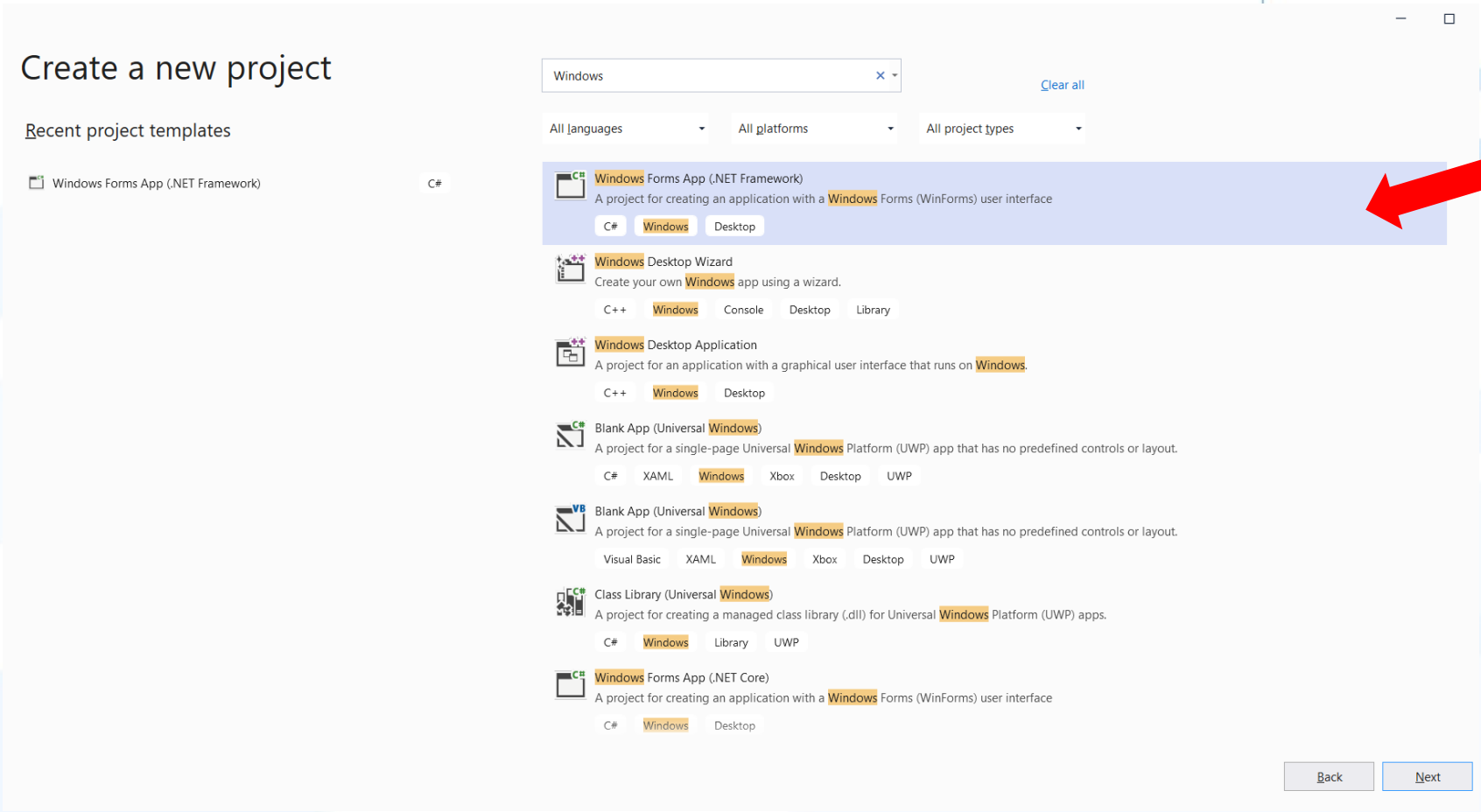
Why Visual Studio?

- VS Provides the IDE (Integrated Development Environment) for a number of programming languages including C#
- It contains all the features needed to easily create, run, and test programs
- It provides an intelligent editor for entering program code
- It includes a compiler for running and testing programs
- It allows easy debugging of programs

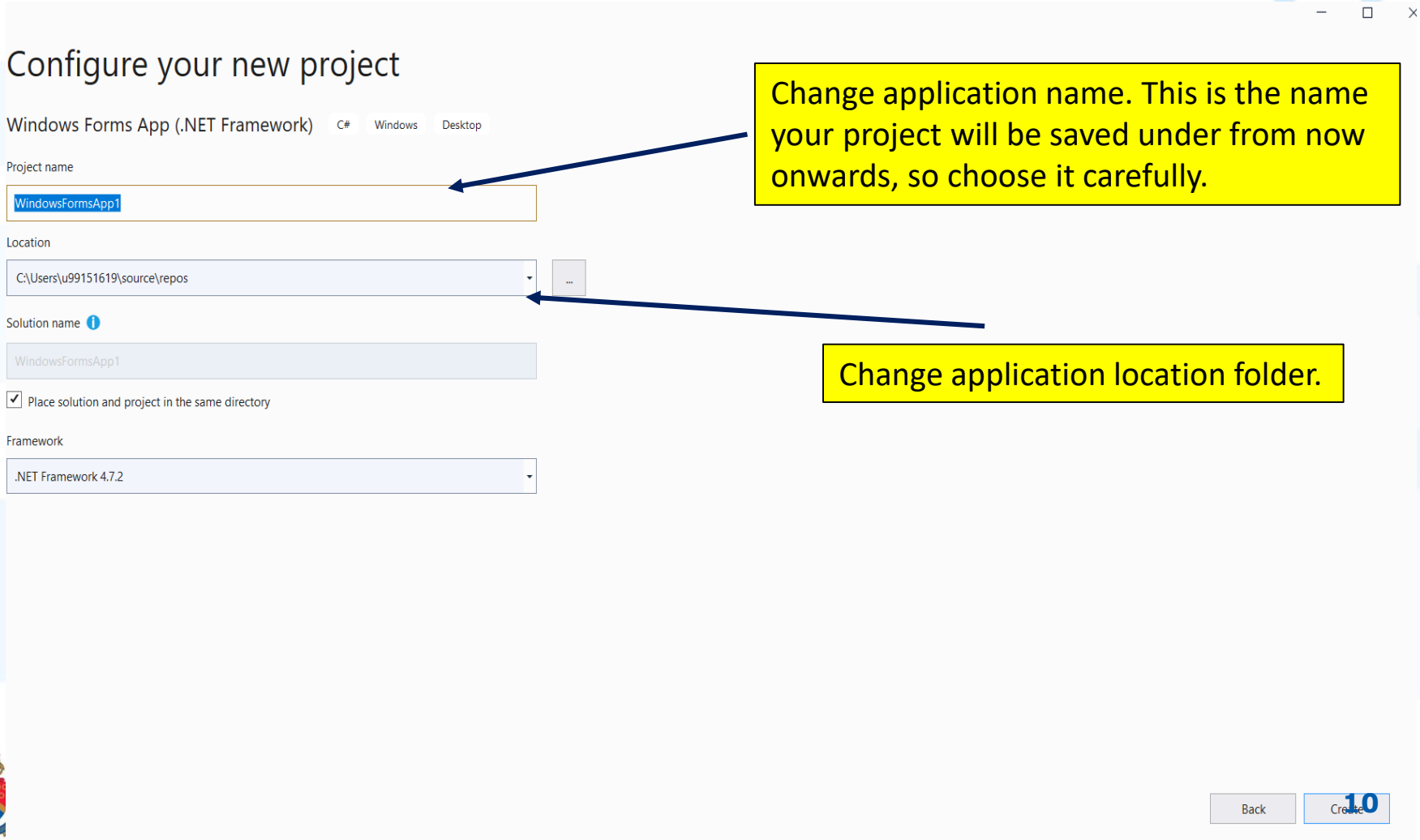
GETTING TO KNOW VISUAL STUDIO



GETTING TO KNOW VISUAL STUDIO



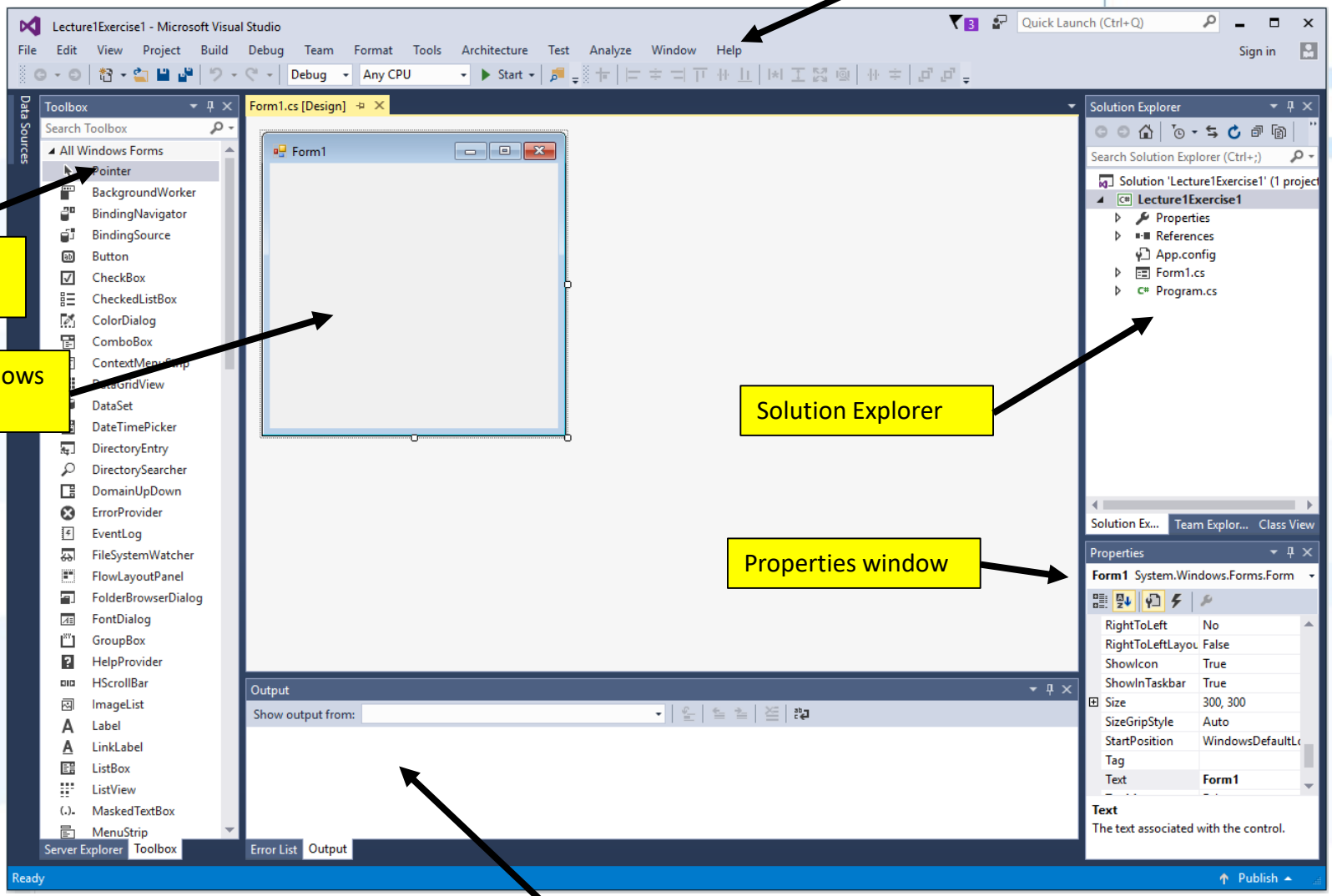
GETTING TO KNOW VISUAL STUDIO



Change application name. This is the name your project will be saved under from now onwards, so choose it carefully.

Change application location folder.

GETTING TO KNOW VISUAL STUDIO

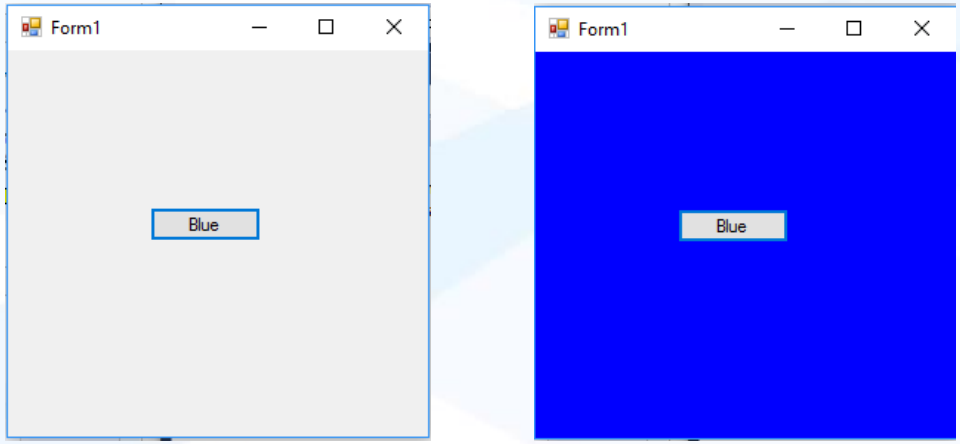


GETTING TO KNOW VISUAL STUDIO

- In INF154 and part of INF164 you will be writing **Windows Forms Programs**
- In Windows Forms programs coding often starts with creating the Graphical User Interface (GUI)
- A program's **GUI** consists of one or more **Forms**
- Every Form has a number of **Properties**, for instance:
  - Name
  - Text (The title that appears at the top of the Form)
  - Size
  - BackColor
- To build the GUI we place **Controls** on the Forms to define
  - Input mechanisms
  - Processing mechanisms
  - Output mechanisms
  - Information to guide the user

YOUR FIRST C# PROGRAM

- Our first program will consist of one Form with a single Button control.
- The text on the button will be “Blue”.
- When we run the program and click on the “Blue” button, the Form’s background colour will turn blue.
- It will appear as follows:

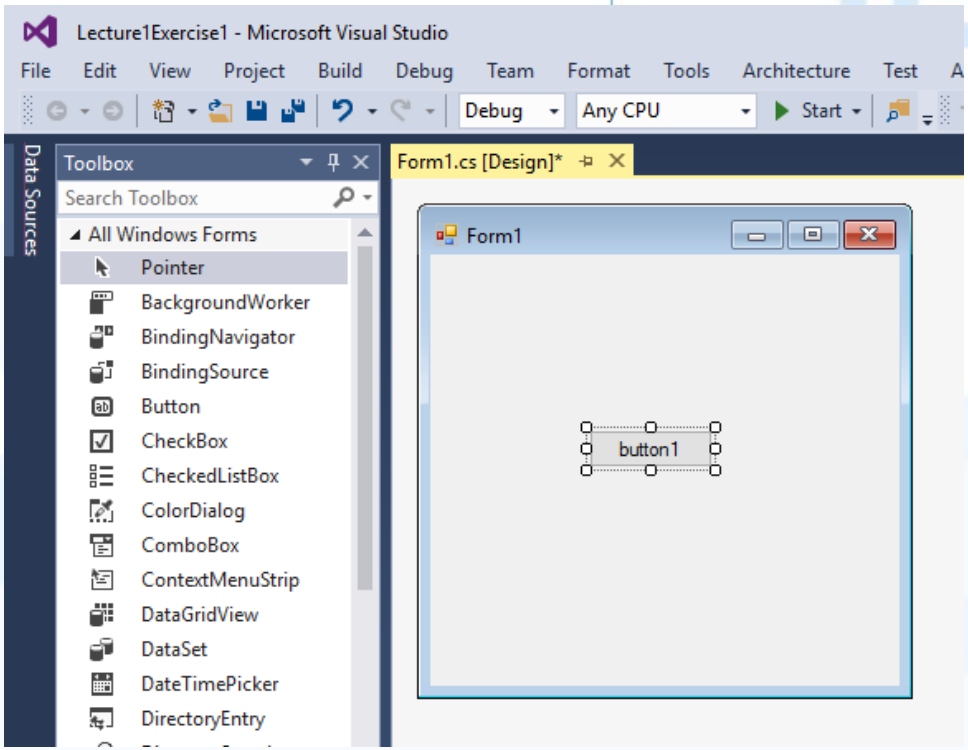


- In the next slides we write this program step-by-step and explain the programming concepts as we go along.

YOUR FIRST C# PROGRAM

Step 1 – Place a Button control on the form.

- The controls appear in the Toolbox, to the left of the Form (if it is not visible, click on View in the Menu and choose Toolbar)
- There are two ways to place a control on the Form:
  - Drag it from the Toolbar onto the Form, or
  - Double-click on it in the Toolbar and then drag it to the correct position on the Form
- You can move the button around on the form by dragging it

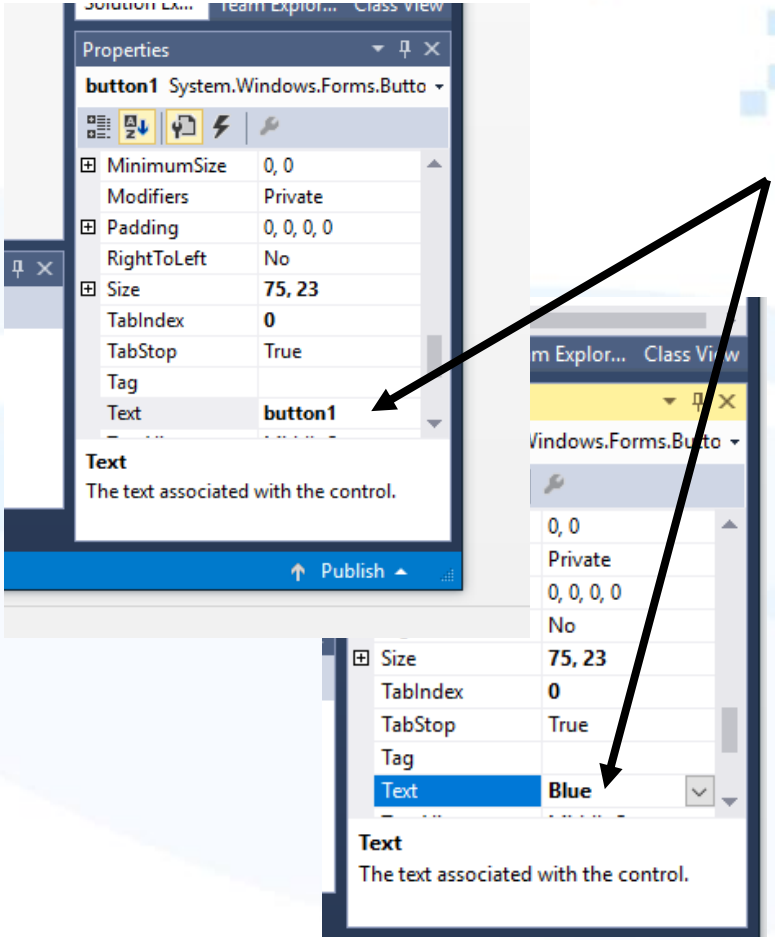


- You can change it's size by dragging the little "handles" that appear on its edges.
- To remove a control from the Form, make sure it is selected and press Delete on the keyboard.

YOUR FIRST C# PROGRAM

Step 2 - Changing the button's Text to "Blue"

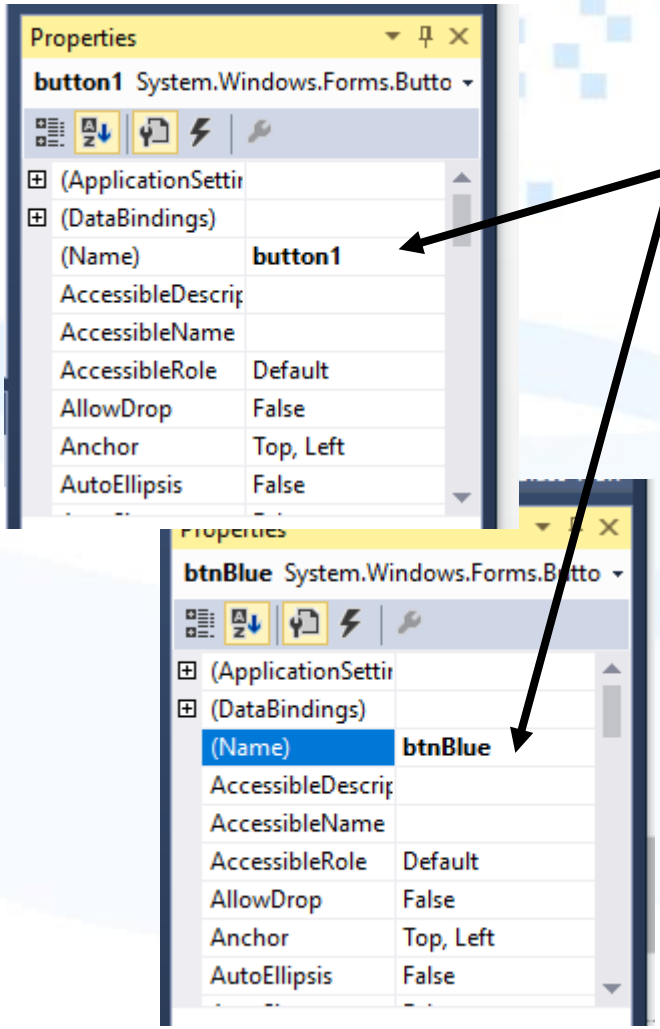
- The button appears with the default text "button1"
- We use the Properties Window to change a control's property values
- To change the text on the button, make sure the button is selected, then find the Text property in the Properties window
- In the right hand column, change "button1" to "Blue" and press Enter on the keyboard
- The text on the button should now be "Blue"



YOUR FIRST C# PROGRAM

Step 3 - Changing the button's Name to "btnBlue"

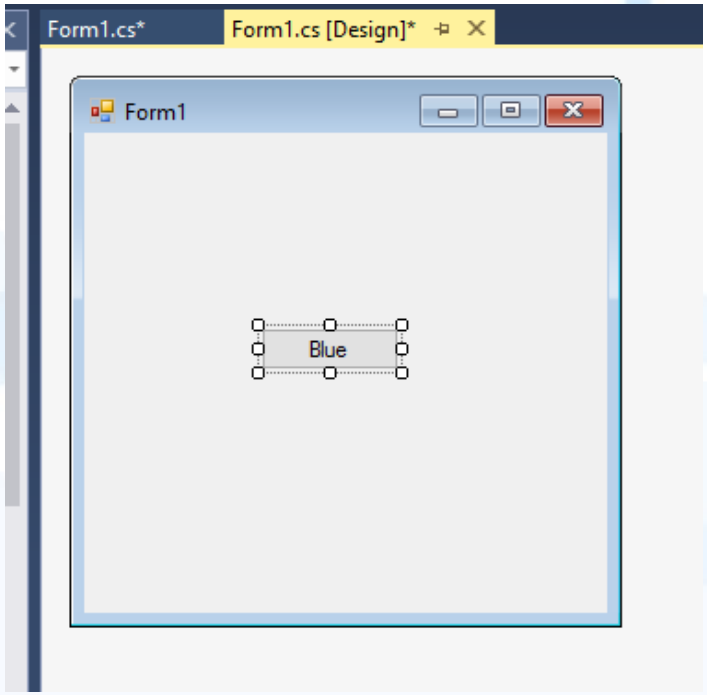
- A button control has a Text property and a Name property that are both initially button1 (or button2….).
- When we change the Text property to Blue, the Name property will still be button1.
- The Name property is not visible on the control – it is used when referring to a control in the program code
- To change the Name of the button, make sure the button is selected, then find the Name property in the Properties window
- In the right hand column, change "button1" to "btnBlue" and press Enter on the keyboard
- For every type of control we use three small letters in the beginning of its name so that it is clear what kind of control it is (e.g. btn for buttons). The rest of the name indicates the purpose of the control.



YOUR FIRST C# PROGRAM

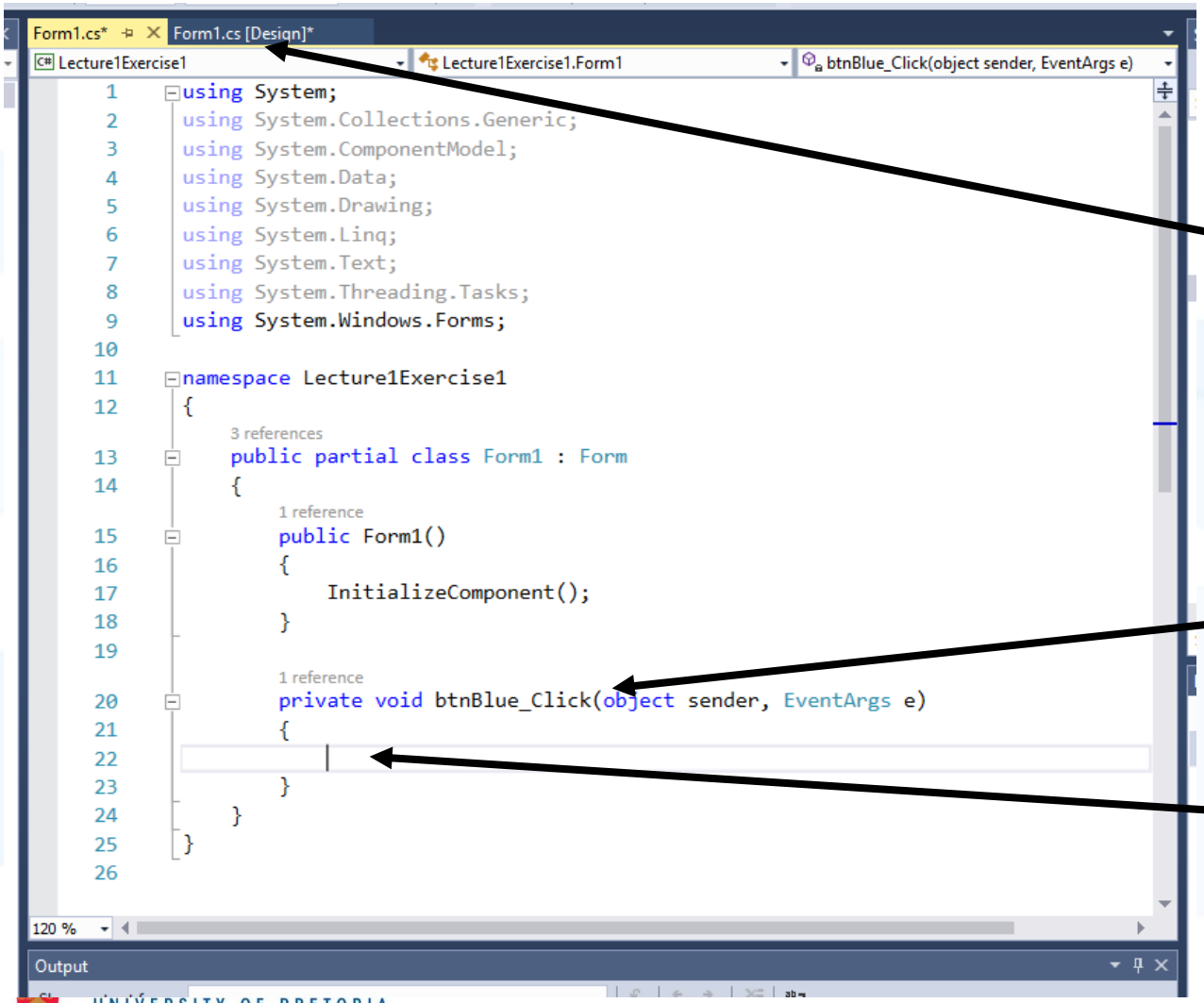
Step 4 – Adding program code to the button

- We now want to link a program statement to our button that will cause the form's background colour to turn blue when we click on the button.
- We call a user action (such as clicking) an "Event" and the code associated with the event an "Event handler" or "Event procedure".
- Now, to create the event handler for the button's click event, double-click on the button.
- This will open up the code window shown on the next slide in the place where the Form designer was.
- The cursor will appear where we have to write our code in the btnBlue_Click event handler – between the two curly brackets.



YOUR FIRST C# PROGRAM

Step 4 – Adding program code to the button



You can use these tabs to move between the Form designer and the Form's code

btnBlue's Click event handler

Where we write the program code that will execute when we

YOUR FIRST C# PROGRAM

Step 4 – Adding program code to the button

Type the following program statement into the event handler exactly as shown.

During the course of the year you will learn more about what each of the parts in the header line means. For now just remember not to change or delete any part of the header statement.

```
private void btnBlue_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}
```

"this" refers to the current Form you are programming in

The Form's BackColor property

The assignment operator

The value assigned to the BackColor property

Note that the program statement ends with a semicolon

YOUR FIRST C# PROGRAM

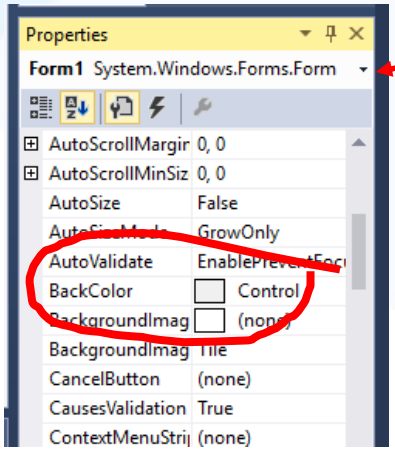
Step 4 – Adding program code to the button

A control's property can be changed in two ways:

1. Using program code (here the Form's BackColor property is changed to Blue).

```
private void btnBlue_Click(object sender, EventArgs e)
{
    this.BackColor = Color.Blue;
}
```

2. We can also change the same property in the Properties Window.

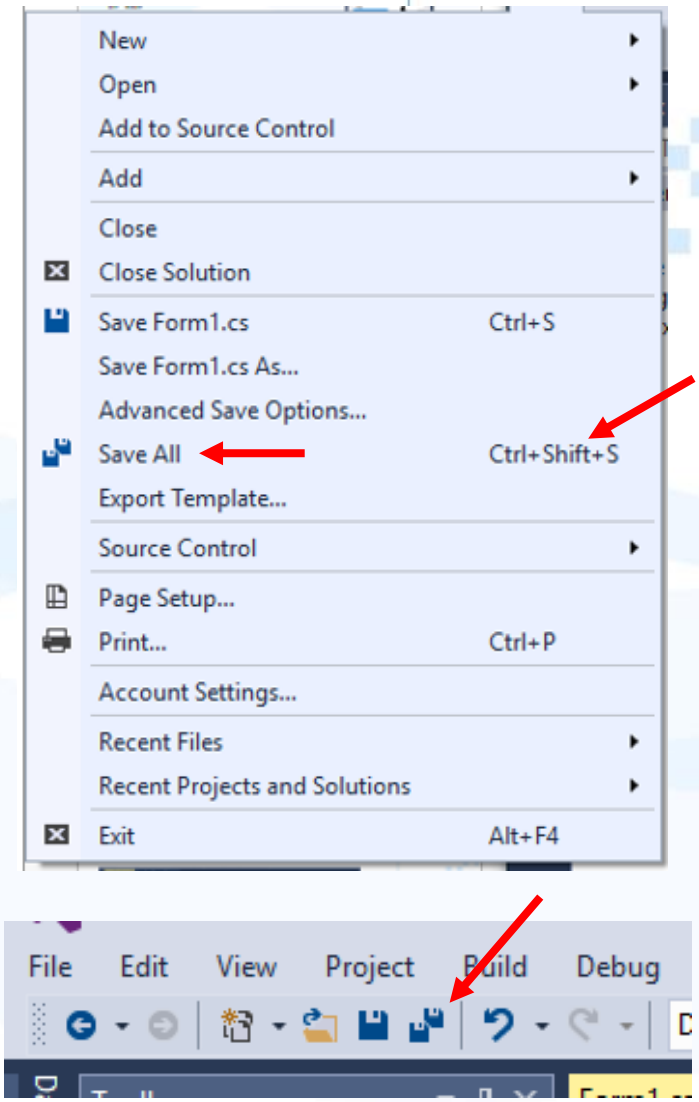


Choose the control you want to work with from this drop-down list, or select it on the form to make it active here. (Note – a Form is also a control.)

YOUR FIRST C# PROGRAM

Step 5 – Saving the project

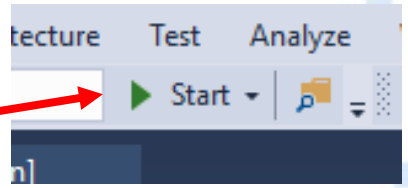
- It is very important that you save your project in the correct way, otherwise you will lose parts of the project and it may not execute correctly when you open it again later.
- We always use "Save All" on the menu (the arrows show the three options for doing this).
- It will be saved using the name and folder location that you specified when you first created the new project.
- If you use "Save Form1.cs" or Ctrl+S, only a part of your project is saved.



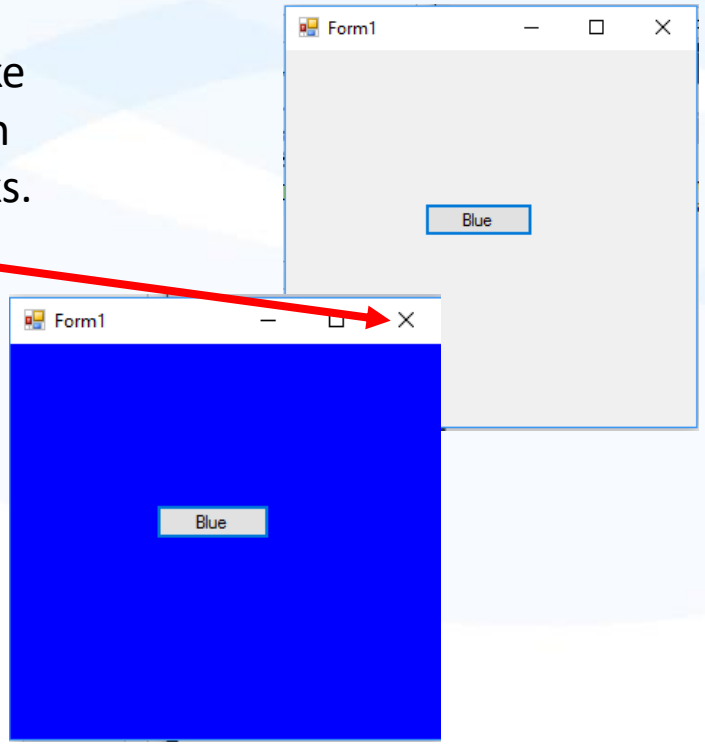
YOUR FIRST C# PROGRAM

Step 6 – Executing your program

- You can now run the program in one of the following ways:
  - Click on the Start button in the menu bar, or
  - Press F5



- When you run it for the first time it will take some time before the Form appears. When it does, click on the button to see if it works.
- To close the program click on the Close button on the Form



YOUR FIRST C# PROGRAM

Dealing with errors

There are two kinds or errors in programming – **syntax errors and logical (or run time) errors:**

- **Syntax errors are mistakes in the code syntax or typing errors such as leaving out the semicolon at the end of a program statement or not spelling the name of a control correctly.** Most of these errors are picked up by the Visual Studio editor and will be indicated with a red squiggly line in your code.
- In the example below there are two syntax errors. The text property must be written with a capital T, and Blue has to be preceded by Color – i.e. Color.Blue;
- If you hover with your mouse pointer over the red line a message box will pop up telling you what the error is.
- You have to fix all of these errors before you can run the program.

```
private void btnBlue_Click(object sender, EventArgs e)
{
    this.BackColor = Blue;
    this.text = "Blue";
}
```

YOUR FIRST C# PROGRAM

Dealing with errors

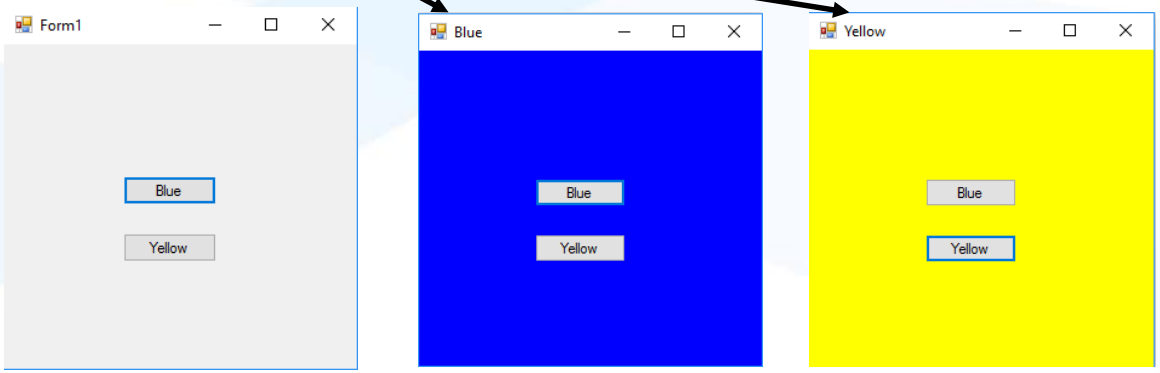
- **Run-time errors can be more difficult to identify. They occur when there are no syntax errors but the program is not executing correctly when you run it (i.e. it gives the wrong output).**
- A very simple example would be if you click on the "Blue" button and the Form's colour changes to red instead of blue.
- In this case you have to inspect the code to find the error – the Visual Studio editor will not able to pick it up. The programmer will have typed Color.Red instead of Color.Blue but to the compiler this would seem acceptable.
- As we write more complex problems you will learn to "debug" your programs because it is human to make errors while programming.

YOUR FIRST C# PROGRAM

Expanding the program

See if you can do the following on your own without looking at the solution on the next slides:

- Add a second button to the Form
- Name it btnYellow
- Make it's Text property "Yellow"
- Create a click event handler with code to change the background of the form to yellow
- Also add a statements to each of the two click event handlers that will change the Form's Text property to the current background colour.



Lecture 1 Homework

Review the following:

- Watch 3 videos on algorithms (See next slide)
- Watch 3 videos based on theory lecture (See ClickUP)

Additional videos: WHAT IS AN ALGORITHM?

https://www.youtube.com/watch?v=ZnBF2GeAKbo

https://www.youtube.com/watch?v=oRkNaF0QvnI

https://www.youtube.com/watch?v=CvSOaYi89B4