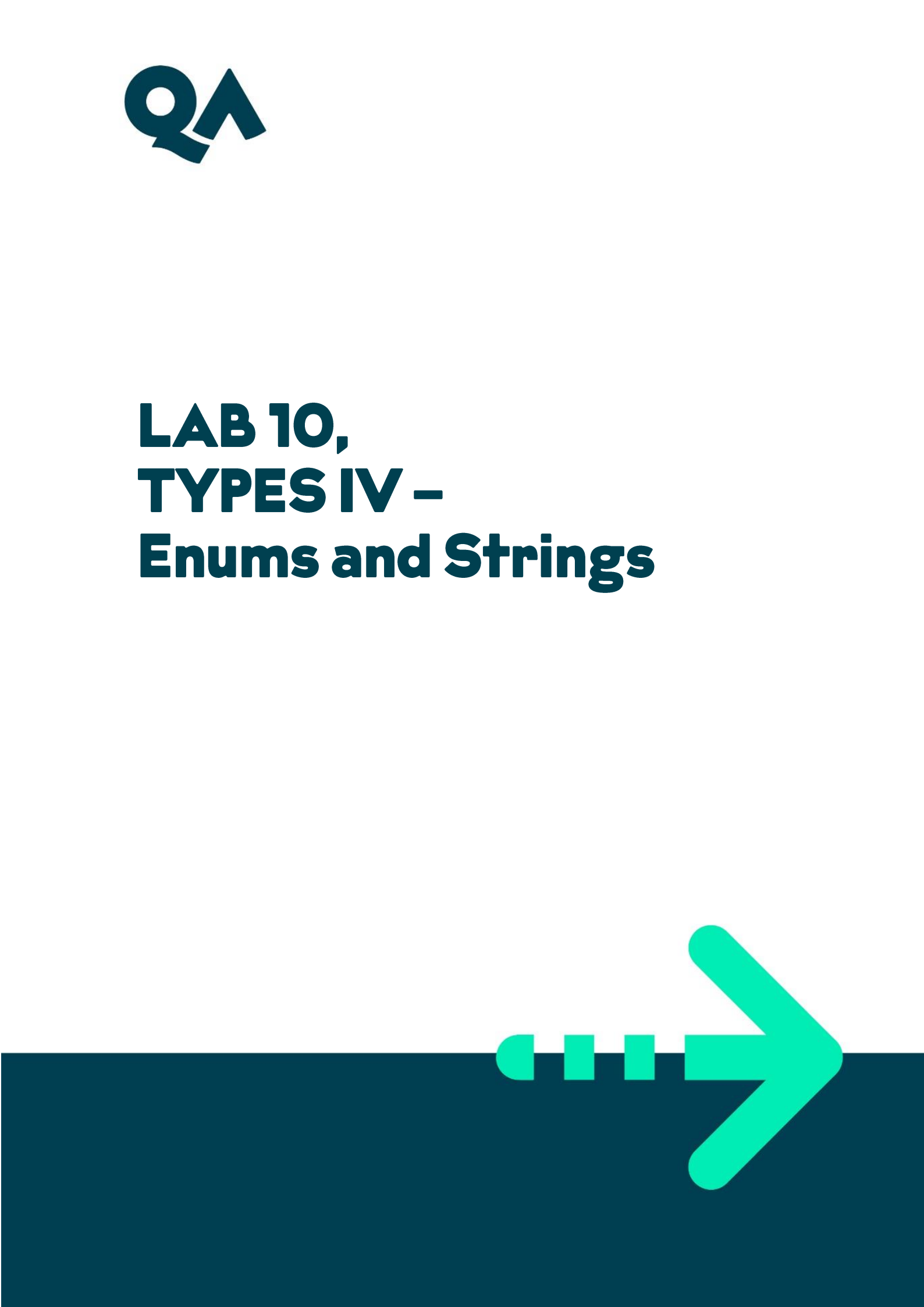

LAB 10, TYPES IV – Enums and Strings

Lab 10– Enums & Strings

Objective

See how to define and use the **`enum`** keyword to define a new type. Consolidate on knowledge of the functionality of class **`String`** and introduce the very useful **`StringBuilder`** class.

Part 1 using an enum

1. Open the Game project you created in Lab 9.
2. Circle is not the only shape! You can give the Ball class a property that dictates its shape. However the shape must be limited to a list which you define. You will need to create this type as an **enum**.
3. Define a new enum called SHAPE_TYPE with following values
   Rectangle,
   Ellipse,
   Pie

   Please create this *enum* outside of the Ball class or in its own file.
4. Now you can see the name **Ball** does not look like a good choice!
   Please change the name of this class to **Shape** in your project. The best way to do this is to open the Ball class and right mouse click on the word *Ball* and then choose the *Rename* menu options. The editor will change all references to Shape.
5. Define a new private field called shapeType of type SHAPE_TYPE
   as: `private SHAPE_TYPE shapeType;` (in the Shape class)
6. Create a method called GetShapeType() to return this field.
7. Set this value inside the constructor.
   Tip: Add a parameter of type SHAPE_TYPE to the constructor.
8. Back in the paint method, you can now examine the getShapeType() to see what to draw. for example

```
if ( b.GetShapeType() == SHAPE_TYPE.Rectangle)
                    e.Graphics.DrawRectangle(Pens.Red …);
```

9. Run your application to see different shapes bouncing about!

   You can also change the colour of your shape by creating a new property and then set it in the constructor of Shape as:

```
public Pen Colour { get; set; }
```

Part 2 using String

Please use the `MessageBox.Show(<string>);` statement to display a string in your Windows app;

1. Expand M`ain()` and declare a string called name whose value is any first name of any length greater than 3 characters.
2. Display its 3rd character using the [] array notation
   (can also be done with substring)
3. Display it converted to lowercase and to uppercase.
4. Use an enhanced `for` loop to iterate over its characters (use a simple foreach loop) and display each of them tab separated. Throw a line feed after this display.
5. Display whether it `StartsWith` a `String` of your choosing.
6. Display whether it `EndsWith` a `String` of your choosing.
7. Use the `IndexOf() method` to display the position in the `string` of the first occurrence of a character that you know is in the `string`, and also for a character that you know is not in the `string`.

Part 3- Using StringBuilder

1. Back in Main(), create a `StringBuilder` object called 'sb'. Use the constructor that allows you to initialise the object to contain the `String`

   "Bruce Springsteen<space>".

   (You can use the name of your favourite artist instead!)
2. Now use the `Append()` instance method of *`StringBuilder`* to append exactly the text "is the artist ever" (no error in that!).
3. Use the `ToString()` method of the `StringBuilder` to produce a string that you can display to see the current value of the `StringBuilder`.

   You are looking at a strange sentence that needs some amending.
4. Now we would like you to `Insert()` an adjective in front of the word "artist". Words like "greatest "obviously spring to mind, but make your own choice.
5. Now use the `Replace()` method of `StringBuilder` to replace the word
6. "artist" with a noun of your own choice. e.g "rock singer". Display the final result.

** End **