**Intro to Object Oriented Programming Using Python**

**Objects - Turtle race**

****

**You will need to use a Python Programming Environment to complete this task or you can use an online environment like Trinket (register so that you can save your work). The link for Trinket is:** [**https://trinket.io/features/python3**](https://trinket.io/features/python3)

Let’s have a go at writing a program using objects and to do this we are going to use Python’s turtle module

First, we need to ask Python to import the Turtle class, which is like a blueprint for making a turtle. To do this add the following code to a new Python file:



Now save your code as *yourName.py*. Eg. jamesh.py. Next, we will create an **instance** of a Turtle object. I’m going to name my Turtle object ‘mpc’ because I’m going to get my turtle to race against the rest of computing department. You can give your turtle whatever name you like.



Since the name of the Turtle object is a variable name, it must start with a letter and it cannot contain any spaces. In fact, we are creating a variable in exactly the same way as we usually do, except that the data type of the variable is not an integer or a string, but a Turtle! We need to give each Turtle object a different name, so that, when we give instructions, we can be specific about which object we are giving the instructions to.

Now, let’s tell our Turtle object what it should look like. Inside the object are ***attributes***, which are pieces of data we can define and change. The Turtle object has attributes for color and shape, so let’s use the color and shape methods to customise those attributes:



We can also tell our Turtle object what to do by calling other **methods**. With the code below, we are instructing the object to stop drawing with penup(), then to move to a location with goto(), and finally to get ready to draw a line with pendown().



Save your code and run it. What happens?

**Challenge**

Now it’s your turn. Create three more **instances** of a Turtle object, each with a different name. We don’t want to send all of the turtles to the same starting point, so tell one new turtle to goto(-160, 70), one to goto(-160, 40) and one to goto(-160, 10). You can set a different colour for each turtle if you like.

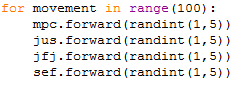
Save and run your code to check that each of your turtles positions itself correctly, ready to start the race!

**Creating the race…**

You will also need to go back to the top of your program and add this line of code so that we can generate random integers:



Now let’s add some code to make the Turtle objects race. Below your four turtle objects, add this code, replacing the names with the names of your own turtle objects:



Just as when we used methods to tell the turtle to penup(), pendown(), and goto(), with this code we are calling a method on each turtle object: we are asking it to move forward() a random distance between one and five units.

Save and run your code and see which turtle wins! The result should be different each time you run the code.

Congratulations – you have created your own turtle racing game!

Now try experimenting with your game. You could add additional turtles, change the speeds, see if you could edit the code to sometimes make the turtles travel backwards, keep count of how far each turtle has travelled and display the winner and change the attributes of the turtles.

**Expanding Learning Further**

There is a really good introduction to using OOP, in which you create the start of a text based game. All the concepts are also explained really well. The way you can access this course is to register join the course, hosted by FututeLearn. The link for the course is:

[**https://www.futurelearn.com/courses/object-oriented-principles#:~:text=This%20online%20course%20will%20introduce,other%20learners%2C%20and%20see%20theirs**](https://www.futurelearn.com/courses/object-oriented-principles#:~:text=This%20online%20course%20will%20introduce,other%20learners%2C%20and%20see%20theirs)**.**

This begins with the Turtle tutorial you have completed but then progresses on to the text based game.

**Best Preparation over the Summer…**

At A-Level we use C# as our main programming language therefore the best preparation you can make is to start learning the syntax for that language. This can be done by completing the tutorials on the left of the page, on the following link:

<https://www.tutorialspoint.com/csharp/index.htm>

You will need to either download Visual Studio Community Edition (<https://visualstudio.microsoft.com/downloads/>) to do this, or use an online programming environment if you unable to do so.

**Glossary**

**Attribute** – a named piece of data stored within an object

**Class** – a blueprint for making an object

**Instance** – a specific example of an object

**Method** – a function called upon an object allowing interaction with that object

**Object** – groups together data and functions to model something in code. Examples could include a physical item such as an LED, or a digital unit such as a bank account or an enemy in a video game

**Object-oriented programming** – a different style of programming from procedural programming which you have done at GCSE.