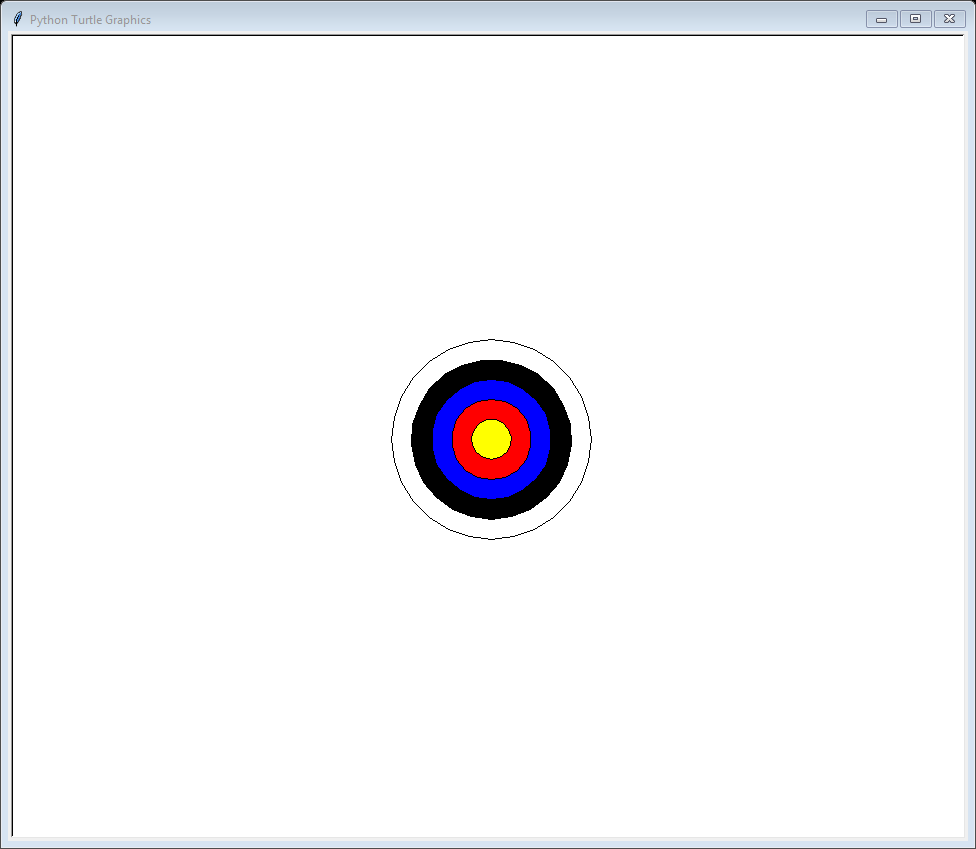
# Tutorial 04 – Functions

This tutorial is aimed at familiarising you with using and creating functions.

Do not worry if you can’t do all the exercises, especially the difficult ones. Give them a try and if you are stuck, ask your tutor.

For each exercise you should make a copy of the example you are working on **before you modify it**. In that way you will have both your new version and the original version to compare it with.

You should make a record all of your work in your COMP1753 logbook.

1. Download the examples and last week’s solutions. Check through the solutions and make sure you could do them all**.** Unzip this week’s examples.
2. Modify 05Calculator\_ifElifElse **from last week** so that it uses the **output()** and **input\_and\_convert()** functions from the lecture notes (also in 08Calculator\_inputAll).
3. Without looking at the lecture notes, fix examples 10, 11, 12, 13 & 15 to get rid of any errors and bugs. You should make a copy of the originals first.
4. Make a copy of 01Turtle\_squares and modify the **draw\_square()** function to use a **length** parameter which specifies the length of the sides of the square. Now modify the main body of the code to draw 4 squares of different sizes.
5. Make a copy of 17RandomValues\_validation **from last week**. Now create a function called **random()** which is passed two parameters, **minimum** and **maximum**, and returns a random number between those two values. If only one parameter is passed in then **maximum** should default to **100**. If no parameters are passed in then, in addition, **minimum** should default to **1**. Finally replace all the calculations of random numbers with a calls to **random()**, making sure that you test all three versions (no parameters, one parameter, two parameters).
6. Make a copy of 14Concessions\_orOperators from last week. Now create a function called **get\_message()** which constructs the message about pricing for the user. You will need to work out which parameters you will need to pass in to **get\_message()**. The function should return the message that it constructs, which will then be printed out in the main body of the code.
7. [Optional: if you are finding Python easy] Write a Turtle graphics function called circle\_at() which is given four parameters, **x**, **y**, **r** & **colour**, and draws a filled circle **centred** at (**x**, **y**) with a radius of **r** and the specified **colour**. To do this correctly you will need to call the function **setheading(0)** to set the direction to 0⁰ (your code will probably work without it but not if used in combination with other Turtle graphics commands). Test your function by drawing a bullseye like the one on the right.
8. Read the w3schools page mentioned in the lecture. We have covered everything there but it may you more insight into how functions work:
   1. <https://www.w3schools.com/python/python_functions.asp>