

## **EMAT10007 – Introduction to Computer Programming**

### **Exercises – Week 2:**

#### **Control Structures: Conditional Statements**

##### **Getting Started: Pycharm IDE**

###### **Open PyCharm on linux lab computers**

- Scroll down to bring up log in screen and log in with your UoB user name and password.
- Click activities (top left corner) to bring up the side panel.
- Click the grid of 9 dots to bring up applications.
- Choose JetBrains PyCharm
- When prompted about the user agreement click accept and read

##### **Create a new project and Python file**

- Click New project or File >> New project >> Pure python
- Unselect 'Create a main.py welcome script'
- Note the file location:  
`/home/UoB_username/PycharmProjects/your_projectname/venv`  
where **UoB\_username** is your UoB username and rename **your\_projectname** to be a name of your choice e.g. EMAT10007\_exercises
- Right click on the folder icon with project name next to it (top left of window).
- Choose new >> python file
- Give your file a name e.g. week\_1\_exercises.py

## **Write and run code**

Type some code and click the green play arrow at the top to run.

## **Save your project**

File >> Save all to save your work

## **Open a project you created previously**

Click File >> Open >> /home/**UoB\_username**/PycharmProjects/**your\_projectname**/venv,  
Open >> New window

## **Rules for naming variables**

- Variable names may contain letters or numbers
- Variable names must begin with a letter
- Variable names are case sensitive (`time` is not the same as `Time`)
- Some **keywords** are reserved by the Python language and cannot be used as variable names.  
For a full list of keywords reserved by Python, enter the following run the following command in the editor you are using:  
  
`help("keywords")`
- Use a consistent naming convention:
  - **snake\_case**: lower case letters, words separated by underscore (-)
  - **camel\_Case**: first letter of each word capitalised, excluding first word
  - **Pascal\_Case**: first letter of each word capitalised

## **Exercise 1 - Conditional Statements**

1. Create a variable, `a`, and assign it a string value. Use an `if` statement to test if the final character of the string is the letter `e`
2. Expand your answer to test if the final character of the string is the letter `e` or the letter `r`
3. Create a variable, `b`, and assign it a numerical value. Use an `if...else` control structure that tests if the number is zero or non-zero and print a message to say which is the case.
4. Write a program that outputs the name of the layer for a given depth from the earth's surface (Figure 1). The values shown in the figure are the maximum depth of each layer.

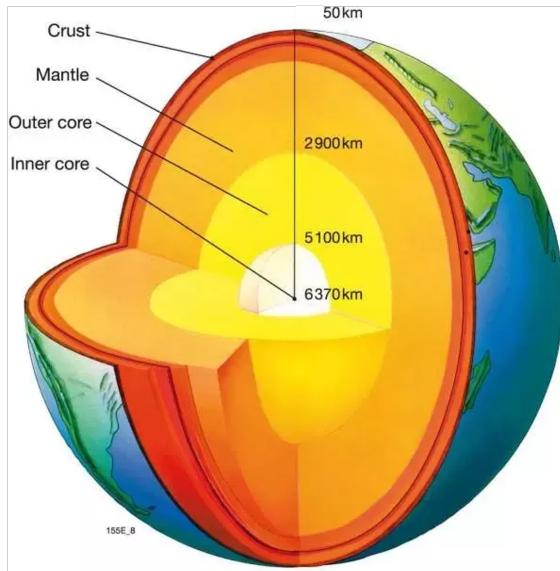


Figure 1: Depth from the earth's surface to the maximum depth of each layer.

### Exercise 2 - Currency Trading

A currency trader uses the following equation to calculate the amount in US dollars (USD) for the amount the customer pays in pounds sterling (GBP):

$$USD = GBP \times M \times R$$

where  $R = 1.38$  is the market rate and the multiplier,  $M$  is found using the table below, based on the amount paid.:.

GBP	Multiplier
< 50	0.9
< 500 and $\geq 50$	0.92
< 5,000 and $\geq 500$	0.95
< 50,000 and $\geq 5000$	0.97
$\geq 50,000$	0.98

Write a program that prints:

- the amount in US dollars for a given amount in pounds sterling
- the effective exchange rate for the conversion=  $\frac{USD}{GBP}$

### Exercise 3 - Control structures and floating point comparisons

1. Create three variables  $c=1.1$ ,  $d=3.2$ ,  $e=4.3$ .

Create a new float variable,  $f$ , with a value of your choice. Write a program that tests if  $f + 1.1$  is less than  $c$

2. Expand your program to check if which of the following conditions is satisfied by  $f + 1.1$ :
- less than c
  - greater than or equal to c but less than d
  - greater than or equal to d but greater than e
  - greater than or equal to e

and print a message to indicate which is True

### Exercise 4 - Nested conditionals

1. Write a program that:
  - checks if a number is odd or even.
  - if it is odd, checks if the number is a multiple of 3
  - if it is even, checks if the number is a multiple of 4
2. Write a program that checks a number,  $x$ , and prints:
  - ‘positive’ if  $x$  is positive
  - ‘negative’ if  $x$  is negative

If  $x$  is positive the program should **also** print:

- ‘square’ if  $x$  is a square number  
(a number of the form  $x = n^2$  where  $n$  is an integer)
- ‘not square’ otherwise

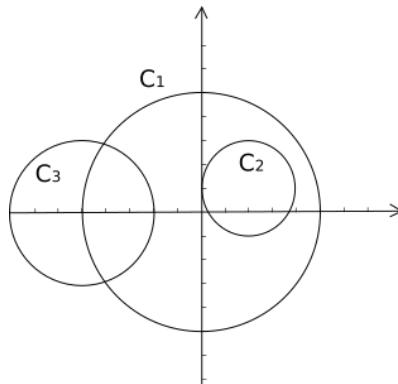


Figure 2: Overlapping circles  $C_1$ ,  $C_2$  and  $C_3$ .

### Exercise 5 - Putting it all together: Circles

Suppose we have three circles in the  $xy$ -plane (Figure 2).

Circle  $C_1$  is centred at  $(0, 0)$  with radius of length 5.

Circle  $C_2$  is centred at  $(2, 1)$  and has radius of length 2.

Circle  $C_3$  is centred at  $(-5, 0)$  and has a radius of length 3.

Write a program which takes in the variables  $x$  and  $y$  and tells the user which circle(s) the point  $(x, y)$  is in.

Think about the order in which your program evaluates the expressions? Is this the most efficient way to structure the code?