EMAT10007 – Introduction to Computer Programming

Exercises – Week 7. Functions

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 wor

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 comand 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 - Basics

1. Write a function called `print_name` that prints your name to the screen
2. Write a function that defines $a = 1$ and $b = 2$ and then prints the value to the screen. Be sure to give your function a sensible name.
3. What would be wrong with defining a single function whose goal is to add and multiply two numbers?

Exercise 2 - Functions with input and output

1. Can you complete the `compute_square` function below by replacing the blank `<?>` with the correct variable name?

```
def compute_square(number):
    # Returns the square
    squared = <?> ** 2
    return <?>
```

2. Use the `compute_square` function and a `for` loop to print the squares of all integers from 1 to 10.
3. Can you write a general function for raising numbers to powers? The function should take two arguments `number` and `power` and return the number raised to the given power. Remember to give your new function a sensible name.
4. Use your new general function to print the powers of 2 up to 2^{10}.

5. Write a new function which has arguments `number` and `powers`. If I have inputs `number=2` and `powers=[1, 3, 4]`, the function should return the list `[2, 8, 16]`.
6. Write a function called `inverse` that computes the inverse of a number x. That is, the function returns $1/x$ if $x \neq 0$. If $x = 0$ then your function should return the string "undefined". Show that your function works by computing `inverse(2)` and `inverse(0)`.
7. Write a function that swaps the values of two variables. Make sure your function returns the new values. Then, create two variables `a = ['red', 'blue', 'green']` and `b = {1, 3, 5}` and use your function to swap their values.

Exercise 3 - Functional programming

1. Create a Python file called shapes.py. Define a function that computes the area of a rectangle. The values of the width and height of the rectangle should be provided as arguments. The area should be returned.
2. Add more functions that compute (i) the area of a triangle and (ii) the area of an ellipse.
3. Add docstrings to each function to explain what they do.
4. Use your code to compute the area of the house-like shape that is obtained by placing a triangle with a base of 3 and a height of 1 on top of a rectangle of width 3 and height 2. *Answer = 7.5*

Exercise 4 - Numerical integration

Calculating the area under a curve given by $y(x)$ is important for many engineering problems. Calculus tells us that the area A under the curve $y(x)$ between the points $x = a$ and $x = b$ is given by the integral

$$A = \int_a^b y(x)\, dx. \tag{1}$$

However, in real-world engineering problems, the formulas for the curves are often so complex that it is impossible to carry out the integration exactly. Therefore, the area under the curve must be found approximately. One way to approximate the area under the curve is by the trapezium rule, which divides the area under a curve into N trapezoids. According to the trapezium rule, the area under the curve can be approximated by the formula

$$A \approx \frac{1}{2}[y(a) + y(b)]\Delta x + \sum_{i=1}^{N-1} y(x_i)\Delta x, \tag{2}$$

where $\Delta x = (b - a)/N$ and $x_i = a + i\Delta x$.

1. Write a Python program that uses the trapezium rule to calculate the area under the curve $y = x^2$ between $x = 0$ and $x = 1$ by setting $N = 1$, $N = 10$, and $N = 100$. Use a Python function to define the curve $y(x)$. In this case, the area can be calculated exactly as $A = 1/3$. Use this result to check that your code is working correctly. You should find that as N increases, the value of A calculated from the trapezium rule becomes more accurate. *Answers:* $A = 0.5$, 0.335, and 0.33335.

2. The density of a hot bar of length $L = 0.1$ m is given by the function

$$y(x) = \alpha e^{-x^2/\beta^2}, \tag{3}$$

where x is a point in the bar, $\alpha = 2$ kg m^{-1} and $\beta = 0.2$ m are constants. Assuming that $0 < x < L$, calculate the mass of the bar. Hint: the mass of the bar can be determined by calculating the area under the curve from $x = 0$ to $x = L$. Use the code `from math import *` to access the exponential function. This code should be added at the top of your Python file. The exponential of a variable `x` can be computed using the code `exp(x)`. *Answer: when* $N = 1000$ *the area is* $A \simeq 0.184512396$ *(which has units of kg).*

3. Write a Python function that calculates the area under any curve $y(x)$ between $x = a$ and $x = b$ using the trapezium rule. Hint: functions can be passed to other functions as arguments.

Exercise 5 - Advanced questions

1. Write a function that computes the median of three distinct numbers that are provided by the user. Recall that the median is the number in the middle; it's neither the largest nor the smallest. For example, the median of 4, 1, and 2 is 2.
2. Write a function called `sum_digits` that computes the sum of the digits of an integer. For example `sum_digits(1234) = 1 + 2 + 3 + 4 = 10`. There are several ways of doing this; however, if you want a challenge, try to do this using only mathematical operations.
3. Write a function which returns the prime factorization of a number. **Note:** Learn about prime factorization here: `https://www.mathsisfun.com/prime-factorization.html`