

General Instructions

Make sure that you read and then follow the instructions for each task carefully.

Please make sure that you save all of your work to a safe place and that you make regular back-ups.

You should begin all tasks with the following steps unless otherwise stated:

- Create a new Python file with the name specified in the task
- Save the file with your completed code (and/or any other files specified) in a folder named for the tutorial (*w01_t1*, *w01_t2*, *w02_t1*, *w02_t2*, etc.), which should itself be in a root folder called *sdam*.

If you are in any doubt about what you must do please ask your tutor.

Task 1 – Simple Calculator

Project Name: *simple_math*

Beginner

Step 1: Create a new subfolder in your *software_dev* folder called *w01_t2*

The following examples assume that you have created your *w01_t2* folder on your H drive.

Step 2: Create a new Python script file

- Create a new file called *simple_math.py* in your *w01_t2* folder
- Type the following code into your file

```
age = 2
print(age)
age = 21
print(age)
age = age + 1
print("Age next Birthday: " + str(age))
```

Step 3: Run your program

- Run the program and make sure that you are getting the output that you are expecting.

Step 4: Modify your code

- Change the type of *age* to a *float*.

Step 5: Run program

- Run the program again and the output should be different (if it isn't different then ask your tutor to help you to identify the problem)
- Why would the output be different?

Step 6: Take a screenshot of the output

- Take a screenshot of the application window showing the program output
- Paste the screenshot into Paint and save it in your project folder as *simple_math.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file *simple_math.py*
- *simple_math.jpg* from Step 6

Task 2 – Breakeven**Project Name: breakeven**

Beginner

The break-even point for a business is defined as the point where the total revenue equals the business expenses (fixed costs). The following values are required to calculate this:

- What is the cost to produce each item
- What is the sale price per item (item cost + profit per item)
- What are the fixed costs

Dividing the fixed costs by the profit per item (the difference between the cost to produce each item and the sale price) gives the number of items that must be sold before "breaking even".

For example, if it costs £50.00 to produce an item which is then sold at £100.00, there is £50.00 profit per item. If the fixed costs are £1000.00, the company would need to sell 20 items to break even.

Exercise data:

- Cost to produce each item = £20.00
- The sale price per item (item cost + profit per item) = £40.00
- Fixed costs = £50000.00

Write a program to output the data with a meaningful label for each item and calculate and output the number of items we need to sell to breakeven. The output should look similar to the following (cropped for convenience).

```
#BreakEven
C:\Users\pcw1\AppData\Local\Programs\Python\Python35-32\py
Cost to produce each item: 20.0
Sale price for each item: 40.0
Fixed costs: 50000.0
Profit per item: 20.0
Breakeven: 2500.0 items.

Process finished with exit code 0
|
```

$$\text{Hint: } \text{break_even} = \frac{\text{fixed_costs}}{\text{sale_price} - \text{item_cost}}$$

Step 1: Analyse the problem

- Decide what type of data is being used
- Using the notes given in Lecture 2 analyse the problem

Step 2: Create a Python script file

- Create a new file called *breakeven.py*
- Make sure that you put it in your *w01_t2* folder

Step 3: Write source code

- Using your analysis from Step 1, write code to produce the required outputs
- Ensure that you use appropriate names for all your variables

Step 4: Run and test your application

- Run your application with the example data to ensure that you get the correct answer
- Identify and correct any errors in your source code if you don't get the output that you were expecting

Step 5: Take a screenshot of the output

- Press [ALT + Print Screen] for your output screen
- Paste your screenshot into Paint and save as *breakeven.jpg*

Portfolio

The task contribution to your portfolio is:

- The Python source code file *breakeven.py*
- *breakeven.jpg* from Step 5

Task 3 – Basic data types

Project Name: *datatypes*

Beginner

Create a text document called *datatypes.txt*

State the most appropriate data type for the following data with a brief justification for your decision:

- A company's profit
- A person's weight
- A book title
- Average rainfall in September
- Number of runs scored by a cricketer
- A telephone number, including area code
- A dress size

Example:

a) float – because the profit number may contain decimal places

Save your file in your *w01_t2* folder.

Portfolio

The task contribution to your portfolio is:

- The completed *datatypes.txt* file

Task 4 – Sweet Tooth

Project Name: *sweet_tooth*

Intermediate

A teacher has bought a packet of 40 sweets that she is going to share out equally between her 14 students. Because a single sweet cannot be shared, the teacher will keep what is not given to the children.

The teacher will keep the minimum number of sweets possible (obviously this scenario is not based on real life).

Write a program to determine and output the number of sweets each child will receive. As a single sweet cannot be shared between pupils, the program must calculate and output the number of sweets per child and the number that the teacher keeps for herself. **The calculation must be done using the remainder (%) operator.**

Take a screen shot of the finished output saved as *sweet_tooth.jpg* in your *w01_t2* folder.

Portfolio

The task contribution to your portfolio is:

- The Python source code file *sweet_tooth.py*
- *sweet_tooth.jpg* showing output from *sweet_tooth.py*

Task 5 – Coffee Pseudo-Code

Project Name: *make_coffee*

Beginner

Create a text file called *make_coffee.txt*.

Using the notes from the lecture, write the pseudo-code to represent the process of making a cup of coffee.

You can make the pseudo-code as simple or as complex as you like as long as it:

- adequately represents the process of making a cup of coffee, and
- makes use of the pseudo-code notation that was described in the lecture.

Portfolio

The task contribution to your portfolio is:

- The text file containing your pseudo-code *make_coffee.txt*

Task 6 – Cans of Paint

Project Name: *cans_of_paint*

Intermediate

To help you complete this task you should use the Python *math* module. You can consult the online documentation to see what methods this will make available to you:

<https://docs.python.org/3.0/library/math.html>

To use a module within your program you must import it using the `import` keyword. By convention, import statements occur at the beginning of a file.

```
import math
```

You will need some of the methods from that module.

For example, to call the `ceil()` method, you type:

```
math.ceil(a_number_or_variable)  
e.g.: math.ceil(num)
```

Write a program that uses methods from the `math` module to solve the following problem.

A can of paint covers 5.1 m² of wall. Each can of paint has a diameter of 15 cm and a height of 30 cm. The shop that sells the paint helps customers by packing the cans into boxes with internal measurements of 0.60 x 0.30 x 0.35 metres (L x W x H).

Solve and output the following:

1. The minimum number of cans that must be bought to paint the walls of a hall whose floor measures 40 x 30 metres, and whose

ceiling is 3.4 metres above the floor (use an appropriate method from the `Math` class).

2. The number of full boxes given to the customer who buys this quantity of paint (use an appropriate method from the `Math` class rather than integer division).
3. The number of cans not packed into boxes.

Portfolio

The task contribution to your portfolio is:

- The Python source code file *cans_of_paint.py*
- *cans_of_paint.jpg* showing output from *cans_of_paint.py*

All portfolio requirements for this tutorial

Beginner

simple_math.jpg

simple_math.py

breakeven.jpg

breakeven.py

datatypes.txt

make_coffee.txt

Intermediate (opt)

sweet_tooth.jpg

sweet_tooth.py

cans_of_paint.jpg

cans_of_paint.py