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 *Project Name* specified in the file.
- It is suggested that you save the file with your completed code (and/or any other files specified) in a folder named for the tutorial week (*week01, week02,* etc.), which should itself be in a root folder called *sdam*. However, you can use whatever method you wish to organise your work – just make sure that each file can be located quickly and efficiently based on the week number and the project name.

If you are in any doubt, please check with your tutor.

Initials
Project Name: initials

Beginner

Write a program to prompt the user to enter a single line containing their name (first name and family name) and then the program will output their initials.

Use only one input operation to get the full name but you can use *split()* to separate their first name and family name.

Step 1: Write your source code

- write the code necessary to prompt the user for the required data, and output their initials.

Step 2: Run, test and screenshot your application outputs

- run the program and input appropriate data
- take a screen shot of your output, call your screenshot *initials.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file for the task
- *initials.jpg* showing output from *initials.py* when tested.

Character Print
Project Name: character_print

Beginner

Write a program that prompts the user to enter a positive number and a text character and prints the character the number of times entered by the user.

The print code should be written inside a function that has the following:

Purpose	Prints a character a specified number of times.
Identifier	character_print
Parameters	The character to print, An integer for the number of times to print
Return Type	void

When complete test your program with suitable data. Take a screen shot of the output and save it as a file in your project folder called *character_print.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file for the task
- *character_print.jpg* showing output from *character_print.py* when tested.

Functional Calculator
Project Name: fun_calc

Beginner

Write a program that prompts the user to enter two integers (either may be positive or negative). The program will then display the following menu:

1. Add

2. Subtract

3. Multiply

4. Divide

5. Remainder

After the user selects the operation, the program will perform the operation on the two entered integers and output the result.

Each arithmetic operation should be implemented using a function.

The generic format for each operation name is as follows:

Purpose	Perform indicated operation (i.e. if the function is the Add function it adds the two integers) on the two integers and output the result.
Identifier	Appropriate to the operation.
Parameters	The two operands.
Return Type	void

When complete test your program and each operation with suitable data. Take a screen shot of the output from each operation and save it as a file in your project folder called *add_function.jpg*, *subtract_function.jpg*, *multiply_function.jpg*, *divide_function.jpg* and *remainder_function.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file for the task
- *add_function.jpg*, *subtract_function.jpg*, *multiply_function.jpg*, *divide_function.jpg* and *remainder_function.jpg* showing output from *fun_calc.py* when tested.

Swap
Project Name: swap

Beginner

Write a program that prompts the user to enter two integers (either may be positive or negative). The program will store the two numbers in a list and print the list. It will then swap the two numbers in the list and print the swapped list.

An example output from the program:

```
Enter two integers: 5 9 You entered: 5 9 Your entries
swapped: 9 5
```

Both the swap and print functionality should be written as functions.

The format for the swap function:

Purpose	Swaps the elements of a 2 element integer list.
Identifier	swap
Parameters	An integer list.
Return Type	void

The format for the print function:

Purpose	Prints the elements of an integer list separated by a space.
Identifier	print_int_list
Parameters	A list.
Return Type	void

When complete test your program with suitable data. Take a screen shot of the output and save it as a file in your project folder called *swap_ints.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file for the task
- *swap_ints.jpg* showing output from *swap.py* when tested

Recursive Reversion
Project Name: reversal

Beginner

Write a program that prompts the user to enter a string. The program should then reverse the string and print out the string in reverse. This should be completed using a recursive function. The reversed string should be printed in a single line

An example output from the program:

```
Enter a string to be reversed: I love recursion
noisrucer evol I
```

When complete test your program with suitable data. Take a screen shot of the output and save it as a file in your project folder called *reversal.jpg*.

Portfolio

The task contribution to your portfolio is:

- The Python source code file for the task
- *reversal.jpg* showing output from *reversal.py* when tested

You sank my battleship!
Project Name: destroyer

Expert

You may or may not be familiar with the old children's game Battleships (if not see https://en.wikipedia.org/wiki/Battleship_(game)).

You are going to create a simplified version.

In your program you will have a 5 x 5 game grid on which you will place a Destroyer which will take up two adjacent squares either vertically or horizontally (see Fig. 1 for example).

Your program will then display a representation of the gameboard (see Fig. 2) and prompt the user to enter a coordinate for a "shot", for example row 1 column 3. Using a function it will check to see if the shot is a hit or a miss and display an appropriate message (Fig. 3.).

The game will continue until the player "sinks" the destroyer by guessing the two positions of the ship (Fig. 4.).

```
  1  2  3  4  5      1  2  3  4  5
1[ ][ ][ ][ ][ ]  1[ ][ ][ ][ ][ ]
2[ ][D][D][ ][ ]  2[ ][ ][ ][ ][ ]
3[ ][ ][ ][ ][ ]  3[ ][ ][ ][D][ ]
4[ ][ ][ ][ ][ ]  4[ ][ ][ ][D][ ]
5[ ][ ][ ][ ][ ]  5[ ][ ][ ][ ][ ]
  Gameboard 1         Gameboard 2
```

Fig. 1. Examples of ship placement on the gameboard

```
  1  2  3  4  5
1[ ][ ][ ][ ][ ]
2[ ][ ][ ][ ][ ]
3[ ][ ][ ][ ][ ]
4[ ][ ][ ][ ][ ]
5[ ][ ][ ][ ][ ]
```

Fig. 2. Example of the initial displayed gameboard.

```
Enter a shot (r c): 1 3
  1  2  3  4  5
1[ ][ ][ ][ ][ ]
2[ ][ ][ ][ ][ ]
3[M][ ][ ][ ][ ]
4[ ][ ][ ][ ][ ]
5[ ][ ][ ][ ][ ]
Sorry you missed.
 Enter a shot (r c): 2 3
  1  2  3  4  5
1[ ][ ][ ][ ][ ]
2[ ][ ][H][ ][ ]
3[M][ ][ ][ ][ ]
4[ ][ ][ ][ ][ ]
5[ ][ ][ ][ ][ ]
Well done you hit!
```

Fig. 3. Example of the displayed gameboard after player has taken a shot.

```
Enter a shot (r c): 2 2
  1  2  3  4  5
1[ ][ ][ ][ ][ ]
2[ ][H][H][ ][ ]
3[M][ ][ ][ ][ ]
4[ ][ ][ ][ ][ ]
5[ ][ ][ ][ ][ ]
Glug, glug, glug...you won.
```

Fig. 3. Example of the displayed gameboard after the player has won.

Write a suitable test plan and save it as *destroyer_test_plan.docx*.

When complete test your program using your test plan. Take a screen shot of the output when a user misses, hits and completes a game save your screenshots as *misses.jpg*, *hits.jpg* and *wins.jpg*.

Portfolio	The task contribution to your portfolio is: • The Python source code file for the task • Your test plan *destroyer_test_plan.docx* • *misses.jpg*, *hits.jpg* and *wins.jpg* showing output from *destroyer.py* when tested.

Multiple Initials
Project name: multi_init

Expert

This task is similar to Task 1, except that your program should accept user input of any number of names. Your program is to ignore any extra spaces, whether they be at the leading, trailing, or in the middle. Hyphenated names should result in hyphenated initials.

```
User input: " Joseph David Kingsley-Montgomery "
```

```
Initials output: J.D.K-M.
```

Step 1: Write your source code

- write the code necessary to prompt the user for the required data and output the initials - you must use at least two string methods in your code.

Step 2: Run, test and screenshot your application outputs

- run the program inputting appropriate data
- take a screen shot of your output, call your screenshot *multi_init.jpg*.

Portfolio	The task contribution to your portfolio is: • The Python source code file for the task • *multi_init.jpg* and *new_entry.jpg* showing output from *multi-init.py* when tested.

All portfolio requirements for this tutorial

Beginner	*initials.jpg* *initials.py* *character_print.jpg* *character_print.py* *add_function.jpg* *subtract_function.jpg* *multiply_function.jpg* *divide_function.jpg* *remainder_function.jpg* *fun_calc.py* *swap_ints.jpg* *swap.py* *reversal.jpg* *reversal.py*
Expert (opt)	*misses.jpg* *hits.jpg* *wins.jpg* *destroyer_test_plan.docx* *destroyer.py* *multi_init.jpg* *multi_init.py*