e-Yantra MOOC: Software Foundation (Part I)

Week 5: Assignment 1
Power of N

[*Last Updated on:* ***05th May 2021, 14:00 Hrs***]

- Aim
- Given
- Procedure
    - `raise_power_of_N()`
- Expected Output
- Grading and Submission Instructions
- References

Aim

Your old mathematician friend was not much happy with your previous help in finding Euclidean distance, where he had to provide the values of X and Y co-ordinates of two points in the program manually each time and then run to check the output.

In his words: "*Hey, no doubt you are good at programming, but can I provide the values to your program at runtime? If yes, then can you help me with that. This time I want you to find the square of a number given at runtime and also it should provide cube of it* ***only*** *when I ask for it. Can you do it my friend?*"

What do you think? Is that possible with Python?

Given

One file is provided to solve this assigment.

- Skeleton program file: `assignment1.py`
    - The skeleton consists of one function which you have to modify:
        - `raise_power_of_N()`

Procedure

- Open the skeleton program file, `assignment1.py` .
- You will notice pre-written comments included in skeleton program for your assistance to solve the assignment.
- One function to modify is:

    - `raise_power_of_N()`

Function Name	`raise_power_of_N()`
Purpose	Computes square of positional argument (N) and cube as well if optional argument 'c' or 'cube' is passed.
Input Arguments	None
Output Arguments	None
Example Call	***raise_power_of_N()***

- The various parts of the program are as follows:
    - **description** of program: `Calculate square and/or cube of N`
    - **epilog** of program: `Week-5 Assignment-1`
    - **disallow the abbreviations** for optional arguments
    - **positional argument**: `N`

      This is an *integer number* input to your program at the runtime provided through command line.

        - **helper message**: `input the number`
    - **optional argument**: `c` or `cube`

      Either of 'c' or 'cube' when passed, they should be stored as Boolean value of **True**.

        - **helper message**: `calculate cube of N as well`

> **NOTE:** All of the above logic should be written ***ONLY*** inside the `raise_power_of_N` function under the block of:
>
> *####### ADD YOUR CODE HERE ########*

- To run and debug your solution, type the below command in Terminal:

```
$ python3 assignment1.py 5

-OR-

$ python3 assignment1.py 5 -c

-OR-

$ python3 assignment1.py 5 --cube
```

  This command will run the Python script `assignment1.py`.

- Refer the **Expected Output** section below and debug your code to get the correct output.

Expected Output

- The output of the program when the command is `python3 assignment1.py -h` OR `python3 assignment1.py --help` is:

```
usage: assignment1.py [-h] [-c] N
```

```
Calculate square and/or cube of N

positional arguments:
  N           input the number

optional arguments:
  -h, --help  show this help message and exit
  -c, --cube  calculate cube of N as well

Week-5 Assignment-1
```

- When the command is: **`python3 assignment1.py 5`**

```
Square of N: 25
```

- When the command is: **`python3 assignment1.py 5 -c`** OR **`python3 assignment1.py 5 --cube`**

```
Square of N: 25
Cube of N: 125
```

- When the command is: **`python3 assignment1.py 5 --c`** OR **`python3 assignment1.py 5 --cu`** OR **`python3 assignment1.py 5 --cub`**

```
usage: assignment1.py [-h] [-c] N
assignment1.py: error: unrecognized arguments: --c

-OR-

usage: assignment1.py [-h] [-c] N
assignment1.py: error: unrecognized arguments: --cu

-OR-

usage: assignment1.py [-h] [-c] N
assignment1.py: error: unrecognized arguments: --cub
```

- When the command is: **`python3 assignment1.py -c`** OR **`python3 assignment1.py --c`** OR **`python3 assignment1.py --cu`** OR **`python3 assignment1.py --cub`**

```
usage: assignment1.py [-h] [-c] N
assignment1.py: error: the following arguments are required: N
```

- When the command is: **`python3 assignment1.py`**

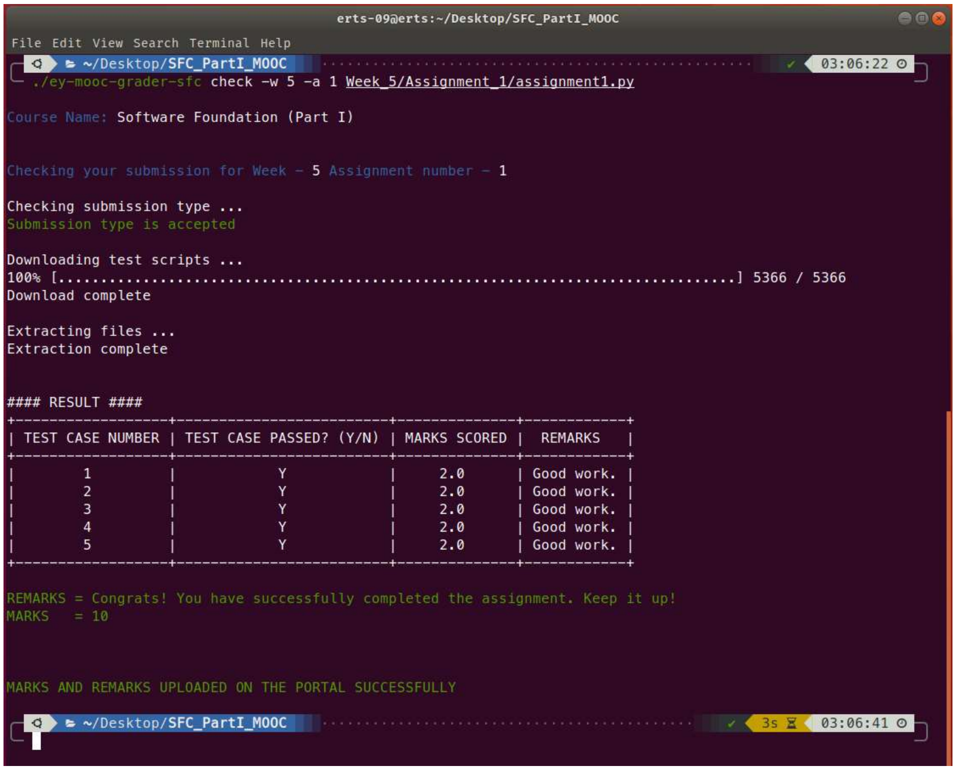
```
usage: assignment1.py [-h] [-c] N
assignment1.py: error: the following arguments are required: N
```

Grading and Submission Instructions

- Navigate to the folder where the ***ey-mooc-grader-sfc*** application resides.
- To grade your solution, run the **`check`** command of the application as follows:

```
$ ./ey-mooc-grader-sfc check -w 5 -a 1 Week_5/Assignment_1/assignment1.py
```

- This will run your program **`assignment1.py`** against random test cases and grade it. Marks and appropriate remarks will be provided as shown in Figure 1.
- Your program file **`assignment1.py`**, marks scored and remarks will get uploaded to the MOOC portal.

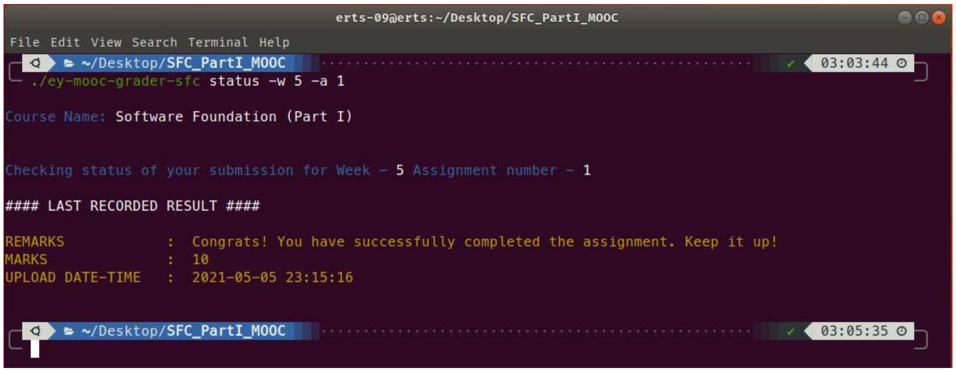

erts-09@erts:~/Desktop/SFC_PartI_MOOC
File Edit View Search Terminal Help

```
~/Desktop/SFC_PartI_MOOC ............................................ ✓ 03:06:22
./ey-mooc-grader-sfc check -w 5 -a 1 Week_5/Assignment_1/assignment1.py

Course Name: Software Foundation (Part I)


Checking your submission for Week - 5 Assignment number - 1

Checking submission type ...
Submission type is accepted

Downloading test scripts ...
100% [..................................................................................] 5366 / 5366
Download complete

Extracting files ...
Extraction complete


#### RESULT ####
+------------------+-------------------------+---------------+------------+
| TEST CASE NUMBER | TEST CASE PASSED? (Y/N) | MARKS SCORED |  REMARKS   |
+------------------+-------------------------+---------------+------------+
|        1         |            Y            |      2.0      | Good work. |
|        2         |            Y            |      2.0      | Good work. |
|        3         |            Y            |      2.0      | Good work. |
|        4         |            Y            |      2.0      | Good work. |
|        5         |            Y            |      2.0      | Good work. |
+------------------+-------------------------+---------------+------------+

REMARKS = Congrats! You have successfully completed the assignment. Keep it up!
MARKS   = 10


MARKS AND REMARKS UPLOADED ON THE PORTAL SUCCESSFULLY

~/Desktop/SFC_PartI_MOOC ...................................... ✓ 3s ⧗ 03:06:41
```

Figure 1: Output of running check command for Week 5 Assignment 1

- You can verify this by running the **`status`** command of the application as given below, refer Figure 2.

```
$ ./ey-mooc-grader-sfc status -w 5 -a 1
```


erts-09@erts:~/Desktop/SFC_PartI_MOOC
File Edit View Search Terminal Help

```
~/Desktop/SFC_PartI_MOOC ............................................ ✓ 03:03:44
./ey-mooc-grader-sfc status -w 5 -a 1

Course Name: Software Foundation (Part I)


Checking status of your submission for Week - 5 Assignment number - 1

#### LAST RECORDED RESULT ####

REMARKS            :  Congrats! You have successfully completed the assignment. Keep it up!
MARKS              :  10
UPLOAD DATE-TIME   :  2021-05-05 23:15:16


~/Desktop/SFC_PartI_MOOC ............................................ ✓ 03:05:35
```

Figure 2: Output of running status command for Week 5 Assignment 1

References

- Official Python documentation of Argparse module
- Blog on Command Line Interfaces with Argparse by RealPython

All The Best!