

# INFS1201-Computer Program

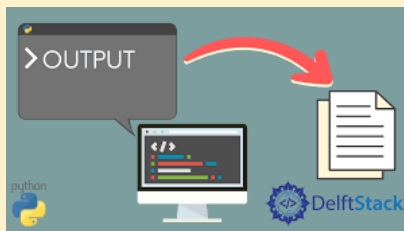
## Week 13 Lab 10

Submit all exercises!

Always start your code by the following comments:

```
##  
#firstName lastName - StudentID  
#Week13 lab Lab 10  
#Exercise XXX
```

### Key Information for Successful Lab Completion



Up to this point, we've used the **print** function to show our program output on the screen. However, in this lab, we'll transition to using **file.write** to print the output to file. It's important to note that the algorithm itself will remain unchanged; the only modification will be in the part where the function displays its results. To see the difference, run the below the following code.

**# Example using print to display output on the screen**

```
print("Hello, World!")
```

**# Example using print to display output to a file**

**# Open the file in write mode**

with open('test.txt', 'w') as file:

**# Write the result to the file**

    file.write("Hello, World!")

**#after running the code, open the file test.txt to see the output.**

## Exercise 1

We learned how to make text files in Python. Because text files have so many uses, they are much more powerful than you might think. Because text files have so many uses, they are much more powerful than you might think. One interesting use is to create a basic webpage, and in this exercise, we'll make a simple webpage displaying a colored rectangle.

Write a function `createRectangleHtml(fileName,width,height, color)` which has the following parameters:

- `fileName`: the name of the file, should have the extension `.html`
- `width`: an integer representing the width of the rectangle you will draw
- `height`: an integer representing the width of the rectangle you will draw
- `color`: a string representing... the color of the rectangle

Your function's output should generate a file with the correct filename and identical content to `exercise1.html`. However, the width, height, and color will vary according to your input parameters. To test the function, open the resulting file in a browser and observe its appearance.

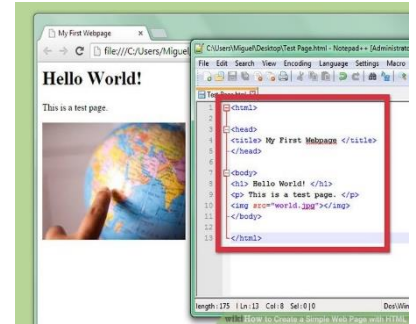
Hints:

- The content inside the file serves as a sample run, open the file in Notepad to view the contents.
- The function will write the output to a file using the **file write method** instead of presenting it on the screen with the **print** function.

Test the function using the below code:

```
>>> createRectangleHtml('myfirstwebsite.html',500,100, 'blue')
```

Click on the resulting file `myfirstwebsite.html` to open it in the browser and check the appearance.



## Exercise 2

GPA again...but this time with files!

Create a function `updateTranscript(fileName)` with a unique parameter which is the name of the file.

This function **adds** to the file the grades of a new semester. It should request the semester name, then the number of courses taken, then for each course, the course code, the number of credits of the course and the grade of the course. It appends the semester, then one course on each line with the three information values separated by a comma (this is a CSV file).

A sample run of the function is as follows:

```
Which semester are you adding? Spring 2022
How many courses did you take in Spring 2022? 2
Course code #1: INFS1301
```

```
Number of credits of INFS1301: 3
Grade in INFS1301: B+
Course code #2: SCIE1001
Number of credits of SCIE1001: 3
Grade in SCIE1001: C+
```

The file exercise2.csv shows an example of the result of calling the function twice on the same file exercise2.csv. Open the file in Notepad to view the contents. The content inside the file serves as a sample run.

Test the function using the below code:

```
>>> updateTranscript('mytranscript.csv')
```

Click on the resulting file mytranscript.csv to open it in Excel and check the results.

### Exercise 3

Write a function computeGPA(filename) that takes a transcript such as the one generated in exercise 2 and computes the corresponding GPA.

```
>>> computeGPA('exercise2.csv')
3.15625
```

Hints:

- To identify lines containing course code, credit, and grade, rely on splitting every line you read by commas and utilize the **len** function to distinguish these lines from others.
- You need to create a function that converts letter grade to point grade.

### Exercise 4

GPA once more!

Some students retake courses to improve their GPA, where only the best grade in a repeated course is counted. Additionally, grades marked as 'W' are excluded from the GPA calculation.

Modify the function from exercise 3 to consider these two particular cases, then create a file corresponding to your actual transcript and calculate your GPA!

```
>>> computeGPA('exercise4.csv')
3.09375
```