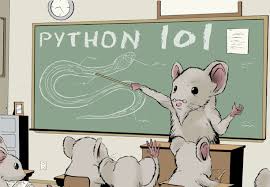
Python intro

**

Date: *Feb 2020*  
Version 1.1

# *First reader*

This document is a guideline to Practical assignment of programing course. Make sure you read this document carefully to know how you should deliver your assignments.

How to deliver your assignments?

1. Create a git repository and submit your assignments every week in the repository. Make sure you use this pattern for naming the repository: <**StudentNumber-StudentFullName-Feb22-Python101**>. Any other repository name will not be accepted!
2. Work on the weekly assignment individually and make sure you have at least one commit per week. Any week without commit means you didn’t work on the programming skill in that week.
3. Make a proper folder structure in your repo. For example: every week will have a folder with the same name and deliverables will be placed there. There are 3 different type of file delivered, make sure these files does not contain unnecessary information:
   1. Python code
   2. Documentation
   3. Screenshots
4. Add Readme file with information of the repo, student name and number, folder structure, etc.
5. Add proper commit messages including week number. For example, a commit message could look like: “week1- initial commit: created the repo and added Readme.md file”

*A close up of a map

Description automatically generated*

#### *Assignments 1 – Setup the environment & Python interpreter*

Task1.1 Setting up the environment

Python is an interpreted programming language and also is cross platforms, means you can code and use python in any type of operating system. You can make a decision in which operating system you want to install and code.

For windows users this link will be useful: [https://docs.python.org/3/using/windows.html#windows-store](https://docs.python.org/3/using/windows.html)

1. Download and Setup the Python interpreter and test it with your operating system’s default command prompt. Make sure you download the latest version of Python interpreter, Currently 3.9.2 . (use this link <https://www.python.org/>)
2. Install and use the **Visual studio code** IDE to code. You are free to use any other IDE or Text editor but configuration and making sure that the deliverable does not include extra files, will be your own responsibility. (you can download it from Microsoft Azure Dev tools or via this [link](https://code.visualstudio.com/download) )

A screenshot of a computer screen

Description automatically generated

Task 1.2

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-2_5.png. Estimated time: 30 minutes.

Run the python interpreter from command prompt and check the start-up message and answer below questions. Take screenshots of the results of each command.

1. What is the version of your python?



1. What is the release date?



1. What is the full command to get help?



1. Show the list of all keywords.

A screenshot of a computer

Description automatically generated with medium confidence

1. Show the list of all Module

Timeline

Description automatically generated

#### *Assignments 2 – Variables and Operators*

Task 2.1

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 30 minutes.

Answer the following questions by running the same code in your interpreter. Provide screenshots.

2.1.1 What is the value of variable *result* after running following commands?

**>>>** result = 2 + (17 - 5\*6) / 4

A picture containing text, device, gauge, meter

Description automatically generated

2.1.2 What is the value of variable *result* after running following commands?

**>>>** width = 20

**>>>** height = 2 \*\* 5

**>>>** width \* height

Graphical user interface, text

Description automatically generated

2.1.3 What is the last value printed in the interpreter? And what is the meaning of \_ in the last line?

**>>>** tax = 21.5 / 100

**>>>** price = 1989.95

**>>>** price \* tax

**>>>** price + \_

Graphical user interface, text, application, chat or text message

Description automatically generated

2.1.4 What is the type of variable result after running following code?

**>>>** result = 21.5 / 100

A picture containing text, device, gauge

Description automatically generated

2.1.5 What is the difference between the following code? And what is each one output?

**>>>** 23 / 3

**>>>** 23 // 3

A picture containing text, road, device, meter

Description automatically generated

Task 2.2

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 30 minutes.

Answer the following questions by running the same code in your interpreter. Provide screenshots.

2.2.1 What will be the outputs in the code below?

**>>>** s = 'First line.\nSecond line.'

**>>>** s

**>>>** print(s)

Graphical user interface, text

Description automatically generated

2.2.2 What will be the output in the code below?

**>>>** print('C:\programs\name')

**>>>** print(r'C:\programs\name')

Text

Description automatically generated

2.2.3 produce the following output:

Usage: myscript [OPTIONS]

-h Display the help message

-U username assign a username

2.2.4 following code has an error. What is the error? And how you can fix it to print “Mikaeil Shaghelani”?

**>>>** firstName = 'Mikaeil'

**>>>** firstName 'Shaghelani'

Graphical user interface, text

Description automatically generated

2.2.5 what will be printed on the screen?

**>>>** word = 'Programming Basic'

**>>>** word[0]

**>>>** word[10]

**>>>** word[-3]

**>>>** word[-12]

**>>>** word[0:6]

**>>>** word[0:11] + word[0:11]

Text

Description automatically generated

#### *Assignments 3 – conditions & loops*

Task 3.1

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-2_5.png. Estimated time: 30 minutes.

3.1.1 create a script that asks for an integer value from user and evaluate the number and print a proper message on screen if it’s positive or negative number.

3.1.2 create a script that asks user to press one letter key. If the inserted character is lower case your app must print the capital form of the character and vice versa.

3.1.3 create a 3rd script, this time the app will ask for integer value and evaluate it (checks if it’s positive or negative). The program must keep asking the next number. User can press ‘Q’ to exit the app.

3.1.5 create a script that ask user to input several numbers (can be integer or decimal) and when user entered ‘0’, it will print the sum, average, minimum and maximum of all previous entered values ( the last 0 should not be included in the calculation).

3.1.6 create a script which asks for your weight and height and calculates BMI (Body Mass Index) and print a proper message on the screen based on the BMI value.

BMI = [weight (kg) / height (cm) / height (cm)] x 10,000

**BMI Categories:**

Underweight = <18.5

Normal weight = 18.5 – 24.9

Overweight = 25 – 29.9

Obesity = BMI of 30 or greater

#### *Assignments 4- functions*

Task 4.1

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 30 minutes.

4.1.1 Create a simple calculator by defining 4 basic functions

1. sum = accepts 2 numeric values, a and b, and returns the sum of 2 values
2. sub = accepts 2 numeric values, a and b, and returns the subtraction of b from a
3. multi = accepts 2 numeric values, a and b, and returns the multiplication of 2 values
4. divide = accepts 2 numeric values, a and b, and returns the division of a by b

use these functions and do 10 mathematical operations with different values (integer and decimal) and investigate the results. Provide screenshots of the results.

4.1.2 how can you improve your calculator to be able to perform the operation with 3 values? Make it possible to use the functions with 2 or 3 value. Provide screenshot of the working calculator.

Task 4.2

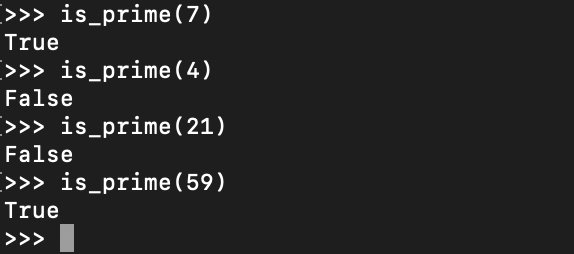
Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 45 minutes.

4.2.1 Create a function that prints the following results in command line.

A close up of a logo

Description automatically generated

4.2.2 Create a function that checks if the given number is prime or not and return a Boolean value as a result. Like example below:



4.2.3 Create another function to print all prime number from 0 to a given number.

Hint: make use of the function you created in the previous assignment. The output should look like this:

*A close up of a logo

Description automatically generated*

#### *Assignments 5- Data structure*

Task 5.1

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 30 minutes.

5.1.1 What is the outputs printed in the interpreter? Provide screenshot.

**>>>** marks = ['Outstanding', 'Good', 'Unsatisfactory', 'Good', 'Poor']

**>>>** marks.count('Poor')

**>>>** marks.count('Good')

**>>>** marks.index('Good')

**>>>** marks.index('Good', 2

**>>>** marks.reverse()

**>>>** marks

**>>>** marks.append('Satisfactory')

**>>>** marks

**>>>** marks.sort()

**>>>** marks

**>>>** marks.pop()

**>>>** marks.pop(2)

**>>>** marks

A computer screen capture

Description automatically generated with medium confidence

5.1.2 Write a Python program to check if a list is empty or not and displays a proper message.

5.1.3 Create a script that asks user to input several names and when user inputs an empty name script must stop and print first a list of all input values in the same order and print them in the “Descending order” (from Z to A).

5.1.4 Write a Python program to remove duplicates elements from the given list:

Example [1,2,2,2,3,1,4,2,3,6,7,9,7,3]

5.1.5 Create a program that asks user for an integer input and prints squares values from 0 to the given number as a list.

5.1.6 Create a program that accepts a list of values and removes all the odd values in the list and then display the original list and the list without odd values.

Example: originalList = [2, 5, 7, 4 , 99, 20, 11, 3 ]

manupulatedList = [2, 4, 20 ]

#### Assignments 6 - Class

Task 6.1

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-2_5.png . Estimated time: 15 minutes.

Create a class named *Person* that has:

1. instance variable: name
2. function: getName, must return the current value of name instance variable
3. function: setName, must get a new name as parameter and update the name variable
4. constructor: that sets the name instance variable on the invoke time.

Task 6.2

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-2_5.png. Estimated time: 20 minutes.

6.2.1 Extend the task 4.1 by creating a class named Calculator that is capable of processing 2 and 3 values operation.

6.2.2 add a new function to Calculator class named linearEquation with 3 parameters for value of a, b and c in the equation. The function must solve the equation by finding and returning the value of x.

Tip: Linear equation example: a x+ b = c

Task 6.3

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 30 minutes.

Design a simple application for your ToDo list by creating a class with the same name and following requirements:

1. must store a list of string values representing the items of your to-do list
2. a function to add a new to-do to your list
3. a function to remove an existing item from your list. In case the item does not exist, print a proper message for user.
4. A function to check if an item exists in your to-do list or not. The proper message must be printed.
5. A function to print the items in the list on screen
6. A function that returns the to-do list of string

Task 6.4

Difficulty: C:\Users\874156\Desktop\flatastic-icons-part-1-by-custom-icon-design\png\16x16\star-3_5.png. Estimated time: 15 minutes.

Extend the task 6.1 by improving the Person class:

1. Add an instance object named myToDoList from the type ToDo

Create an object from Person class and make sure you can add and remove items from the to-do list

Challenge:

Assign a unique number automatically to items in your to-do list start from 100.