Programming Fundamentals

Lab Manual - Week 14

Introduction

Welcome Back to your favorite Programming Lab students. In this lab manual, we shall work together to complete our Business Applications and Game Project along with their documentations and also practice Python Programming Language.

Task 01(CP): Business Application in C++

1. Complete your business application with file handling.

Task 02(CP): Game Project in C++

1. Complete your game project with gotoxy and getCharAtXY function (2D arrays optional).

For Practice in Python

Setting up Python Environment

The first thing you need to do, to learn Python for Data Science, is to install Python and a suitable Integrated Development Environment (IDE) for coding in Python. At the end of this manual, you will be able to start coding in Python.

Setting up Python

1. Download Python Executable File

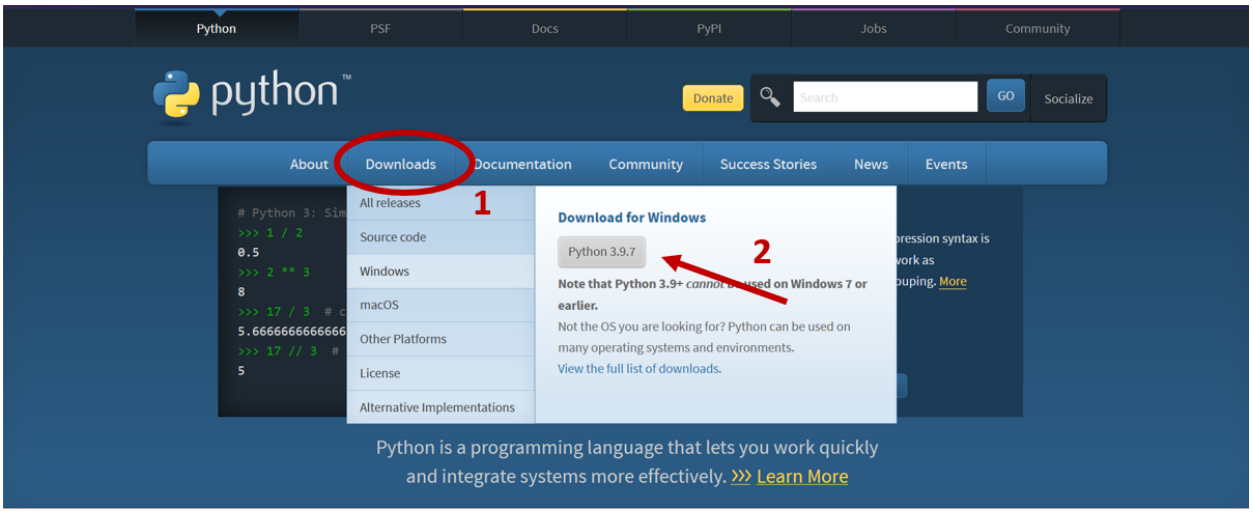
To initiate Python installation on your system, you first have to download and run the Python.exe file. For this, navigate to the Python official website and download the latest version of Python.

a. Open your web browser (Chrome, Firefox, etc.)

b. Go to the website: https://www.python.org/

c. Go to the downloads for windows section

d. Click on Python 3 and download 'Windows installer (64-bit)'. At the time of the download, the file was 27.5 MB.



2. Running Python Executable File

a. Now, go to the directory where you have downloaded the file and open it. (OR press '**ctrl + j**' on your browser, it will show your downloads, double click on the installer).

b. Click on '**Install Now**' with the shown settings.

c. After installation, click on the **'Close'** button. Your Python installation is successful.

3. Check Installation

a. To verify your installation, go to the command prompt (type **'cmd'** in the Search box).

b. Now type, "**Python –V**" to check the version of Python installed.

Now, open .py file in Visual Studio code.

Task 03(CP): Get the Area of a Country

Create a function that takes a country's name and its area as arguments and returns the area of the country's proportion of the total world's landmass.

Examples

area_of_country("Russia", 17098242) ➞ "Russia is 11.48% of the total world's landmass"
area_of_country("USA", 9372610), "USA is 6.29% of the total world's landmass"
area_of_country("Iran", 1648195) ➞ "Iran is 1.11% of the total world's landmass"

Notes

- The total world's landmass is 148,940,000 [Km^2]

Task 04(CP): Length of Number

Create a function that takes a number num and returns its length.

Examples

number_length(10) ➞ 2
number_length(5000) ➞ 4
number_length(0) ➞ 1

Notes:

The use of the len() function is prohibited.

Task 05(CP): Cricket balls in Over

In cricket, an over consists of six deliveries a bowler bowls from one end. Create a function that takes the number of `balls` bowled by a bowler and calculates the number of **overs** and **balls** bowled by him/her. Return the value as a float, in the format `overs.balls`.

Examples

total_overs(2400) ➞ 400
total_overs(164) ➞ 27.2 # 27 overs and 2 balls were bowled by the bowler.
total_overs(945) ➞ 157.3 # 157 overs and 3 balls were bowled by the bowler.
total_overs(5) ➞ 0.5

Notes

The number following the decimal point represents the balls remaining after the last over. Therefore, it will only ever have a value of 1-5.

Task 06(CP): Oddish and Evenish

Create a function that determines whether a number is **Oddish** or **Evenish**. A number is **Oddish** if the sum of all of its digits is odd, and a number is **Evenish** if the sum of all of its digits is even. If a number is **Oddish**, return "Oddish". Otherwise, return "Evenish".

For example, oddish_or_evenish(121) should return "Evenish", since 1 + 2 + 1 = 4. oddish_or_evenish(41) should return "Oddish", since 4 + 1 = 5.

Examples

oddish_or_evenish(43) ➞ "Oddish"
4 + 3 = 7
7 % 2 = 1

oddish_or_evenish(373) ➞ "Oddish"
3 + 7 + 3 = 13
13 % 2 = 1

oddish_or_evenish(4433) ➞ "Evenish"
4 + 4 + 3 + 3 = 14
14 % 2 = 0

Task 07(CP): Likes vs Dislikes

YouTube currently displays a like and a dislike button, allowing you to express your opinions about particular content. It's set up in such a way that you cannot like and dislike a video at the same time.

There are two other interesting rules to be noted about the interface:

1. Pressing a button, which is already active, will undo your press.
2. If you press the like button after pressing the dislike button, the like button overwrites the previous "dislike" state. The same is true for the other way round.

Create a function that takes in a list of button inputs and returns the final state.

Examples

like_or_dislike(["Dislike"]) ➞ "Dislike"
like_or_dislike(["Like", "Like"]) ➞ "Nothing"
like_or_dislike(["Dislike", "Like"]) ➞ "Like"

like_or_dislike(["Like", "Dislike", "Dislike"]) → "Nothing"

Notes

- If no button is currently active, return "Nothing".
- If the list is empty, return "Nothing".

Task 08(CP): Design Rugs

Write a function that accepts the height and width (`m, n`) and an optional **proc** `s` and generates a list with `m` elements. Each element is a string consisting of either:

- The default character (hash `#`) repeating `n` times (if no **proc** is given).
- The character passed in through the proc repeating `n` times.

Examples

```
make_rug(3, 5, '#') → [
  "#####",
  "#####",
  "#####"
]

make_rug(3, 5, '$')  → [
  "$$$$$",
  "$$$$$",
  "$$$$$"
]

make_rug(2, 2, 'A')  → [
  "AA",
  "AA"
]
```

Task 09(CP): Storing Data in File

Take names from user and store into the file in python

Task 10(CP): Reading Data from the File

Read names from the file and print on screen in python