Mazyar Kazemi

Nov. 19th, 2019

Foundation of programming in Python

Assignment07

# Assignment 07

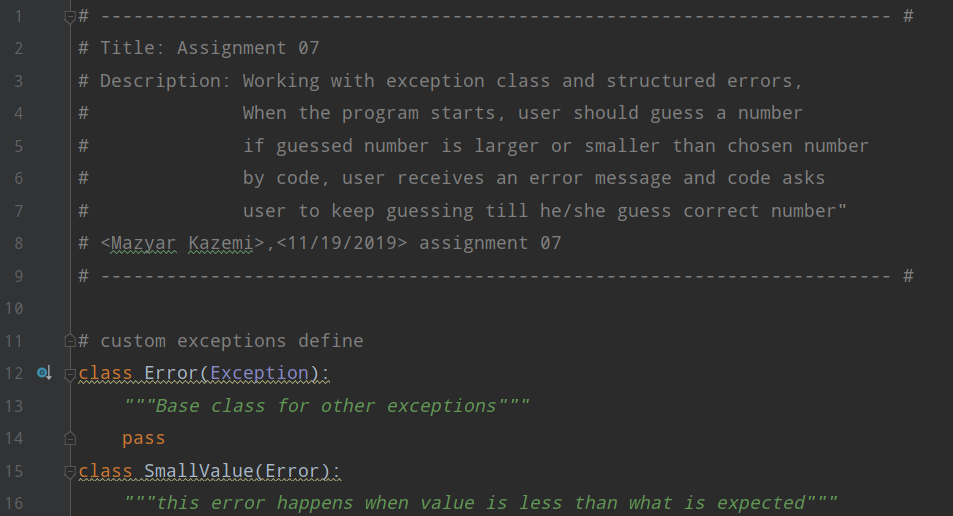
## Introduction

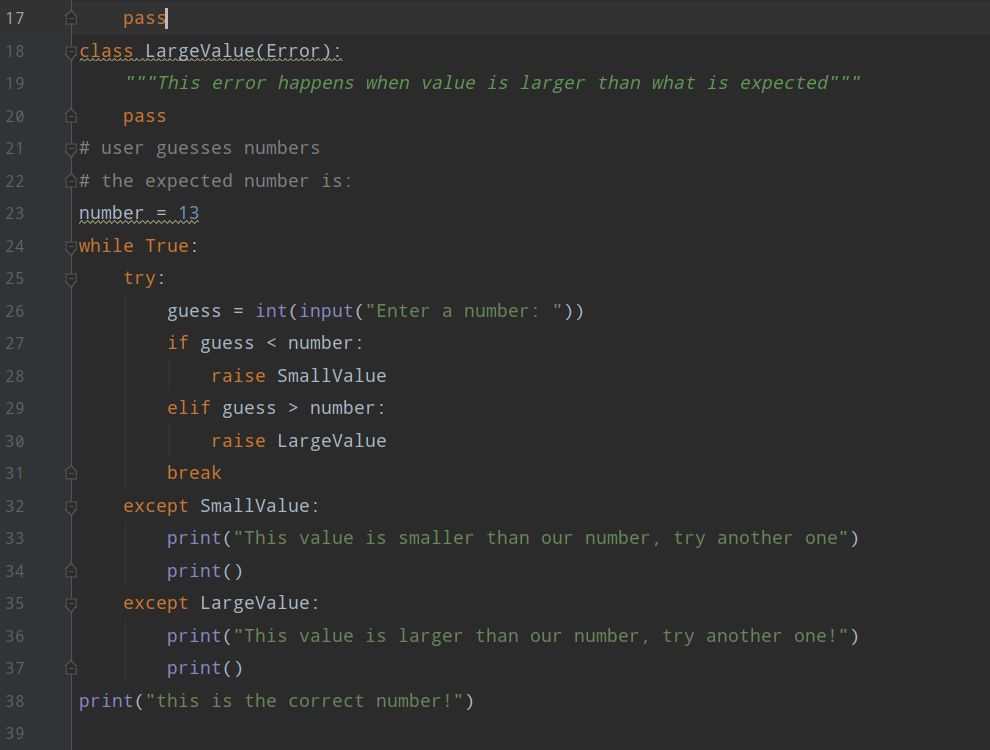
In this week assignment I present my Python program then I explain how my script demonstrate pickling and structured error handling work. Later, I’ll come up with a summary of what I learned from this week of course.

### Create a Python Script

Create a new script that demonstrates structured error:

This is my Python script:





Following is the running result of above script in PyCharm and Command Prompt as following:

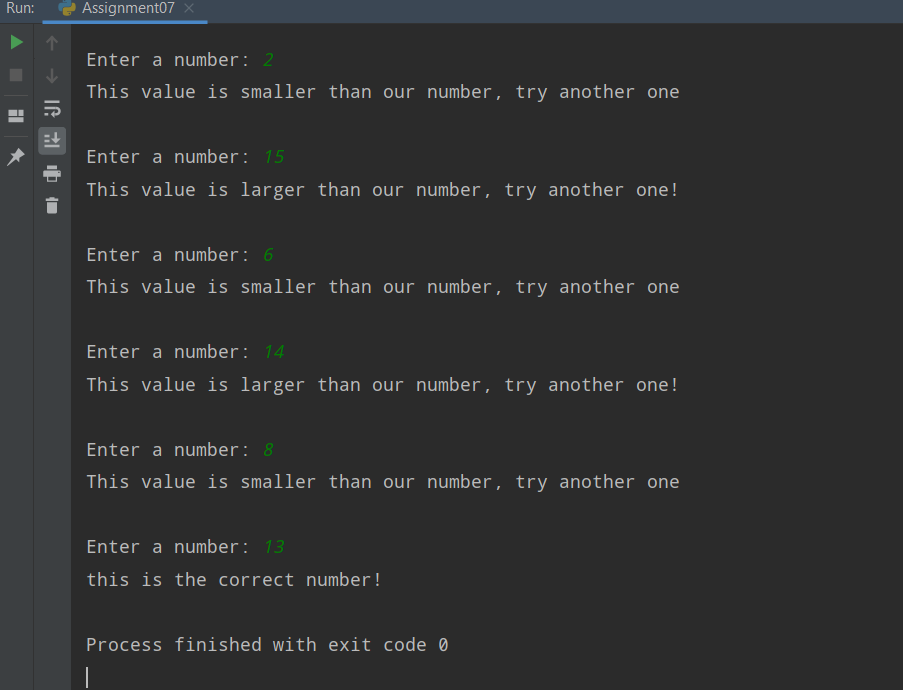


Figure 2. A screenshot of the script running in PyCharm

### 

Figure 4. A screenshot of the script running in Command Prompt

### Script Description

* Lines 12, 15 and 18, Base Class exception and user defined class exception (**SmallValue** and **LargeValue)** are defined.
* Line 23, chosen number, which should be guessed by user is defined.
* Lines 26 to 31, **while** loop is defined.
* Lines 32 and 35, completed the structured error definition.
* Line 38, printing “This is the correct number!” when user picks the correct number after all.

### Summary of what is learned in this module

* Using built-in Python commands into function would help to reduce repeating code in script body, splitting a difficult problem in to several simplified parts, improving simplicity of code and make it convenient to reuse the code.

* Using structured error handling helps end users, and sometimes even other developer to deal with a simplified and clear to understand error messages rather than with complicated computer language.
* A text files is made of alphabets, digits, and other characters by storing their ASCII values which is understandable by human. A binary file however, is not readable by human since it is formed by a collection of bytes only understandable by computers.
* Classes are the category of errors. Each class has some sub-class exceptions which is called exception class hierarchy. Therefore, when we want to use an exception in our code, we should know the expected error belongs to which category (class) and which sub category. For example, **ArithmeticError** have three different sub-classes as **FloatingPointError, OverflowError, ZeroDivisionError.**
* Though Python has many built-in exceptions, sometimes it is required to create a custom exception per our need. Users can define exception by creating a new class. This custom exception must be derived directly or indirectly from exception class. Most of the built-in exceptions are derived from Exception class.
* When a specific exception class is required and it doesn’t exist among Python built-in exception class, we can create a class derived from Exception class.
* **Markdown** is a lightweight markup **language** with plain text formatting syntax. ... **Markdown** is often used to format readme files, for writing messages in online discussion forums, and to create rich text using a plain text editor.
* For using Markdown on a GitHub webpage first, we create a new GitHub repository, then create a “docs” folder with a file called “filename.md”. Then we type in some simple markdown commands to format our document. Then will file it in repository. With using “Preview” tab we can see our new markdown file and edit it.