**Python Assignment 8**

Q1. Is the Python Standard Library included with PyInputPlus?

Ans. No, the Python Standard Library is not included with PyInputPlus. PyInputPlus is a third-party module that provides additional input validation and handling functionality beyond what is available in the Python Standard Library. However, PyInputPlus does depend on some modules from the Python Standard Library, such as sys and re.

Q2. Why is PyInputPlus commonly imported with import pyinputplus as pypi?

Ans. PyInputPlus is commonly imported with import pyinputplus as pypi to provide a shorter and more convenient alias for the module name. Using pypi as an alias makes it quicker to type when using the module's functions, and it also makes the code more readable by reducing the amount of repetition in the module name.

By convention, the alias pypi is used to avoid conflicts with other modules that might have similar names. Additionally, the use of an alias makes it easier to switch to a different module or library in the future without having to modify every instance of the module name in the code.

Q3. How do you distinguish between inputInt() and inputFloat()?

Ans. inputInt() and inputFloat() are two different functions in the PyInputPlus module that can be used to prompt the user for integer and float input, respectively. The main difference between the two functions is the type of input they accept.

inputInt() function only accepts integer input from the user. If the user enters a value that cannot be converted to an integer, an error message will be displayed, and the function will prompt the user to enter a valid integer.

inputFloat() function, on the other hand, accepts float input from the user. If the user enters a value that cannot be converted to a float, an error message will be displayed, and the function will prompt the user to enter a valid float. Additionally, inputFloat() can also accept scientific notation and accepts both integer and float values.

In summary, inputInt() is used to accept only integer input, while inputFloat() is used to accept both integer and float input, including scientific notation.

Q4. Using PyInputPlus, how do you ensure that the user enters a whole number between 0 and 99?

Ans. To ensure that the user enters a whole number between 0 and 99 using PyInputPlus, you can use the inputInt() function with several keyword arguments.

Here's an example code snippet that uses inputInt() function to achieve this:

import pyinputplus as pypi

num = pypi.inputInt(prompt="Please enter a whole number between 0 and 99: ", min=0, max=99)

In the above code, inputInt() function is called with a prompt message that asks the user to enter a whole number between 0 and 99. The min and max arguments are used to set the minimum and maximum values that the input can be. By setting min=0 and max=99, we ensure that the user input is restricted to the range of integers from 0 to 99.

If the user enters a value outside of this range or a non-integer value, inputInt() will raise a ValueError exception and ask the user to enter a valid integer.

Q5. What is transferred to the keyword arguments allowRegexes and blockRegexes?

Ans. The allowRegexes and blockRegexes keyword arguments in PyInputPlus functions are used to specify regular expressions that determine which input strings are allowed and which are not. allowRegexes is a list of regular expressions that allow certain input strings to pass validation. If the user's input matches any of the regular expressions in allowRegexes, the input will be accepted. blockRegexes is a list of regular expressions that block certain input strings from being accepted. If the user's input matches any of the regular expressions in blockRegexes, the input will not be accepted and an error message will be displayed.

Both allowRegexes and blockRegexes accept a list of regular expressions as input, which are compiled into regular expression objects that can be used for matching input strings.

For example, if you want to accept input strings that only contain uppercase letters and numbers, you can use allowRegexes to allow those specific characters as follows:

python

Copy code

import pyinputplus as pypi

input\_str = pypi.inputStr(prompt="Enter uppercase letters and numbers only: ",

allowRegexes=[r'^[A-Z0-9]+$'])

In the above code, the regular expression r'^[A-Z0-9]+$' matches input strings that contain only uppercase letters and numbers. The allowRegexes argument is used to pass this regular expression to inputStr() function to allow such input strings.

Q6. If a blank input is entered three times, what does inputStr(limit=3) do?

Ans. If a blank input is entered three times while using the inputStr(limit=3) function in PyInputPlus, the function will raise a TimeoutException exception.

The limit keyword argument in the inputStr() function sets the maximum number of attempts the function will allow before raising a TimeoutException exception. In this case, the limit is set to 3, which means that the user will be prompted for input up to three times.

If the user fails to provide any input within the first three attempts, or if they only provide blank input on all three attempts, then the TimeoutException exception will be raised.

Here's an example code snippet that demonstrates this behavior:

import pyinputplus as pypi

input\_str = pypi.inputStr(prompt="Please enter some text: ", limit=3)

# if the user enters blank input on all three attempts, a TimeoutException will be raised.

In the above code, if the user enters a blank input on all three attempts, then the function will raise a TimeoutException. If the user provides valid input before the third attempt, the function will return the user's input without raising any exceptions.

Q7. If blank input is entered three times, what does inputStr(limit=3, default='hello') do?

Ans. If blank input is entered three times while using the inputStr(limit=3, default='hello') function in PyInputPlus, the function will return the default value of 'hello'.

The limit keyword argument in the inputStr() function sets the maximum number of attempts the function will allow before returning the default value. In this case, the limit is set to 3, which means that the user will be prompted for input up to three times.

The default keyword argument in the inputStr() function sets the default value that will be returned if the user fails to provide any input after the specified number of attempts. In this case, the default value is set to 'hello'.

Here's an example code snippet that demonstrates this behavior:

import pyinputplus as pypi

input\_str = pypi.inputStr(prompt="Please enter some text: ", limit=3, default='hello')

# if the user enters blank input on all three attempts, the function will return 'hello'.

In the above code, if the user enters a blank input on all three attempts, then the function will return the default value of 'hello'. If the user provides valid input before the third attempt, the function will return the user's input without using the default value.