Python Basics Assignment 8

```
!pip install PyInputPlus
```

```
Collecting PyInputPlus
  Downloading PyInputPlus-0.2.12.tar.gz (20 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Collecting pysimplevalidate>=0.2.7
  Downloading PySimpleValidate-0.2.12.tar.gz (22 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Collecting stdiomask>=0.0.3
  Downloading stdiomask-0.0.6.tar.gz (3.6 kB)
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
    Preparing wheel metadata: started
    Preparing wheel metadata: finished with status 'done'
Building wheels for collected packages: PyInputPlus, pysimplevalidate, stdiomask
  Building wheel for PyInputPlus (PEP 517): started
  Building wheel for PyInputPlus (PEP 517): finished with status 'done'
  Created wheel for PyInputPlus: filename=PyInputPlus-0.2.12-py3-none-any.whl size=11297
sha256=83db9b812c14ebe959b9a3ff2c12ee22575b099c5cdd45dcf356fe4a47fa4060
  Stored in directory:
c:\users\mihir\appdata\local\pip\cache\wheels\b4\6e\2f\8a852732646cabec36c3fe8fc060ec5bea1c1be7114
32c47f7
  Building wheel for pysimplevalidate (PEP 517): started
  Building wheel for pysimplevalidate (PEP 517): finished with status 'done'
  Created wheel for pysimplevalidate: filename=PySimpleValidate-0.2.12-py3-none-any.whl size=16175
sha256=37501718896711134a4d2974a73d4e6a35e1f6d07d517f5c01d67646ae11fba0
  Stored in directory:
c:\users\mihir\appdata\local\pip\cache\wheels\b1\44\4a\043a4f4c4512c7cdfb0c2b8408b18b0de5fd45cac5
7f5dfa02
  Building wheel for stdiomask (PEP 517): started
  Building wheel for stdiomask (PEP 517): finished with status 'done'
  Created wheel for stdiomask: filename=stdiomask-0.0.6-py3-none-any.whl size=3306
```

```
sha256=7a9495433ee53fb7b22f10f7851eb5f14f9ca6d7406b60265fe914fe19b6291c
  Stored in directory:
c:\users\mihir\appdata\local\pip\cache\wheels\1d\aa\47\f41f117d22c5de82e95d9342f44da578c80610739a
2d5ebec4
Successfully built PyInputPlus pysimplevalidate stdiomask
Installing collected packages: stdiomask, pysimplevalidate, PyInputPlus
Successfully installed PyInputPlus-0.2.12 pysimplevalidate-0.2.12 stdiomask-0.0.6
```

1. Is the Python Standard Library included with PyInputPlus?

Ans: No, **PyInputPlus** is not a part of Python Standard Library, it needs to be installed explicitly using the command **!pip install PyInputPlus**

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

Ans: You can import the module with **import pyinputplus as pypi** so that you can enter a shorter name when calling the module's functions.

```
import pyinputplus as pypi
```

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

Ans: inputInt() function Accepts an integer value. This also takes additional parameters **min**, **max**, **greaterThan** and **lessThan** for bounds. And it always returns an int.

Whereas **inputFloat()** function Accepts a floating-point numeric value. this also takes additional **min**, **max**, **greaterThan** and **lessThan** parameters. and always returns a float.

```
inp = pypi.inputInt(prompt = "Enter an Integer... ",
                    default = 0, limit = 3)

print(inp)

inp2 = pypi.inputFloat(prompt = "Enter Float value... ",
                    default = 0, limit = 3)

print(inp2)
```

```
Enter an Integer... 1
1
Enter Float value... 23.22
23.22
```

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

Ans: PyInputPlus module provides a function called as **inputInt()** which only returns only integer values. inorder to restrict the input between 0 and 99, i'ii use parameters like **min** & **max** to ensure that user enters the values between the defined range only.

```
import pyinputplus as pyip
wholenumber = pyip.inputInt(prompt='Enter a number: ', min=0, max=100)
print(wholenumber)
```

```
import pyinputplus as pyip
wholenumber = pyip.inputInt(prompt='Enter a number: ', min=0, max=100)
print(wholenumber)
```

```
Enter a number: 12
12
```

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

Ans: we can use **allowRegexes** and **blockRegexes** keyword arguments to take list of regular expression strings to determine what the pyinputplus function will reject or accept valid input.

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

Ans: The statement **inputStr(limit=3)** will throw two exceptions **ValidationException** and **RetryLimitException**. The first exception is thrown because blank values are not allowed by inputStr() function by default. it we want to consider blank values as valid input, we have to set **blank=True**.

The second exception is occured because we have reached the max limit we have specified by using **limit** parameter. inorder to avoid this exception we can use **default** parameter to return a default value when max limit is reached.

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

Ans: Since the default parameter is set to **hello**. after blank input is entered three times instead of raising **RetryLimitException** exception. the function will return **hello** as response to the calling function