UDITI NAMDEV SE-B 60004180111

IBM Python Exercise – 3 Sys Module

1)code:

```
import sys

print(sys.version)

print(sys.getwindowsversion())
```

output:

```
3.7.6 (default, Jan 8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
sys.getwindowsversion(major=10, minor=0, build=17763, platform=2,
service_pack='')
```

Explanation: sys.version : This returns a string that gives information about the version of the Python interpreter, the Build no. and the compiler used.

sys.getwindowsversion : It returns a named tuple describing the windows version which is currently running.

2)code:

```
import sys

a = sys.stdin

print("You write: ",a.readline())

print("This is the msg after newline")
```

output:

```
Hello
You write:  Hello


This is the msg after newline
```

Explanation:

This function will take something from the user means it takes input from users like input() function. It is the standard way for interactive input in Python. The function sys.stdin will automatically append a newline after the input string.

3)code:

```
import sys
```

```
c = input()

sys.stdout.write(f"You write: {c}")

print("This is the msg after newline")
```

output:

```
Hello
You write: HelloThis is the msg after newline
```

Explanation: sys.stdout :

This function will write something on the console or to the file. It is the standard way for output in Python. It will not do any formatting to the output text string, unlike the print() function which automatically
inserts whitespace between each argument and a newline at the end.

4)code:

```
import sys

print(sys.executable)
```

output:

```
C:\Users\Uditi\anaconda3\python.exe
```

Explanation:

This returns a string that contains the absolute path of Python Interpreter Executable Binary on the system.

5)code:

```
import sys

print(sys.version_info)
```

output:

```
sys.version_info(major=3, minor=7, micro=6, releaselevel='final', serial=0)
```

Explanation:

This will return a tuple that contains the information about the Python Version only.