LAB # 11

MODULES AND PACKAGES

EXERCISE

A. Point out the errors, if any, and paste the output also in the following Python programs.

1. Code:

```
import sys as s
print(sys.executable)
print(sys.getwindowsversion())
```

Output:

Sys is written instead of s.

```
>>> %Run cp2.py
  Traceback (most recent call last):
    File "E:\Semester 1\P fund\LAb 11\cp2.py", line 2, in <module>
      print(sys.executable)
  NameError: name 'sys' is not defined
```

2. Code:

```
import datetime
from datetime import date
import times
# Returns the number of seconds
print(time.time())
# Converts a number of seconds to a date object
print(datetime.datetime.now())
```

Output:

Times is written instead of time.

```
>>> %Run cp2.py
  Traceback (most recent call last):
    File "E:\Semester 1\P fund\LAb 11\cp2.py", line 3, in <module>
      import times
  ModuleNotFoundError: No module named 'times'
```

3. Code:

```
From math import math
# using square root(sqrt) function contained
```

```
print(Math.sqrt(25) )
print(Math.pi)
# 2 radians = 114.59 degreees
print(Math.degrees(2))
```

Output:

```
>>> %Run cp2.py
  Traceback (most recent call last):
    File "E:\Semester 1\P fund\LAb 11\cp2.py", line 1
      From math import math
               ^
  SyntaxError: invalid syntax
```

Math is causing error in line 1

B. What would be the output of the following programs:

1. Code:

```
import calendar
yy = 2017
mm = 11
# display the calendar
print(calendar.month(yy, mm))
```

Output:

```
>>> %Run 'cp 1.py'

     November 2017
  Mo Tu We Th Fr Sa Su
           1  2  3  4  5
   6  7  8  9 10 11 12
  13 14 15 16 17 18 19
  20 21 22 23 24 25 26
  27 28 29 30
```

2. Code:

```
import sys
print(sys.argv)
for i in range(len(sys.argv)):
   if i==0:
       print("The function is",sys.argv[0])
   else:   print("Argument:",sys.argv[i])
```

Output:

```
>>> %Run cp2.py

  ['cp2.py']
  The function is cp2.py
```

3. Code:

```
import numpy as np
# Creating array object
arr = np.array( [[ 1, 2, 3],
                 [ 4, 2, 5]] )

# Printing array dimensions (axes)
print("No. of dimensions: ", arr.ndim)

# Printing shape of array
print("Shape of array: ", arr.shape)

# Printing size (total number of elements) of array
print("Size of array: ", arr.size)
```

Output:

```
>>> %Run cp2.py

  No. of dimensions:  2
  Shape of array:  (2, 3)
  Size of array:  6
```

C. Write Python programs for the following:

1. Write a NumPy program to create an 1D array of 10 zeros, 10 ones, 10 fives

CODE:

```
import numpy as npy
Ones=npy.ones(10)
Zeros=npy.zeros(10)
Fives=npy.ones(10)*5
print("10 Ones Of 1 Dimension Arrays \n ",Ones,'\n')
print("10 Zeros Of 1 Dimension Arrays \n",Zeros,'\n')
print("10 Fives Of 1 Dimension Arrays \n",Fives,'\n')
```

OUTPUT:

```
>>> %Run task1.py

  10 Ones Of 1 Dimension Arrays
    [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]

  10 Zeros Of 1 Dimension Arrays
   [0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]

  10 Fives Of 1 Dimension Arrays
   [5. 5. 5. 5. 5. 5. 5. 5. 5. 5.]
```

2. Write a NumPy program to create a 3x3 matrix with values ranging from 2 to 10.

CODE:

```
import numpy as npy
Mat=npy.arange(1,10).reshape(3,3)
print("A Matrix of 3x3 Matrix")
print(Mat)
```

OUTPUT:

```
>>> %Run 'task 2.py'

  A Matrix of 3x3 Matrix
  [[1 2 3]
   [4 5 6]
   [7 8 9]]
```