Question Bank of Programming In Python for CAT II

1. Define Python Pandas?
2. Mention different types of Data Structures in Panda?
3. Explain different ways of creating Data Frames in Panda?
4. Build a Numpy array filled with all zeros.
5. Reverse a Numpy array.
6. Find the number of occurrences of a sequence in a NumPy array.
7. Analyse the simple working of an algorithm in Tensor Flow?
8. Describe steps involved in making plots. Explain plotting two or more lines on the same plot with an example.
9. Develop a Python program to plot two or more lines with legends, different widths and colours.
10. Describe anatomy of a plot. Explain steps involved in making plots.
11. Characterize the Data Frames in Pandas?
12. Explain the following:
    a. Matplotlib
    b. Seaborn
    c. Plotly
    d. ggplot
13. Explain the Applications of SciPy, Scrapy, Scikit-learn, PyGame, PyTorch, PyBrain and Keras.
14. List the advantages NumPy Arrays have over (nested) Python lists?
15. Briefly explain the use of finalise method in python.
16. Explain the use of init function in python.
17. Explain Python's static methods.
18. Explain Python's Nested Class.
19. List the advantages of using OOPs.
20. Explain access specifiers in python.
21. Can a parent class be called without first creating an instance of it? Explain.
22. How can you determine whether a class is a subclass of another class?
23. How do you make a Python class that is empty?
24. Explain a class object or instance in python.
25. Can you call the base class method without creating an instance? Explain.
26. What is the difference between a class and a structure?
27. Write a Python program to import a built-in array module and display the namespace of the said module.
28. Write a Python class named Circle constructed from a radius and two methods that will compute the area and the perimeter of a circle.
29. Are class and structure the same? If not, what's the difference between a class and a structure?
30. Create a Temperature class. Make two methods:

a) convert Fahrenheit - It will take Celsius and will print it into Fahrenheit.

b) convert Celsius - It will take Fahrenheit and will convert it into Celsius**.**

31. Write a Python program that imports the abs() function using the built-ins module, displays the documentation of the abs() function and finds the absolute value of -155.
32. Write a Python class to implement pow(x, n).
33. Write a Python class to reverse a string word by word.
34. Write a Python class that has two methods: get_String and print_String , get_String accept a string from the user and print_String prints the string in upper case.
35. Illustrate the difference between:
    a) read( ) and readlines ( )
    b) write( ) and writelines( )
    c) r+ file mode and rb+ mode.
    d) w’ and ‘a’ modes
36. Apply the below instructions when writing the program.
    (a) Import Module_Imp2 as mi
    (b) Take two integers a and b as inputs from the user
    (c) Call the function mi.arithoperation by passing a, b.
37. Demonstrate the use of Class while writing a code following the given instructions to add the details of the two students by taking the inputs from the user.
    a) Create a class Student.
    b) Create an instance Stud_1 of class Student.
    c) Create another instance Stud_2 of class Student.
    d) Take name, age, and degree of the student as inputs from the user.
    e) Print the details of the student.
38. f= open(“data.txt”)

Make use of the code given above and write the answers of the following:

a) Identify name of the file.
b) What is ‘f’ in above code?
c) What is the mode of operation in the above file?
d) Discuss the different types of close ().

39. Build a class Employee, which contains the details of an employee like name and salary. Take name and salary as inputs from the console, print the result.
40. Assume a filename and write a Python program to **copy** one file to another file in file handling. Explain the tell() and seek() in python file handling.
41. Define a function checkNegativeNumber which has an argument num. Write a program to check the given num is a **positive** or **negative**. Take the input number from the user, and print the result as shown in the examples.

Sample Input and Output 1:

```
a: 20
positive
```

Sample Input and Output 2:

```
a: -90
negative
```

42. The below program **Module_Imp2** is already written.

- This file is same as **Module1** written earlier, which takes two parameters and does arithmetic operations on these two, and print the result.
- Import it in your **Module4.py**.
- Take two integers a and b as input from the user.
- Call the function **arithoperations()** of **Module_Imp2** module by passing the arguments a and b.

Sample Input and Output:

```
a: 12
b: 10
addition: 22
subtraction: 2
multiplication: 120
division: 1.2
```

43. Apply the below instructions when writing the program.

(a) Import Module_Imp2 as mi

(b) Take two integers a and b as inputs from the user.

(c) Call the function mi.arithoperation by passing a, b.

44. Inspect whether class and structure are the same? If not, what's the difference between a class and a structure?

45. Inspect whether a parent class be called without first creating an instance of it? Explain.

46. Build a two 2-D array. Plot it using matplotlib.

47.Analyse the steps to create a 1D array and 2D array.

48.Write a program that uses * in import.

- from Module_Imp3 import *
- Take integer as input from user and store it in the variable **side.**
- Call the function **calculatearea**(side, side)
- Call the function pivalue()
- print the third element in shapes.
- Note: The Module_Imp3.py already is written.

49. Follow the given instructions while writing the program

- Use the **Module_Imp3** which contains functions that can be imported.
- Use from Module_Imp3 import *

- Take an integer as **input** from user and store it in the variable **side.**
- Call the function calculatearea(side,side)
- Call the function calculatediameter(side)
- Call the function pivalue()
- print shapes[1:2]

50. Write a simple program followed by the instructions given below:

- A base class **Person** and a derived class **Student** with **Person** as its base class.
- Add two methods **setname()** (which takes the parameter self and name)and **getname()** which prints the name in the base class.
- Add two methods in the derived class: **setage()** (which takes the parameters self and age) which sets the age and **getage()** which prints the age.
- Create an instance of **Student** and name it as **s1**.
- Take **name** and **age** as inputs from the console.
- Call the **setname()** and **setage()** on this instance by passing the **name** and **age** parameters.
- Call the **getname()** and **getage()** on this class, which prints the passed parameters