# Assignment-1

Q1. Why do we call Python as general purpose and high-level programming language?

Ans. Python is a popular general-purpose programming language that has a wide variety of applications. All sorts of technological solutions have Python at their core, from web applications, search engines, and games to animation software and even other programming languages. It's an object-oriented, high-level programming language. Object-oriented means this language is based around objects (such as data) rather than functions, and high-level means it's easy for humans to understand as compared to other programming languages which use more complex syntax and coding for defining and declaring the variables.

Q2. Why is Python called a dynamically typed language?

Ans. Since we know that Dynamic typing means, the variables are defined only at runtime and Python is a dynamically typed language. It doesn’t know about the type of the variable until the code is run. So declaration is of no use. What it does is, It stores that value at some memory location and then bind that variable name to that memory container. And makes the contents of the container accessible through that variable name. So the data type does not matter. As it will get to know the type of value at run-time only. Due to dynamic typing, in Python, the same variable can have a different type at different times during the execution. Dynamic typing allows for flexibility in programming, but with a price in performance.

Q3. List some pros and cons of Python programming language.

Ans. The Pros and Cons of the Python Programming language are -

\* Pros

1. It's a High-level programming language which makes it more user-friendly and easy to learn as compared to other programming languages.

2. Being a "Dynamically Typed Programming language", Python increases your productivity and reduces the extent of stress you carry while working on any project as you don't have to worry about the declaration of the variables while creating a function.

3. It has a vast collection of libraries, so you don't have to depend on external sources for creating and developing your software. And in case you don't find any library in it then you can use the Python package manager (pip) and the Python package index (PyPi) — these two resources work hand in hand.

In the Python package index, there are over 200,000 packages for you to work with, and you can import these packages with the use of a Python package manager. Indeed, there are no limits to what you can achieve with Python!

4. It's a Portable language, As it is able to work on different platforms and you don’t need to edit the program for your operation to be successful.

5. Python is interpreted language, the program is executed from the source code directly. So all you need to do is to ensure that the right libraries are linked to the program. Python's mode of operation is special. It converts the source code into bytecodes and these bytecodes are then translated into the native language before the program is executed. You don't have to compile programs here, which makes things simpler and faster for you.

\* Cons

1. It has speed limitations. The program is not fast when executing codes, and this has a lot to do with the fact that Python is a dynamically typed and interpreted programming language.

2. It's not so strong with mobile computing. It was built to be used in the server-side programming, so the client-side is rarely used — and that’s if it is ever used at all. Because of this, Python does not do well with the making of mobile applications.

3. Python can have runtime errors; While using Python, you can expect to see runtime errors because of the dynamical typing feature of this programming language. Since the data of a variable is not static, it can change at any time, so runtime errors are really hard to avoid.

4. It consumes a lot of memory space, IT naturally uses a huge amount of memory to carry out all the features that made you choose it, so if you are building an application that needs memory optimization, using Python will restrict your memory space.

5. Not easy to test. When your program is being executed, all the errors are caught in the process. And for you to successfully launch the output, you must clear out or modify every single error, so testing takes a lot of time and work.

Q4. In what all domains can we use Python?

Ans. We can use Python in different domains like-

\* Machine learning / Artificial intelligence

\* Desktop GUI

\* Data analytics and data visualization

\* Web development

\* Game development

\* Mobile app development

\* Embedded systems

Q5. What are variables and how can we declare them?

Ans. A variable is created when some value is assigned to it. The value assigned to a variable determines the data type of that variable But Python has no command for declaring a variable i.e we need not declare a variable with some specific data type here. Thus, declaring a variable in Python is very simple. For Example-

-> Just name the variable -> Assign the required value to it and the data type of the variable will be automatically determined from the value assigned, we need not define it explicitly.

Q6. How can we take input from the user in Python?

Ans. input (): This function first takes the input from the user and converts it into a string. The type of the returned object always will be <type ‘str’>. It does not evaluate the expression it just returns the complete statement as a String. For example, Python provides a built-in function called input which takes input from the user. When the input function is called it stops the program and waits for the user’s input. When the user presses enter, the program resumes and returns what the user typed.

Syntax: inp = input('STATEMENT')

Q7. What is the default datatype of the value that has been taken as an input using input() function?

Ans. By default, input returns a string.

Q8. What is typecasting?

Ans. Type Casting is the method to convert the variable data type into a certain data type in order to the operation required to be performed by the users.

Q9. Can we take more than one input from the user using single input() function? If yes, how? If no, why?

Ans.  Python provides the two methods that help us to take multiple values or input in one line.

      1. Using split() method

      2. Using List Comprehension

Q10. What are keywords?

Ans. Python keywords are special reserved words that have specific meanings and purposes and can't be used for anything but those specific purposes. These keywords are always available—you'll never have to import them into your code. Also, they cannot be used as variable names, function names, or any other identifiers.

Q11. Can we use keywords as a variable? Support your answer with reason.

Ans. No, we can't use keywords as a variable.

For example, We can't name our variable as follows, as print is a keyword in Python.

 //print (we can't use it as a variable to define or declare anything in our program); But, we can name our variable by changing the case of any letter in the keyword print.

Q12. What is indentation? What's the use of indentation in Python?

Ans. Indentation refers to the spaces at the beginning of a code line. Whereas in other programming languages, the indentation in code is for readability only, the indentation in Python is very important. Python uses indentation to indicate a block of code. And it will give you an error if you skip the indentation.

Q13. How can we throw some output in Python?

Ans. We can throw the output in the Python just by simply using the “print” Keyword with the sentence we want to print.

Q14. What are operators in Python?

Ans. Operators are used for performing operations on variables and values.

The Different types of operators in the Python language are-

1. Assignment operators

2. Comparison operators

3. Logical operators

4. Identity operators

5. Membership operators

6. Bitwise operators

7. Arithmetic Operators

Q15. What is the difference between / and // operators?

Ans. // is used to get the integer value after performing division while / will give you the value of the division in float form.

Q16. Write a code that gives the following as an output.

```iNeuroniNeuroniNeuroniNeuron’’’

Ans. print(‘iNeuroniNeuroniNeuroniNeuron’)

Q17. Write a code to take a number as input from the user and check if the number is odd or even.

Ans.

num = int(input("Enter the number"))

if num%2 == 0 :

    print("Number is even", num)

else:

    print("Number is odd", num)

Q18. What are boolean operators?

Ans. Boolean Operators are those that result in the Boolean values of True and False. These include and, or and not. While and & or require 2 operands, not is a unary operator. Boolean operators are most commonly used in arithmetic computations and logical comparisons.

Q19. What will the output of the following?

```

1 or 0

0 and 0

True and False and True

1 or 0 or 0’’’

Ans. 1 or 0 Gives 1, True.

0 and 0 also give 1, True.

True and false give False, Then false and true again gives false value.

1 or 0 = 1, 1 or 0 = 1. Thus it will return a true value, i.e 1.

Q20. What are conditional statements in Python?

Ans. Conditional Statements are basically decision-making statements. In Python there are mainly 3 conditional statements, also called as If-Elif-Else ladder. This is used when we try to write code for something whose value is determined by the user input and that input is then moved to the different blocks of if-elif-else statements, where each one has specific conditions so if condition one satisfies it enters the if block, and if not then moves towards elif block and if this also not satisfies then moves into else block and print that statement.

Q21. What is the use of 'if', 'elif', and 'else' keywords

Ans. The if/elif/else structure is a common way to control the flow of a program, allowing us to execute specific blocks of code depending on the value of some data.

IF Statement:

If the condition following the keyword is evaluated as true, the block of code will execute. Also, Parentheses are not used before and after the condition.

Else statement:

You can optionally add an else response that will execute if the condition is false.

Elif statement

Multiple conditions can be checked by including one or more elif checks after the initial if statement but only one condition will execute in the end.

Q22. Write a code to take the age of the person as input and if age >= 18 display "I can vote". If age is < 18 display "I can't vote".

Ans. age=int(input(‘Enter the age of a person))

if age >= 18:

print(“ I can Vote”, age)

else:

print(“I can’t Vote”,age)

Q23. Write a code that displays the sum of all the even numbers from the given list.

```

numbers = [12, 75, 150, 180, 145, 525, 50]

```

Code:

numbers = [12, 75, 150, 180, 145, 525, 50]

even = []

for items in numbers:

    if items%2 == 0:

        even.append(items)

    else:

        pass

print("Sum of even numbers are", sum(even))

Q24. Write a code to take 3 numbers as an input from the user and display the greatest no as output.

Code:

a = int(input())

b = int(input())

c = int(input())

if a>b:

    if c>a:

        print("The greatest number is", c)

    else:

        print("The greatest number is", a)

else:

    print("The greatest number is ", b)

Q25. Write a program to display only those numbers from a list that satisfy the following conditions

- The number must be divisible by five

- If the number is greater than 150, then skip it and move to the next number

- If the number is greater than 500, then stop the loop

```

numbers = [12, 75, 150, 180, 145, 525, 50]

Code:

numbers = [12, 75, 150, 180, 145, 525, 50]

for items in numbers:

    if items%5 == 0:

        if items<=150:

            print(items)

        else:

            pass

        if items>500:

            break