**Assignment Part-1**

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

**Python is called as general-purpose language because it is designed to be used for building software in a wide variety of application domains, across a multitude of hardware configurations and operating systems**.

**Python is called as high-level programming language, because they are not written in machine-readable language, Python programs need to be processed before machines can run them**. Python is an interpreted language. This means that every time a program is run, its interpreter runs through the code and translates it into machine-readable byte code.

**Q2. Why is Python called a dynamically typed language?**

Python is a dynamically typed language. **There is no need to declare the type of a variable or manage the memory while assigning a value to a variable in Python**. Python don't have any problem even if we don't declare the type of variable. It states the kind of variable in the runtime of the program. Python also take cares of the memory management which is crucial in programming. So, Python is a dynamically typed language.

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

**Pros**

1. Python programming language focuses on code readability. It’s versatile, neat, easy to use and learn, readable, and well-structured.

2. You can find a library for basically anything you can think of: from web and mobile development, through game development, to machine learning - and if there isn’t one available already, you can easily create your own.

**3.** As it was mentioned before, Python is easy to learn and fast to develop with and offers asynchronous coding. You can do more with less code, which means you can build prototypes, such as graphical user interfaces, and test out ideas much quicker in Python than in other languages.

**Cons**

**1. Seed Limitation:** Python is an interpreted language, so you may find it slower than other popular languages.

**2. Problem with threading:** Threading is not really good in Python due to the Global Interpreter Lock (GIL). GIL is simply a mutex that allows only one thread to execute at a time. As a result, executing code through multi-threaded CPU-bound programs may be slower than single-threaded ones.

**3.** **Memory Consumption:** You should take into consideration that Python’s memory consumption is very high. For that reason, it might not be the best choice for memory intensive tasks

**Q4. In what all domains can we use Python?**

we can program all kinds of applications using Python. The general-purpose language can be used to read and create file directories, create GUIs and APIs, and more. Whether it's blockchain applications, audio and video apps, or machine learning applications, you can build them all with Python.

**Q5. What are variable and how can we declare them?**

A Python variable is a reserved memory location to store values. A variable is created the moment you first assign a value to it.

Example: 1

x = 5  
y = "John"  
print(x)  
print(y)

Variables do not need to be declared with any particular type, and can even change type after they have been set.

x = 4       # x is of type int  
x = "Sally" # x is now of type str  
print(x)

**Casting :**

If you want to specify the data type of a variable, this can be done with casting.

x = str(3)    # x will be '3'  
y = int(3)    # y will be 3  
z = float(3)  # z will be 3.0

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

**Python input() function** is used to take user input. By default, it returns the user input in form of a string.

**Python input() Function Syntax**

***Syntax:****input(prompt)*

* ***prompt [optional]:****any string value to display as input message*

***Returns:****Return a string value as input by the user.*

Example : # Taking name of the user as input

# and storing it name variable

name **=** input("Please Enter Your Name: ")

# taking age of the user as input and

# storing in into variable age

age **=** input("Please Enter Your Age: ")

print("Name & Age: ", name, age )

**Output:**

Please Enter Your Name: Rohit

Please Enter Your Age: 16

Name & Age: Rohit 16

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

By default, it returns a string object. Hence, the input() function by default returns the value as string data type.

**Q8. What is type casting?**

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 users.

There can be two types of Type Casting in Python –

Implicit Type Casting: In this, methods, Python converts data type into another data type automatically.

Explicit Type Casting: In this method, Python need user involvement to convert the variable data type into certain data type in order to the operation required.

Mainly in type casting can be done with these data type function:

**Int() :**Int() function take float or string as an argument and return int type object.

**float() :**float() function take int or string as an argument and return float type object.

**str() :**str() function take float or int as an argument and return string type object.

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

**Using**[**split()**](https://www.geeksforgeeks.org/python-string-split/)**method**

   
This function helps in getting multiple inputs from users. It breaks the given input by the specified separator. If a separator is not provided then any white space is a separator. Generally, users use a split() method to split a Python string but one can use it in taking multiple inputs.

**Syntax :**

input().split(separator, maxsplit)

x, y **=** input("Enter two values: ").split()

print("Number of boys: ", x)

print("Number of girls: ", y)

**print**()

**Q10. What are keywords?**

Python has a set of keywords that are reserved words that cannot be used as variable names, function names, or any other identifiers: All the keywords in python are written in lower case except True and False. There are 33 keywords in Python.

['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break', 'cl ass', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'fr om', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']

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

We can’t use keywords as a variable. If we attempt to use any of these reserved keywords as variable names in the application, we will see a **compile time error**

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

Indentation refers to the spaces at the beginning of a code line.

Where 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.

if 5 > 2:  
  print("Five is greater than two!")

**Q13. How can we throw some output in Python?**

The print() function **prints the specified message to the screen, or other standard output device**. The message can be a string, or any other object, the object will be converted into a string before written to the screen.

**Q14. What are operators in Python?**

Operators are used to perform operations on variables and values.

print(10 + 5)

Python divides the operators in the following groups:

* Arithmetic operators: +-\*/%\*\*//
* Assignment operators =+=-=\*=/=\*\*==&=/=^=>>=<<==
* Comparison operators ===!<>>=<=
* Logical operators: and or not
* Identity operators: is or is not
* Membership operators: in or not in
* Bitwise operators : &|^~<<>>

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

'/' is the division operator.

'//' is the floor division operator.

Python supports different types of operators:

They are arithmetic operators, logical operators, assignment operators, etc.

'/' and '//' belong to the arithmetic operators.

'/' is used for the normal division of two numbers.

'//' is used to obtain the smallest integer nearest to the quotient obtained by dividing two numbers.

Let us see an example to understand this.

x = 15

y = 3

print(x / y)   #This prints output as 5

print(x // y)  #This prints output as 5

a = 19

b = 4

print(a // b)  #This prints output as 4

print(a / b)  #This prints output as 4.75

So, if the quotient obtained by dividing two numbers is not an integer, then operators '/' and '//' will return different answers.

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

Example : 1 print(“ iNeuroniNeuroniNeuroniNeuron ”)

Example 2: x = (“ iNeuroniNeuroniNeuroniNeuron”)

print(x)

Example 3: str = ‘iNeuron’)

Print(str\*3)

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

# Python program to check if the input number is odd or even.

# A number is even if division by 2 gives a remainder of 0.

# If the remainder is 1, it is an odd number.

num = int(input("Enter a number: "))

if (num % 2) == 0:

print("{0} is Even".format(num))

else:

print("{0} is Odd".format(num))

**Q18. What are boolean operator?**

In programming you often need to know if an expression is True or False.

You can evaluate any expression in Python, and get one of two answers, True or False.

When you compare two values, the expression is evaluated and Python returns the Boolean answer:

print(10 > 9)  
print(10 == 9)  
print(10 < 9)

Print a message based on whether the condition is True or False:

a = 200  
b = 33  
  
if b > a:  
  print("b is greater than a")  
else:  
  print("b is not greater than a")

**Q19. What will the output of the following? ``` 1 or 0**

0 and 0 True and False and True 1 or 0 or 0 ```.

Output for 1 or 0 is 1

Output for 0 or 0 is 0

Output for True and False and True is False

Output for 1 or 0 or 0 is 1

**Q20. What are conditional statements in Python?**

A conditional statement as the name suggests itself, is used to handle conditions in your program. These statements guide the program while making decisions based on the conditions encountered by the program.

Python has 3 key Conditional Statements:

* *if*statement
* *if-else* statement
* *if-elif-else* ladder

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

An "if statement" is written by using the if keyword.

a = 33  
b = 200  
if b > a:  
  print("b is greater than a")

In this example we use two variables, a and b, which are used as part of the if statement to test whether b is greater than a. As a is 33, and b is 200, we know that 200 is greater than 33, and so we print to screen that "b is greater than a".

## Elif

The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition".

a = 33  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")

In this example a is equal to b, so the first condition is not true, but the elif condition is true, so we print to screen that "a and b are equal".

## else

The else keyword catches anything which isn't caught by the preceding conditions.

a = 200  
b = 33  
if b > a:  
  print("b is greater than a")  
elif a == b:  
  print("a and b are equal")  
else:  
  print("a is greater than b")

In this example a is greater than b, so the first condition is not true, also the elif condition is not true, so we go to the else condition and print to screen that "a is greater than b".

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

# input age

age = int(input("Enter Age : "))

# condition to check voting eligibility

**if** age>=18:

status="I can vote"

**else**:

status="I can’t vote"

**print**(status)

**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]**

**test\_list = [345, 893, 1948, 34, 2346]**

**even\_sum = 0**

**even\_sum += sum([int(ele)for sub in test\_list for ele in str(sub) if int(ele) % 2 == 0])**

**print("Even digit sum : " + str(even\_sum))**

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

# Python program to find the greatest number among the three input numbers

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

num3 = float(input("Enter third number: "))

if (num1 >= num2) and (num1 >= num3):

largest = num1

elif (num2 >= num1) and (num2 >= num3):

largest = num2

else:

largest = num3

print("The largest number is", largest)

**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**

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

# iterate each item of a list for item in numbers:

if item > 500:

break

elif item > 150:

continue

# check if number is divisible by 5

elif item % 5 == 0:

print(item)