**# Assignment-1(Python)**

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

We call Python as a general-purpose programming language because it is designed to be used in a range of applications, including data science, software and web development, automation, and generally getting stuff done. It’s versatility, along with its beginner-friendliness, has made it one of the most-used programming languages today.

Python is called high-level programming language because it is easy for humans to understand and the coding is also quite easy.

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

In Python, there is no need to define the types of variables. It allows to directly use the variables as its type-checking will be done during the execution of the program. The interpreter checks the program line-by-line and also examines the data type of the variable. So, Python is a dynamically typed language.

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

Some of the main benefits of Python include its ease of use, concise and straightforward syntax, and vast libraries. Other Python advantages are its portability, versatility, large user base, and free & open-source license.

Some of the disadvantages of Python include its slow speed and heavy memory usage. It also lacks support for mobile environments, database access, and multi-threading.

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

Data Science, Machine Learning, Deep Learning, Artificial Intelligence, Scientific Computing Scripting, Networking, Game Development to Web Development.

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

A variable is a named unit of data that is assigned a value. If the value is modified, the name does not change. Python has no command for declaring a variable. A variable is created when some value is assigned to it. The value assigned to a variable determines the data type of that variable. Thus, declaring a variable in Python is very simple.

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

input (): This function first takes the input from the user and converts it into a string. The type of the returned value always will be <class 'str'>. It does not evaluate the expression it just returns the complete statement as String.

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

The default datatype of the value that has been taken as an input using input() function, always will be <class 'str'>.

**Q8. What is type casting?**

Type Casting is the method to convert the Python variable datatype into a certain data type in order to perform the required operation by users.

Example:

b **=** 3.0

print(type(b))

Here, Output will be <class ‘float’>

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

Yes, we can take more than one input from the user with the help of split() function. 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)

**Q10. What are keywords?**

Python Keywords are some predefined and reserved words in python that have special meanings. Keywords are used to define the syntax of the coding. The keyword cannot be used as an identifier, function, or variable name. All the keywords in python are written in lowercase except True and False.

Some of the examples of Keywords are: and, or, not, if, if-else, while, break, continue, etc.

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

We cannot use a keyword as a variable name, function name, or any other identifier. They are used to define the syntax and structure of the Python language. If we try to use keywords as a variable name it will show “SyntaxError: invalid syntax” in the output.

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

Python indentation refers to adding white space before a statement to a particular block of code. In another word, all the statements with the same space to the left, belong to the same code block. Indentation is a very important concept of Python because without properly indenting the Python code, you will end up seeing “IndentationError” and the code will not get compiled.

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

With the help of “print()” function, we can throw some output to the screen or any other standard output device.

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

Operators in general are used to perform operations on values and variables. These are standard symbols used for the purpose of logical and arithmetic operations.

Example: +, -, \*, /, <, >, >=, <=, !=, etc.

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

In Python, “/” is a float-division, and “//” is an Integer-division.

when we apply float-division on two numbers, output will be in float. And, when we apply integer-division on two numbers, output will be in integer form. Let’s see an example to understand it better.

a = 5

b = 3

print(“Float-Division on a and b is = ”, a/b)

print(“Integer-Division on a and b is = ”, a//b)

Output:

Float-Division on a and b is = 1.666666666

Integer-Division on a and b is = 1

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

**```**

**iNeuroniNeuroniNeuroniNeuron**

**```**

str1 = "iNeuron" \* 4

print(str1)

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

a = int(input("Please enter your number"))

if a%2 == 0:

    print(a, " is a Even number.")

else:

    print(a, "is a Odd number.")

**Q18. What are Boolean operators?**

Boolean Operators are those that result in the Boolean values of True and False. These include ‘and’, ‘or’ and ‘not’. While ‘and’ 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** = 1

**0 and 0** = 0

**True and False and True** = 0

**1 or 0 or 0** = 1

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

Conditional statements allow you to control the flow of your program based on conditions that you specify. They provide a way to make decisions in your program and execute different code based on those decisions. The following are the conditional statements provided by Python.

* if
* if..else
* Nested if
* if-elif statements.

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

'if', 'elif' and 'else'are conditional statements that provide you with the decision making that is required when you want to execute code based on a particular condition. These statements used in Python helps to automate that decision making process.

**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".**

age = int( input("Please enter your age"))

if age>=18:

    print("I can vote")

else:

    print("I can't vote")

**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]**

**```**

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

sum = 0

for num in numbers:

    if num%2==0:

        sum = sum + num

    else:

        continue

print("sum of even numbers in the list is =", sum)

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

num1 = int(input("Please enter first number"))

num2 = int(input("Please enter second number"))

num3 = int(input("Please enter third number"))

if num1>num2:

    if num1>num3:

        print(num1, "is the greatest")

    else:

        print(num3, "is the greatest")

elif num2>num3:

    print(num2, "is the greatest")

else:

    print(num3, "is the greatest")

**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]**

**```**

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

final\_list = []

for num in numbers:

    if num>500:

        break

    elif num>150:

        continue

    elif num%5==0:

        final\_list.append(num)

print(final\_list)