

Assignment Part-1

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

Ans:

Python is a general purpose and high level programming language because it is designed to be easy to read and simple to implement and it is easy to understand and the coding is also quite easy.

Q2. Why is Python called a dynamically typed language?

Ans:

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. It stores that value at some memory location and then binds 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 value at run-time.

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

Ans:

Python is a powerful and versatile programming language that is easy to learn and use. It is a good choice for a wide variety of tasks, but it is important to be aware of its limitations. So we discuss pros and cons,

Pros:

- Easy to learn and use: Python has a simple and straightforward syntax that makes it easy to learn for beginners. The language is also very readable, which makes it easy to maintain and debug code.
- Powerful and versatile: Python is a powerful language that can be used for a wide variety of tasks. It is often used for web development, data science, machine learning, and artificial intelligence.
- Open source and free: Python is an open source language, which means that it is free to use and distribute. There is also a large and active community of Python developers who contribute to the language and its libraries.
- Large library ecosystem: Python has a large and well-maintained library ecosystem. This means that there are many libraries available for a wide variety of tasks. This can save developers a lot of time and effort.

Cons:

- Slow performance: Python is not as fast as some other programming languages, such as C or Java. This is because Python is an interpreted language, which means that it is not compiled to machine code before it is executed.
- Not as memory efficient as some other languages: Python can be memory-intensive, especially for large applications. This is because Python uses dynamic typing, which means that the type of a variable is not checked until it is used.

- Lack of some features found in other languages: Python does not have some features that are found in other programming languages, such as pointers and multithreading. This can limit the use of Python for some applications.

Q4. In what all domains can we use Python?

Ans:

Python is a general-purpose programming language, which means that it can be used for a wide variety of tasks.

- Web Development
- Data Science
- Machine Learning
- Artificial intelligence
- Scientific computing
- Game development
- Systems programming

These are just a few of the many domains where Python is used. As a general-purpose programming language, Python can be used for a wide variety of tasks

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

Ans:

In Python, a variable is a named location in memory that stores a value. Variables are used to store data so that it can be used later in the program.

To declare a variable in Python, you use the `var_name = value` syntax. For example, the following code declares a variable named `name` and assigns it the value "Python":

```
name = "Python"
```

Variable names in Python can be any combination of letters, numbers, and underscores. The first character of a variable name must be a letter or an underscore. Variable names are case-sensitive, so `name` and `NAME` are two different variables.

Once we declared a variable we can use it to store the data

```
Print(name)
```

#Output: Python

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

Ans:

In Python, we can take an input from the user using the `input()` function. The `input()` function takes a string as input and returns a string. For example, the following code asks the user to enter their name and stores the input in the variable `name`:

```
name = input("What is your name? ")
```

#Output:

What is your name? Python

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

Ans:

The default datatype of the value that has been taken as an input using the input() function in Python is string. This means that even if we enter an integer or float value, the input() function will convert it into a string. If we want to store the input value as an integer or float, we will need to perform type conversion.

For example:

```
name = input("Enter your name: ")
print(name)
```

If we enter the name "Shashank Kumar" as input, the input() function will convert it into the string "Shashank Kumar" and print it out.

To store the input value as an integer, we can use the following code:

```
#####
age = int(input("Enter your age: "))
print(age)
```

If we enter the age "30" as input, the input() function will convert it into the integer 30 and store it in the variable age.

To store the input value as a float, we can use the following code:

```
height = float(input("Enter your height: "))
print(height)
```

If we enter the height "1.80" as input, the input() function will convert it into the float 1.80 and store it in the variable height.

Q8. What is type casting?

Ans:

Type casting is the process of converting a data type into another data type.

Q9. Can we take more than one input from the user using a single input() function? If yes, how? If not, why?

Ans:

The input() function in Python can only take one input at a time. If we want to take multiple inputs from the user, we can use the split() method to split the input string into multiple values.

For example, the following code will take two inputs from the user and store them in the variables name and age:

```
name, age = input("Enter your name and age: ").split()
print("Your name is", name, "and your age is", age)
```

If the user enters the input "Shashank 20", the split() method will split the string into two values, "Shashank" and "20". The variables name and age will then be assigned the values "Shashank" and "20", respectively.

We can also use the list comprehension syntax to take multiple inputs from the user.

```
inputs = [int(x) for x in input("Enter three numbers: ").split()]
print("The numbers are", inputs)
```

Q10. What are keywords?

Ans:

Python keywords are unique words reserved with defined meanings and functions that we can only apply for those functions. You'll never need to import any keyword into your program because they're permanently present.

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

Ans:

No, We cannot use keywords as a variable in Python.

Keywords are reserved words in Python that have special meanings and purposes. They cannot be used as variable names, function names, or any other identifier.

If we try to use a keyword as a variable name, we will get an error.

For example:

```
keyword = "def"
print(keyword)
```

This code will raise an error because the keyword def is reserved for use as a function definition.

- It would be confusing to the interpreter. The interpreter would not know if we were trying to use a keyword as a keyword or as a variable.
- It would be confusing to other programmers. If we were to share our code with other programmers, they might not be aware that we were using a keyword as a variable.

- It would be a security risk. If we were to use a keyword as a variable, someone could exploit that vulnerability to gain unauthorised access to our code.

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

Ans:

Indentation in Python is the use of whitespace to indicate the structure of a program. It is used to group statements together into blocks, and to indicate the nesting of blocks.

Indentation is not optional in Python. It is a critical part of the language's syntax, and it is used by the Python interpreter to determine the structure of a program. If a program is not indented correctly, the Python interpreter will not be able to understand it, and it will not be able to run.

The amount of indentation used in Python is up to the programmer. However, it is generally recommended to use four spaces for each level of indentation. This makes the code more readable and easier to understand.

Q13. How can we throw some output in Python?

Ans:

There are two ways to throw some output in Python:

Using the print() function. The print() function is used to print text to the console. For example, the following code will print the string "Hello, world!"

```
print("Hello, world!")
```

Using the raise keyword. The raise keyword is used to raise an exception.

An exception is an event that occurs during the execution of a program that disrupts the normal flow of the program. When an exception is raised, the program stops executing the current code and jumps to the exception handler. The exception handler is a block of code that is executed to handle the exception.

```
x = 5
if x != 10:
    raise Exception("The value of x is not equal to 10")
```

Q14. What are operators in Python?

Ans:

Operators in Python are special symbols that are used to perform operations on variables and values.

There are many different types of operators in Python, including arithmetic operators, comparison operators, logical operators, assignment operators, and membership operators.

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

Ans:

The / and // operators in Python are both used to divide two numbers. However, there is a subtle difference between the two operators.

The / operator performs normal division, which means that the result of the division is a floating-point number. For example code will print the floating-point number 1.5:

```
x = 3
y = 2

print(x / y)
```

The // operator performs floor division, which means that the result of the division is an integer. The integer is the quotient of the division, and any decimal part of the quotient is discarded.

For example

```
x = 3
y = 2
print(x // y)
```

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

```
...
iNeuroniNeuroniNeuroniNeuron
...
```

Ans:

```
def print_neuron_five_times():
    for i in range(5):
        print("iNeuron")

print_neuron_five_times()
```

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

Ans:

```
Num = int(input(" ENter a Number :"))
If num % 2==0:
    print("{0} is Even ".format(num))
else:
    print("{0} is odd".format(num))
```

```
=====
Output:                Enter a number: 43
                        43 is Odd
                        Enter a number: 18
                        18 is Even
```

Q18. What are boolean operators?

Ans:

Python has three Boolean operators, or logical operators: and , or , and not
Boolean is type of value that can be either True or False

Q19. What will be 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 # True
0 and 0 # False
True and False and True # False
1 or 0 or 0 # 1

Q20. What are conditional statements in Python?

Ans:

(if, else, and elif)

Conditional statements are fundamental programming constructs that allow we to control the flow of our program based on conditions that we specify

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 we to execute specific blocks of code depending on the value of some data

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

Ans:

```
age = int(input("Enter you 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]  
...
```

Ans:

```
numbers = [12, 75, 150, 180, 145, 525, 50]  
add = 0  
for num in numbers:  
    if num%2 == 0:  
        add = add+num  
    else:  
        continue  
print(add)
```

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

Ans:

```
x, y, z = input("Enter 3 numbers separated by comma: ").split(",")  
if int(x) > int(y) and int(x) > int(z):  
    print(f"{x} is greatest")  
elif int(y) > int(z):  
    print(f"{y} is greatest")  
else:  
    print(f"{z} is 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]  
...
```

Ans:

```
numbers = [12, 75, 150, 180, 145, 525, 50]  
lst = []  
for num in numbers:  
    if num > 150:  
        if num > 500:  
            break  
        elif num%5==0:  
            lst.append(num)  
print(lst)
```


