29th JAN ASSIGNMENT

Q1. Who developed Python programming language?

Python programming Language was developed by Guido Van Rossum during 1985-1990 at CWI in Netherland.The idea of Python programming language has taken from the ABC programming language or we can say that ABC is a predecessor of Python language.

There is also a fact behind the choosing name Python. Guido van Rossum was a fan of the popular BBC comedy show of that time, "Monty Python's Flying Circus". So he decided to pick the name Python for his newly created programming language.

Q2.Which type of programming does python support?

Python supports multiple programming pattern, including object-oriented, imperative, and functional or procedural programming styles.Also,Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development.

Q3.Is python case sensitive when dealing with identifiers?

Yes, Python is case-sensitive when dealing with identifiers, which include variable names, function names, class names, and so on. This means that "myVariable" and "myvariable" would be treated as two different identifiers in Python.

For example, consider the following Python code:

```
myVariable = 5
print(MyVariable)
```

This code would produce a NameError, because "MyVariable" is not the same as "myVariable". Python would not recognize "MyVariable" as a previously defined identifier.

Q4.What is the correct extension of the python file?

The correct extension for Python files is ".py". This is a common convention in the Python community and is used to indicate that a file contains Python code.

For example, if you have a Python script that you want to save as a file, you would typically give it a name like "my_script.py".

Q5.Is python code compiled or interpreted?

Python code is interpreted, meaning that each line of code is executed one by one by the interpreter at runtime. The interpreter reads the code line by line and executes each instruction as it encounters it.

However, Python code is also compiled into bytecode, which is a lower-level language that can be executed more efficiently by the interpreter. When you run a Python program, the interpreter first compiles the source code into bytecode and then executes the bytecode. This compilation process happens automatically and is transparent to the user.

So, while Python is an interpreted language, it uses a hybrid approach that includes compilation to bytecode for improved performance.

Q6.Name a few block of code used to define in python language?

In Python, there are several block structures that are defined by their indentation level. Here are a few examples:

1.If statement block:

```
if x > 0:
    print("x is positive")
else:
    print("x is not positive")
```

2. For loop block:

```
for i in range(10):
    print(i)
```

3.Function definition block:

```
def greet(name):
    print("Hello, " + name + "!")
```

4.Class definition block:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def greet(self):
        print("Hello, my name is " + self.name + " and I am " + str(self.age) + " years old.")
```

Q7.State a character used to give single line comments in python?

In Python, the hash symbol (#) is used to indicate a single line comment.Any text on the same line after the hash symbol is considered a comment and is ignored by the Python interpreter.

```
# This is a comment
print("Hello, World!") # This is also a comment
```

In this example, the first line is a comment and is ignored by Python when the code is executed. The second line is a print statement that will output "Hello, World!" to the console when executed. The third line is also a comment and is ignored by Python.

Q8.Mention functions which can help us to find the version of python that we are currently using?

In Python, there are several ways to find out the version of the Python interpreter that is currently installed and running on your system. Here are a few methods:

Using the sys module: The sys module provides access to some variables used or maintained by the interpreter, including the version information.

```
import sys
print(sys.version)
```

Using the platform module: The platform module provides an interface to various system and hardware information.

```
import platform
print(platform.python_version())
```

Using the command line: You can also find out the version of Python installed on your system by running the python command with the --version or -V option from the command line:

```
python --version
```

Q9.Python supports the creation of anonymous functions at runtime,using a constructor called?

In Python, anonymous functions can be created at runtime using a constructor called lambda.

The lambda keyword is used to define small, anonymous functions that can be created and used on the fly. Here's an example:

```
# Define a lambda function that adds two numbers
add = lambda x, y: x + y

# Use the lambda function to add two numbers
result = add(3, 5)

# Print the result
print(result)  # Output: 8
```

In this example, we define an anonymous function using the lambda keyword that takes two arguments x and y and returns their sum. We then use this function to add two numbers and print the result. The lambda function is a quick and convenient way to define small, throwaway functions without having to use the def keyword to define a named function.

Q10.What does pip stand for python?

pip stands for "Pip Installs Packages". It is a package manager for Python that allows you to easily install, update, and manage Python packages and their dependencies.

With pip, you can install packages from the Python Package Index (PyPI), as well as from other sources such as version control systems and local or remote archives. pip makes it easy to manage packages and their dependencies, and ensures that all necessary libraries are installed correctly and in the right order.

Q11.Mention a few built in function in python?

Python has many built-in functions that are available for use without requiring any import statements. Here are some of the most commonly used built-in functions in Python:

print(): Used to output text or variables to the console.

input(): Used to get user input from the console.

len(): Used to get the length of a string, list, or other iterable.

str(), int(), float(), bool(): Used to convert values to different data types.

range(): Used to create a sequence of numbers.

min(), max(): Used to get the minimum or maximum value in a list or iterable.

sum(): Used to get the sum of all values in a list or iterable.

sorted(): Used to sort a list or iterable.

enumerate(): Used to iterate over a list or iterable and keep track of the index.

zip(): Used to combine two or more lists or iterables into a single iterable.

type(): Used to get the data type of a value.

help(): Used to get help and documentation on a function or module.

Q12.What is the maximum possible length of an identifier in python?

In Python, the maximum length of an identifier is not explicitly defined, but it is limited by the maximum length of a string in Python. In Python 3, the maximum length of a string is sys.maxsize, which is platform-dependent and typically larger than the memory available on the system.. According to the PEP 8 style guide for Python, identifier names should be kept to a maximum of 79 characters per line, although this is not a hard limit.

Q13.What are the benefits of using python?

1.Python is a high-level programming language that offers a wide range of benefits, including:

2.Easy to Learn: Python has a straightforward syntax that is easy to read and write, making it an excellent choice for beginners.

3.Versatile: Python can be used for a wide range of tasks, including web development, data analysis, scientific computing, artificial intelligence, and machine learning

.

4.Cross-Platform: Python is a cross-platform language, which means it can run on various operating systems such as Windows, Linux, and macOS.

5.Libraries and Frameworks: Python has a vast collection of libraries and frameworks that can help developers to reduce the development time and increase productivity. For example, popular Python libraries like NumPy, Pandas, and Matplotlib are used extensively for data analysis and visualization.

6.Scalability: Python is highly scalable and can handle large-scale applications and complex systems with ease.

7.Open-Source: Python is an open-source language, which means the source code is freely available and can be modified by anyone.

8.Rapid Prototyping: Python has a rapid prototyping capability, which means developers can quickly create a working prototype of a project, test it, and iterate until it is refined.

Q14.How is memory managed in Python?

In Python, memory management is handled automatically through a system called reference counting.

Reference counting works by keeping track of how many references, or variables, are pointing to an object in memory. Each time a new reference to an object is created, the reference count for that object is incremented. When a reference is deleted or goes out of scope, the reference count for the corresponding object is decremented. When the reference count for an object reaches zero, it means that no references are pointing to that object, and the memory allocated to that object can be freed.

Python also has a garbage collector that runs periodically to identify and clean up any objects that are no longer being referenced, even if their reference count has not yet reached zero. This helps to ensure that memory is not being unnecessarily used, and that the program's performance is not affected by memory leaks.

Q15.How to install python on windows and set path variable?

Here are the steps to install Python on Windows and set the path variable:

1.Download Python: Go to the official Python website (https://www.python.org/downloads/) and download the latest version of Python for Windows.

2.Install Python: Double-click on the downloaded file to launch the installation process. Follow the on-screen instructions to install Python.

3.Add Python to PATH: During the installation process, make sure to select the "Add Python to PATH" option. If you missed this step, you can still add Python to PATH by following the steps below:

a . Open the Start menu and search for "Environment Variables" and click on "Edit the system environment variables."

b. Click on the "Environment Variables" button in the System Properties window.

c. Under "System Variables", scroll down to find "Path" and click on "Edit".

d. Click on "New" and add the path to the Python executable file. The path should be something like "C:\PythonXX;C:\PythonXX\Scripts" (replace "XX" with the version of Python you installed).
e. Click "OK" to save the changes and close all the windows.

Verify the installation: Open the command prompt and type "python" (without quotes) and hit Enter.

Q16.Is indentation required in python?
Yes, indentation is required in Python. Python uses whitespace, specifically indentation, to indicate blocks of code, instead of using braces or other structural markers like some other programming languages.
Proper indentation is crucial in Python because it affects the way the code is executed. Indentation errors can result in syntax errors or logical errors that can cause the program to behave unexpectedly or not work at all.