ASSIGNMENT (29th Jan)

**Q1) Who developed Python Programming Language?**

1. **Guido van Rossum** createdPython Programming Language.

**Q2) Which type of Programming does Python support?**

1. Python is a general-purpose programming language that supports various programming paradigms, including:

1)Procedural programming: Python supports procedural programming, where you organize your program as a series of procedures or functions.

2)Object-Oriented programming: Python supports object-oriented programming where you can create classes and objects that encapsulate data and methods.

3)Functional programming: Python supports functional programming concepts, where functions are treated as first-class objects.

4)Event-driven programming: Python has libraries that support event-driven programming, such as the asyncio library for asynchronous programming.

5)Scripting: Python is a popular scripting language, where you can write scripts to automate tasks or run commands.

Overall, Python's versatility and flexibility allow it to be used for a wide range of applications and programming tasks.

**Q3) Is Python case sensitive when dealing with identifiers?**

1. Yes, Python is case sensitive when dealing with identifiers. This means that identifiers such as variable names, function names, and class names that differ only in their case are considered distinct and separate entities. It is important to keep this case sensitivity in mind when writing Python code to ensure that you use the correct spelling and capitalization of identifiers in your program.

**Q4) What is the correct extension of python file?**

1. The correct extension for a Python file is ".py".

**Q5) Is Python code compiled or Interpreted?**

1. Python code is interpreted, not compiled, by default.

When you run a Python program, the Python interpreter reads the source code of the program and executes it line by line. The interpreter translates the code into bytecode, which is a lower-level representation of the program that is closer to machine language but is still platform-independent. The bytecode is then executed by the Python virtual machine (PVM).

**Q6) Name a few blocks of code used to define in Python Programming?**

1. In Python programming, blocks of code are defined using indentation, functions, control structures etc.

**Q7) State a character used to give single line comments in Python?**

1. In Python, you can use the hash symbol "#" to add a single-line comment in a code.

**Q8) Mention functions which can help us to find the version of python that we are currently working on?**

1. There are several functions in Python that can be used to find out the version of Python that is currently installed and running on your system:

1) sys.version.

2) platform.python\_version().

3) sys.version\_info.

Example : import sys

print ( sys.version )

**Q9) Python supports the creation of anonymous functions at run time , using a construct called?**

1. Python supports the creation of anonymous functions at run time using a construct called **"lambda**" expressions or **lambda** functions.

A lambda function is a small, anonymous function that can have any number of arguments, but can only have one expression. The lambda function returns the value of the expression when it is called. The syntax for creating a lambda function is as follows:

lambda arguments: expression

Here, "arguments" is a comma-separated list of the function's arguments, and "expression" is the code to be executed when the function is called.

Example: add = lambda x, y: x + y

print (add (2, 3) ) # Output: 5

Lambda functions are often used as arguments to higher-order functions that take other functions as input, such as the "map" and "filter" functions.

**Q10) What does pip stand for Python?**

1. PIP stands for "Pip Installs Packages" or "Pip Installs Python", depending on who you ask. PIP is a package manager for Python that allows you to install and manage third-party libraries and modules that are not part of the standard Python library.

With PIP, you can easily install and update packages, uninstall packages that you no longer need, and manage dependencies between packages. PIP also provides a way to download and install packages from the Python Package Index (PyPI), which is a repository of over 300,000 Python packages contributed by developers from around the world.

PIP is included with most Python installations by default, but you may need to upgrade to the latest version of PIP using the "pip install --upgrade pip" command. Once you have PIP installed, you can use it to manage packages for any Python project that you are working on.

**Q11) Mention few built-in functions in Python?**

1. Python has a large number of built-in functions that are available for use in any Python program. Here are some of the most commonly used built-in functions in Python:

1**) print ()**: The print function is used to display text or variables on the screen or in the console.

2) **input ()**: The input function is used to get user input from the keyboard.

3) **len ()**: The len function is used to get the length of a string, list, tuple, or other sequence.

4) **range ()**: The range function is used to generate a sequence of numbers, which can be used in loops or other contexts.

5**) type ():** The type function is used to get the type of a variable or object.

6) **int ():** The int function is used to convert a string or float to an integer.

7) **str ()**: The str function is used to convert a number or other data type to a string.

8) **float** **()**: The float function is used to convert a string or integer to a float.

9**) list ()**: The list function is used to convert a sequence or other iterable object to a list.

1. **tuple ()**: The tuple function is used to convert a sequence or other iterable object to a tuple.
2. **dict ():** The dict function is used to create a dictionary, which is a collection of key-value pairs.
3. **max ():** The max function is used to get the maximum value in a list or other iterable object.
4. **min ():** The min function is used to get the minimum value in a list or other iterable object.
5. **sum()**: The sum function is used to get the sum of all the elements in a list or other iterable object.

These are just a few of the many built-in functions available in Python. You can find a complete list of built-in functions in the Python documentation.

**Q12) What is the maximum possible length of an identifier in Python?**

1. In Python, the maximum length of an identifier is not explicitly defined. However, the practical limit of an identifier's length is determined by the maximum size of a Python string, which is platform-dependent.

On most platforms, the maximum size of a Python string is limited to 2^31-1 or 2^63-1 bytes, depending on whether Python is running in 32-bit or 64-bit mode. This means that the maximum length of an identifier is limited by the available memory on the system.

In practice, it is rare to encounter an identifier that approaches or exceeds the maximum length of a Python string, as such identifiers would be difficult to read and maintain. It is generally recommended to use concise, descriptive names for variables, functions, and other identifiers in Python code, regardless of the maximum length allowed by the language.

**Q13) What are the benefits of using Python?**

1. Python is a popular programming language with many benefits, some of which include:

**Simple and easy to learn**: Python has a simple and intuitive syntax, which makes it easy to learn and use. It is a great language for beginners and experienced programmers alike.

**Large and supportive community**: Python has a large and active community of developers, who contribute to a variety of open-source projects and libraries. This community provides support, resources, and guidance for users at all levels.

**Cross-platform compatibility**: Python can run on a variety of operating systems, including Windows, macOS, and Linux, making it a highly portable language.

**Extensive library support**: Python has a vast collection of libraries and frameworks, including data analysis, web development, scientific computing, and machine learning, among others. These libraries and frameworks make it easy to build complex applications and projects.

**High-level language**: Python is a high-level language, which means that it abstracts away many low-level details, such as memory management and hardware-specific operations. This makes it easier to write and read code, and reduces the likelihood of errors.

**Rapid development**: Python's simplicity, ease of use, and extensive library support make it well-suited for rapid application development. This allows developers to quickly prototype and iterate on ideas, and bring products to market faster.

Overall, Python is a versatile and powerful programming language that is widely used in a variety of domains, including web development, data analysis, scientific computing, and machine learning, among others. Its simplicity, ease of use, and extensive library support make it a popular choice among developers of all levels.

**Q14) How is memory managed in Python?**

1. Memory management in Python is handled automatically by a mechanism called the Python memory manager. The memory manager is responsible for allocating and deallocating memory as needed for the Python program.

Python uses a technique called reference counting to keep track of objects in memory. Each object in memory has a reference count, which is the number of variables or data structures that refer to the object. When the reference count for an object reaches zero, it is no longer in use and can be deallocated.

In addition to reference counting, Python also uses a technique called garbage collection to free up memory that is no longer in use. The garbage collector periodically scans the memory to identify and deallocate objects that are no longer referenced by any variables or data structures.

Python also includes features such as automatic memory allocation and deallocation, which simplify memory management for developers. For example, Python automatically allocates memory for new objects and deallocates memory for objects that are no longer in use, without the developer having to explicitly manage memory allocation and deallocation.

Overall, Python's memory management system is designed to be efficient and automatic, while also providing developers with the flexibility and control they need to build complex applications and data structures.

**Q15) How to install Python on windows and set path variables?**

1. To install Python on Windows and set the path variables, follow these steps:
2. Download the latest version of Python for Windows from the official website: https://www.python.org/downloads/windows/
3. Run the installer and select the "Add Python to PATH" option during the installation process. This will add the Python executable and associated scripts to your system path.
4. Once the installation is complete, open the command prompt or PowerShell and type "python" to verify that Python is installed and accessible from the command line.
5. If the "python" command does not work, you may need to manually add Python to your system path. To do this, right-click on "This PC" or "My Computer" and select "Properties".
6. Click on "Advanced system settings" and then click on the "Environment Variables" button.
7. In the "System variables" section, scroll down until you find the "Path" variable and click on "Edit".
8. Click on "New" and add the path to the Python executable directory, which should be something like "C:\PythonXX", where "XX" represents the version number of Python you have installed.
9. Click "OK" to save the changes and close all the windows.
10. Open a new command prompt or PowerShell and type "python" to verify that Python is now installed and accessible from the command line.

**Q16) Is indentation required in Python?**

1. Yes, indentation is required in Python. Unlike many other programming languages that use braces or other delimiters to define blocks of code, Python uses indentation to determine the scope of code blocks.