**PYTHON TASK**

**1.What is a Programming language?**  
1. A **programming language** is a formal system of instructions used to communicate with a computer.  
2. Enables developers to write programs for various tasks such as web development, data analysis, and automation.  
3. Examples: C, Java, Python, JavaScript.  
**2.Why Python?**  
1. **Ease of Use:** Simple syntax, ideal for beginners.  
2. **Versatility:** Supports multiple programming paradigms (object-oriented, procedural, functional).  
3. **Extensive Libraries:** Rich ecosystem for tasks like data analysis, machine learning, and web development.  
4. **Community Support:** Large and active community for troubleshooting and learning.  
5. **Interpreted Language:** Python is executed line by line, which makes debugging easier.  
6. **Cross-Platform:** Python programs can run on various platforms without modification (Windows, macOS, Linux).  
7.**Scalability:** While often seen as a scripting language, Python scales well for larger applications when combined with appropriate frameworks or tools (e.g., Django)  
**3.What is Python?**  
1. **Python** is a high-level, interpreted programming language known for its simplicity and readability. It was created by **Guido van Rossum** in the late 1980s and first released in **1991**. Python emphasizes **code readability** and supports multiple programming paradigms, including **object-oriented, procedural,** and **functional programming.**  
a. Monty Python’s Flying Circus  
2. Older than Java (1995).  
**4.Where Python is Used?**  
1**. Web Development:** Frameworks: Django, Flask.  
2. **Data Science and Machine Learning:** Predictive modeling and data analysis using NumPy, Pandas, TensorFlow, scikit-learn.  
3. **Automation (Scripting):** Automating repetitive tasks.  
4. **Game Development:** Framework: PyGame.  
5. **Scientific Computing:** Libraries: SciPy, Matplotlib.  
6. **Artificial Intelligence (AI) and Deep Learning:** Libraries: Keras, PyTorch.  
7. **Web Scraping:** Libraries: BeautifulSoup, Scrapy.  
8. **Cybersecurity:** Used for creating tools to test system vulnerabilities.  
**5.Which Companies using Python?**  
Google, Netflix, Instagram, Spotify, Dropbox, NASA, Uber, Reddit, IBM, 10000 coders dashboard  
**6.Disadvantages of Python?**  
1. Slower Execution Speed  
○ Interpreted language, making it slower than compiled languages like C++ or Java.  
○ Not ideal for performance-critical applications (e.g., real-time systems, gaming engines).  
2. Memory Consumption  
○ Python’s dynamic typing and garbage collection can lead to higher memory usage.  
○ Not suitable for applications where memory optimization is crucial.  
3. Global Interpreter Lock (GIL)  
○ The GIL restricts the execution of multiple threads at the same time.  
○ This makes Python less efficient for multithreaded applications, especially on multi-core processors.  
4. Runtime Errors  
○ Python’s dynamic nature allows runtime errors to occur if code is not thoroughly tested.  
○ This can lead to bugs in production if not handled properly.  
5. Limited Support for Low-Level Programming  
○ Languages like C or Rust are better suited for such tasks.  
**7.Python Installation?**  
1. Download link: Download Python | Python.org  
2. Commands to check installations:  
a. In cmd: python --version  
**8.Different Modes of python Working?**  
Python provides various modes to write and execute code, each suited for different workflows and purposes.  
1. Interactive Mode: Python code is executed directly in the Python shell or terminal.  
a. In cmd: type py or python to enter into the python shell.  
i. exit() to come out of the python shell.  
b. Search idle in the window's search to directly enter the python shell.  
2. Script Mode: a. Write Python programs in a .py file and execute them as a whole.  
b. Run using the command: python filename.py.  
3. Integrated Development Environments (IDEs): a. Use tools like PyCharm, Jupyter Notebook, or VS Code for an enhanced coding experience.  
b. Offers features like debugging, syntax highlighting, and autocompletion.  
**9.What is Variables in Python?**  
a. A variable is a container. Called a variable because we can change the value of a variable.  
b. Snake Case for variables ( using \_’s), Pascalcase for class names, All caps for constants  
c. There are no constants in python. You can only show your intention by writing All Capital letters.  
**10.What is Datatypes in Python?**  
DataTypes (All these are classes)  
a. Built In data types:  
i. Numeric – int, float, complex, bool  
ii. iii. iv. v. Sequence Type – string, list, tuple, Range  
1. There are no characters in Python. They are also strings Mapping Type – dictionary (Key and Value pairs) Set Type – Set None  
1. b. Custom data types:  
i. User defined classes: Custom data types are user-defined classes that can define their own attributes and methods.  
3. Numeric Data types: Numeric types are used to store numerical values.  
a. int: Integer values (whole numbers).  
b. float: Floating-point numbers (decimal values).  
c. complex: Numbers with real and imaginary parts.  
d. bool: Boolean values (True or False).  
**11.What is Operators in Python?**  
Operators  
1. Arithmetic operators - (+, -, \*, /, //, \*\* etc)  
a. / is float division  
b. // is integer division  
2. Assignment operators - (=, +=, -=, \*= etc)  
a. a,b = 2, 3  
b. n = -n  
3. Relational operators - (<, >, <=, >=, ==, !=)  
4. Logical operators - (and, or, not)  
a. Truthy and Falsy values in Python # Falsy: 0, 0.0, '', None, False, [], {}, (), set()  
b. Truthy: Non-zero numbers, non-empty strings, non-empty collections, True.