**Experiment 1**

**To study python programming language & Installation of python on various operating system**

**Requirements:**

Computing System, Python Program

**Introduction to Python**

**1) Brief history of Python**

1. **Origins and Development (Late 1980s - Early 1990s)**

* Python was created by **Guido van Rossum** in the **late 1980s** at **CWI (Centrum Wiskunde & Informatica), Netherlands**.
* It was influenced by **ABC language**, with an emphasis on readability, simplicity, and code reusability.
* Guido started working on Python during the **Christmas holidays of 1989**, aiming to create a better scripting language.

1. **Python 1.0 (1991)**

* Officially released in **February 1991**.
* Featured core data types such as **lists, tuples, dictionaries**, and included exception handling.
* Had **modules, functions, and object-oriented programming** support.

1. **Python 2.x Series (2000 - 2010s)**

* **Python 2.0** was released in **October 2000** with new features like:
  + List comprehensions.
  + Garbage collection with **cycle detection**.
* Python 2 became widely adopted but had many inconsistencies, leading to the need for Python 3.
* **End of life** for Python 2 was declared on **January 1, 2020**.

1. **Python 3.x Series (2008 - Present)**

* **Python 3.0** was released in **December 2008**, introducing:
  + **Print as a function** (print() instead of print statement).
  + **Better Unicode handling**.
  + Improved integer division (5 / 2 = 2.5 instead of 2).
  + More consistent syntax and better library support.
* Despite initial resistance, Python 3 gradually became the standard.

1. **Rise in Popularity (2010s - Present)**

* **Machine Learning & AI**: Libraries like **TensorFlow, PyTorch, and Scikit-learn** made Python the dominant language in AI.
* **Web Development**: Frameworks like **Django and Flask** grew in popularity.
* **Data Science & Automation**: Pandas, NumPy, and Jupyter Notebooks boosted Python’s use in data analysis.
* **Community Growth**: Python became the **most popular programming language**, with extensive industry adoption.

1. **Modern Developments (2020s - Present)**

* Focus on **performance improvements** (e.g., Python 3.11 introduced speed boosts).
* Newer versions introduced **pattern matching (PEP 634)** and **enhanced async capabilities**.
* Python continues to evolve with **better multi-core processing support and JIT compilation research**.

1. **Future of Python**

* Efforts to make Python **faster (Python 3.12 and beyond)**.
* Greater focus on **performance, concurrency, and mobile development**.
* Continued dominance in **AI, automation, and cloud computing**.

**2) Features of Python Program**

1. **Easy to Learn and Readable**

* Python’s simple syntax resembles English, making it beginner-friendly.
* Code readability is enforced through indentation instead of {} brackets.

1. **Interpreted and Dynamically Typed**

* No need for compilation; Python runs code line-by-line using an interpreter.
* Variables do not require explicit type declarations (int, float, string).

1. **Extensive Standard Library**

* Comes with built-in modules for file handling, networking, math, and more.
* Reduces the need for external dependencies in many applications.

1. **Cross-Platform Compatibility**

* Python runs on Windows, macOS, Linux, and even embedded systems.
* Write code once, run it anywhere without modification.

1. **Object-Oriented and Multi-Paradigm**

* Supports Object-Oriented Programming (OOP) with **classes, inheritance, and polymorphism**.
* Also allows **procedural and functional programming** styles.

1. **Support for Multiple Libraries & Frameworks**

* Vast ecosystem of libraries for various domains:
  + **Web Development** → Django, Flask
  + **Data Science & AI** → NumPy, Pandas, TensorFlow
  + **Automation & Scripting** → Selenium, Paramiko

1. **Automatic Memory Management (Garbage Collection)**

* Python manages memory allocation and deallocation automatically.
* Unused objects are removed using a built-in garbage collector.

1. **Large and Active Community Support**

* One of the most active programming communities with regular updates.
* Extensive documentation, forums, and open-source contributions.