Python Programming Introduction and Comparison Notes

I. Features of Python 🐍

Python is a **dynamic, high-level, free and open-source, and interpreted** language. Key features include:

- **Easy to Code & Read:** Simple, concise syntax using **indentation** for code blocks.
- **Object-Oriented** and supports other paradigms (procedural, functional).
- **Portable (Cross-platform):** Runs on different operating systems (Windows, Linux, macOS).
- **Dynamically Typed:** Variable types are checked at **runtime**, not at declaration.
- **Large Standard Library** for various applications (e.g., data science, web development).

II. Execution and PVM

The Python program execution process:

1. **Compilation Phase:** The source code (`.py`) is converted line by line into **Python Bytecode** (`.pyc`).
2. **Execution Phase:** The **Python Virtual Machine (PVM)** executes the bytecode, converting it to **native machine code** for the CPU.

- **Viewing the Bytecode:** The bytecode is typically stored in a `__pycache__` folder with a **`.pyc`** extension to speed up subsequent runs.
- **PVM:** The **runtime engine** that makes Python platform-independent.

III. Frozen Binaries

A **frozen binary** is a standalone executable program that includes the Python **bytecode** and a minimal copy of the **Python interpreter (PVM)**.

- **Purpose:** Allows the application to be run on a user's machine **without requiring a pre-installed Python environment**.

IV. Memory Management

Python uses **automatic** memory management:

- All objects and data structures are stored in a **private heap space**.
- Memory deallocation is handled primarily by **Garbage Collection**, which employs two main methods:
    1. **Reference Counting:** Frees memory as soon as an object's reference count drops to zero.

2. **Generational Garbage Collection:** Cleans up complex **reference cycles** (objects that point to each other but are no longer accessible).

V. Comparison: C vs Python

Aspect	C	Python
Type	**Compiled** Language	**Interpreted** Language
Typing	**Statically Typed** (Type must be declared)	**Dynamically Typed** (Type checked at runtime)
Speed	**Faster** (Compiles directly to machine code)	**Slower** (Interpreted through PVM)
Memory	**Manual** memory management	**Automatic** memory management (Reference counting & GC)

VI. Comparison: Java vs Python

Aspect	Java	Python
Type	**Compiled** to Bytecode, then **Interpreted/JIT** by JVM	**Interpreted** to Bytecode, then **Executed** by PVM
Typing	**Statically Typed**	**Dynamically Typed**

Speed	**Generally Faster** (due to Just-In-Time compilation)	**Generally Slower** (purely interpreted)
Syntax	More **verbose** and structured	More **concise** and readable
Memory	Automatic with **JVM Garbage Collector**	Automatic with **Reference Counting & GC**
http://googleusercontent.com/action_card_content/1		