Introduction to Python and computer programming

**1. Basics of Computer Programming**

* **Purpose of a Program**:
  + A program makes a computer usable, like a player makes a piano playable.
  + Without programs, computers can only perform very basic operations (add, divide, etc.).
* **Nature of Computers**:
  + Extremely fast but only capable of executing simple instructions.
  + Need to be explicitly told what to do, step-by-step.
* **Example**: Calculating average speed requires instructions to:
  + Accept distance
  + Accept time
  + Divide distance by time
  + Display result

**2. Languages in Computing**

* **Natural Languages**: Human communication (English, French, body language).
* **Machine Language**:
  + Native language of computers.
  + Consists of very simple, predefined instructions.
  + Each computer type has its own **Instruction List (IL)**.
* **High-Level Languages**:
  + Bridge between human language and machine language.
  + More readable to humans (e.g., Python, Java, C++).
  + Programs written in them are called **source code** (stored in source files).

**3. Elements of Any Language**

* **Alphabet**: Set of symbols used to build words.
* **Lexis**: Dictionary of valid words.
* **Syntax**: Rules for arranging words into valid sentences.
* **Semantics**: Meaning behind the sentence.

**4. Compilation vs. Interpretation**

* **Compilation**:
  + Source code → machine code once → executable file.
  + Faster execution after compilation.
  + Code is harder to reverse-engineer.
  + Disadvantage: must recompile after any change; different compilers needed for different platforms.
* **Interpretation**:
  + Source code is read and executed line-by-line by an interpreter.
  + Easier to run immediately.
  + More portable (same source code can run on multiple platforms).
  + Disadvantage: slower execution, interpreter needed by end-user.
* **Python**:
  + Interpreted language.
  + Requires a Python interpreter to run.
  + Often called a **scripting language**.

**5. Python – Overview**

* **Type**: Interpreted, high-level, object-oriented, general-purpose language.
* **Creator**: Guido van Rossum (1989) as a hobby project.
* **Goals**:
  + Easy and intuitive but powerful.
  + Open-source.
  + Readable like plain English.
  + Suitable for everyday tasks with quick development cycles.
* **Advantages**:
  + Easy to learn, teach, write, and read.
  + Free, open-source, multiplatform.
* **Drawbacks**:
  + Slower than some compiled languages.
  + Sometimes trickier to debug.

**6. Python in the Real World**

* **Used for**:
  + Web services, cloud tools, social media.
  + Scientific computing.
  + Automation/testing.
* **Not Ideal For**:
  + Low-level system programming.
  + Mobile app development (still evolving here).
* **Competitors**: Perl (traditional), Ruby (innovative).

**7. Python Versions**

* **Python 2**:
  + Older, no major feature updates.
  + Still in use for legacy projects.
* **Python 3**:
  + Current and evolving.
  + Not backward-compatible with Python 2.
  + Recommended for all new projects.
* **Backward Compatibility**:
  + Python 3.x versions are compatible with previous Python 3 releases.

**8. Different Implementations**

* **CPython**:
  + Default Python from Python Software Foundation.
  + Written in C.
* **Cython**:
  + Converts Python code to C for better performance.
* **Jython**:
  + Python implemented in Java (currently Python 2-based).
* **PyPy**:
  + Python implemented in RPython (for experimentation, faster execution).

**❓ Knowledge Check – Q&A**

**Q1.** What is the main purpose of a program for a computer?  
**A:** To provide instructions that make the computer perform useful tasks.

**Q2.** Which is faster for execution: compiled code or interpreted code?  
**A:** Compiled code.

**Q3.** What is Python classified as: compiled, interpreted, or both?  
**A:** Interpreted.

**Q4.** Who created Python and when?  
**A:** Guido van Rossum, in 1989.

**Q5.** Name two advantages of Python.  
**A:** Easy to learn, free and open-source (other acceptable answers: multiplatform, readable code, quick development).

**Q6.** Why can’t Python 2 code run directly in Python 3?  
**A:** The two versions are not backward-compatible.

**Q7.** What does CPython mean?  
**A:** The reference implementation of Python written in C.

**Q8.** Give an example of when Python is not an ideal choice.  
**A:** Low-level programming or mobile app development.