



DAY-1 NOTES

Prof. Sagar Chouksey

Python Trainer, CODING WISE {Awarded as [BEST IT EDUCATION INSTITUTE](#)}

Day 1 of Python 5-Days: Course Notes

Topic 1: What is Programming?

Programming is akin to crafting a set of instructions for a computer to perform specific tasks. It's much like writing a recipe for a dish.

Programming Explained Through Building a House:

1. Planning (Writing the Program):

- Comparable to an architect's role in house design. It involves outlining the program's functionality and objectives.
- Example: Deciding the house's layout and features.

2. Choosing Materials (Selecting Programming Language):

- Each programming language suits different project types, akin to selecting building materials.
- Example: Python is like modern materials; C++ is more traditional.

3. Construction (Coding):

- The phase of actual coding, turning the plan into a working program.
- Example: Building the house as per the design.

4. Inspection (Debugging and Testing):

- Identifying and resolving code issues, similar to house inspections.
- Example: Checking the house's electrical and plumbing systems.

5. Renovation (Updating and Maintenance):

- Regular updates and fixes, like home maintenance.
- Example: Home renovations and repairs.

Topic 2: High vs. Low-Level Languages

- High-Level Language:

- Programmer-friendly and easier to read/write.
- Requires compilation or interpretation.
- Portable across systems.
- Examples: Python, Java, C.

- Low-Level Language:
 - Machine-friendly but harder to read/write.
 - Direct assembly into machine code.
 - Specific to CPU types.
 - Examples: Assembly, Machine Code.

Key Differences:

- High-level languages offer abstraction from hardware, resembling natural language.
 - Low-level languages provide direct hardware control, useful for specific memory or processor tasks.
-

Topic 3: Compiler vs. Interpreter

- Compiler:
 - Translates entire programs before execution.
 - Separates execution from compilation.
 - Faster execution but slower debugging.
 - Examples: C, C++, Rust.
 - Interpreter:
 - Translates programs statement-by-statement.
 - Simultaneous compilation and execution.
 - Slower execution but faster debugging.
 - Examples: Python, Ruby.
-

Topic 4: What is Python?

Python is a high-level, interpreted language known for its simplicity.

- Characteristics:
 - High-Level: User-friendly, abstracting system details.
 - Interpreted: Executes code line-by-line, easing debugging.
 - Dynamically Typed: Automatic type inference.
 - Extensive Libraries: Supports various applications from web development to machine learning.
-

Topic 5: What is IDE?

An IDE is a comprehensive toolset for software development, enhancing productivity and simplifying code writing.

- Key Components:
 - Source code editor, debugger, build automation tools.
 - Some include version control and GUI building tools.
 - Examples: Microsoft Visual Studio, Eclipse, IntelliJ IDEA, PyCharm.

These notes encapsulate the fundamental concepts of programming, different language types, the roles of compilers and interpreters, the basics of Python, and the purpose of an IDE. This foundational knowledge is essential for understanding the subsequent topics in this 5-day Python course.

Submit Assignments by following through this link
<https://codingwise.in/assignmentsubmission/>

★ ***Star this message***
