Python – Short Term Assignment

Module 1) Python Fundamentals

Introduction to Python Theory:

- Introduction to Python and its Features (simple, high-level, interpreted language).

⋆ Introduction to Python

Python is a **simple, high-level, and interpreted programming language** created by **Guido van Rossum** in 1991.
It is widely used in **web development, data science, artificial intelligence, machine learning, automation, and software development**.

⋆ Features of Python

1 Simple Language

Python has a very clean and readable syntax.
Its code looks like English, so it is easy to learn, understand, and write.

---

2 High-Level Language

Python is a high-level language, which means:

- You don’t need to manage memory manually
- You don’t need to understand hardware-level details
  The Python interpreter takes care of these things.

---

3 Interpreted Language

Python is executed **line by line** by an interpreter.
Because of this:

- It gives quick output
- Debugging becomes easier
- No need for separate compilation

---

4 Dynamically Typed

You don’t need to declare variable types.
Python automatically detects the type at runtime.

5️⃣ Object-Oriented

Python supports classes, objects, inheritance, and polymorphism, making it good for large programs.

6️⃣ Large Standard Library

Python has many built-in modules (like math, os, json, datetime) that help in faster development.

7️⃣ Cross-Platform

Python runs on Windows, Linux, and macOS without changing the code.

- History and evolution of Python.

⭑ History of Python

Python was created by **Guido van Rossum**, a Dutch programmer.
He started working on Python in the **late 1980s** at the **National Research Institute for Mathematics and Computer Science (CWI)** in the Netherlands.

Python’s development officially began in **1989**, and the first public release came in **1991**.

Guido wanted to create a language that was:

- Easy to read
- Easy to use
- Powerful enough for real-world applications

Python was inspired by languages like **ABC, Modula-3, C**, and **Unix Shell**.

⭑ Evolution of Python (Important Versions)

1️⃣ Python 1.0 (1991)

- First version released
- Included basic features: functions, modules, exceptions

2 Python 2.0 (2000)

- Major upgrade
- Introduced list comprehensions
- Added garbage collection
- Became widely used for web development
  Python 2 support officially ended in 2020.

3 Python 3.0 (2008)

- Huge improvement
- Not backward-compatible with Python 2
- Cleaner and more powerful design
- Improved Unicode support
- Better libraries and syntax

Today, **Python 3.x** is the standard and most widely used.

★ Modern Evolution

Python continues to grow with updates like:

- Faster performance
- Better support for data science
- Improved libraries for AI, ML, robotics, and automation
- Large community and open-source contributions

★ Why Python Became Popular

- Simple and readable syntax
- High-level and powerful
- Massive library support
- Cross-platform
- Used in trending fields (AI, ML, Data Science)

• Advantages of using Python over other programming languages.

★ 1. Easy to Read and Write

Python has a very simple and clean syntax.
It looks almost like English, so programs are easier to write and understand compared to languages like C, C++, or Java.

⭑ 2. High Productivity

Python allows developers to write fewer lines of code.
This increases productivity and reduces development time.

⭑ 3. Large Standard Library

Python comes with many built-in modules (like `os`, `math`, `json`, `datetime`) that help solve common problems without writing extra code.

⭑ 4. Cross-Platform Language

Python runs on **Windows, Linux, macOS**, and even mobile platforms without needing to change the code.

⭑ 5. Dynamically Typed

You don’t need to declare data types.
Python automatically handles variable types, which makes coding faster and easier.

⭑ 6. Interpreted Language

Python executes code **line by line**, making debugging simpler and faster than compiled languages like C++ or Java.

⭑ 7. Strong Community Support

Python has one of the biggest programming communities.
If you face any error, solutions and tutorials are widely available.

⭑ 8. Excellent for Multiple Domains

Python is used in:

- Web Development
- Artificial Intelligence
- Machine Learning
- Data Science
- Automation / Scripting
- Cybersecurity
- Game Development

No other language dominates so many fields at once.

⭑ 9. Vast Third-Party Libraries

Python has powerful libraries like:

- **NumPy, Pandas** (Data Science)
- **TensorFlow, PyTorch** (Machine Learning)
- **Django, Flask** (Web Development)
- **OpenCV** (Computer Vision)

These make complex tasks very easy.

⭑ 10. Beginner-Friendly

Python is the **best programming language for beginners** because:

- Less code
- Easy syntax
- Fast results

• Installing Python and setting up the development environment (Anaconda, PyCharm, or VS Code).

⭑ Installing Python

1. Go to the **official Python website** (python.org).
2. Download the latest version of Python for Windows, macOS, or Linux.
3. Run the installer and **check "Add Python to PATH"**.
4. Click **Install Now**.
5. After installation, open **Command Prompt** and type:
6.
```
python --version
```

   If it shows the version, Python is successfully installed.

★ Setting Up the Development Environment

You can use any one of the following tools:

1️⃣ Anaconda

Anaconda is a distribution of Python mainly used for **data science, AI, and machine learning**.

How to install:

1. Visit **anaconda.com** and download Anaconda for your OS.
2. Install it using default settings.
3. Open **Anaconda Navigator**.
4. Use Jupyter Notebook, Spyder, or VS Code from Anaconda Navigator.

Why use Anaconda?

- Comes with pre-installed scientific libraries (NumPy, Pandas, Matplotlib).
- Best for data science and ML projects.

2️⃣ PyCharm

PyCharm is a powerful **IDE for Python development**.

How to install:

1. Go to **jetbrains.com/pycharm**.
2. Download **Community Edition (free)**.
3. Install using default settings.
4. Open PyCharm → Create New Project → Select Interpreter (Python.exe).

Why use PyCharm?

- Smart code suggestions
- Debugging tools
- Best for large Python projects

3️⃣ Visual Studio Code (VS Code)

VS Code is a **lightweight and popular code editor**.

How to install:

1. Download VS Code from **code.visualstudio.com**.
2. Install and open it.
3. Go to Extensions (Ctrl + Shift + X).
4. Search and install the **Python extension** by Microsoft.
5. Select Python interpreter:
    - Press **Ctrl + Shift + P**
    - Type **Python: Select Interpreter**
    - Choose your Python version

Why use VS Code?

- Lightweight and fast
- Supports many languages
- Highly customizable
- Extensions for formatting, debugging, Git, etc.

⭑ Summary (Short Points)

- Install Python from python.org.
- Choose one IDE/editor:
  ✓ Anaconda – Best for Data Science
  ✓ PyCharm – Best for professional Python projects
  ✓ VS Code – Lightweight and flexible

• Writing and executing your first Python program.

⭑ Step 1: Open Your Python Environment

You can use any one of the following:

- **IDLE** (comes with Python)
- **VS Code**
- **PyCharm**
- **Jupyter Notebook**
- **Command Prompt / Terminal**

⭑ Step 2: Write Your First Python Program

Type this simple program:

```
print("Hello, World!")
```

This program displays the message **Hello, World!** on the screen.

★ Step 3: Save the File

- Save the file with the extension **.py**
  Example:
- `hello.py`

★ Step 4: Execute the Program

Method 1: Using IDE (IDLE / PyCharm / VS Code)

- Click **Run** or press **F5**
- Output will appear on the screen

Method 2: Using Command Prompt / Terminal

1. Open Command Prompt / Terminal
2. Navigate to the folder where your file is saved
   Example:
3. `cd Desktop`
4. Run the program:
5. `python hello.py`

Output:

```
Hello, World!
```

★ Summary (Short Exam Answer)

To write and execute your first Python program, create a file with `.py` extension, write the code `print("Hello, World!")`, and run it through an IDE or by using the command `python filename.py` in the terminal. The program will display “Hello, World!” as output.

• Write a Python program that prints "Hello, World!".

Lab Program

Program to Print "Hello, World!" in Python

📌 Code:

```
print("Hello, World!")
```

📌 Output:

```
Hello, World~~!~~
```