

Assignment 1

Assignment 1: Introduction to Programming and Tools

Part 1: Introduction to Programming

1. What is programming?

Programming, also known as Coding, is the writing of step-by-step instructions using only logic and rules, in a language that an machine can understand to execute specific tasks, solve problems, or automate processes.

2. Why is programming important in today's world?

Programming play's a very vital role in today's world as it essentially drives all of the technology around us today. From personal equipment like watches, smartphones, appliances, medical equipment, vehicles, to large-scale businesses, inventory, production, sales, accounting, banking, logistics, etc., every thing in all walks of life and industries is running on algorithms programmed into the systems and machines. It helps automate tasks, solve complex problems, and drives innovation, making us more efficient and productive.

3. List at least 3 real-life applications of programming.

1. Traffic Management: Cities are now incorporating algorithms programmed into traffic lights to control and reduce congestion in real-time.
2. Online Banking: 24/7 Secure transactions and account management from anywhere is made possible by relying on programming.
3. Washing Machine: Pre-programmed cycles that control washing process including water temperature, agitation and spin speeds, based on the type of cloths, weight of the cloths and detergents used.

4. What are programming languages? Mention at least 5 popular languages and one use case for each.

Programming languages are a system of notation with it's own unique syntax, structure and vocabulary, that let humans write ideas into instructions that are translated into machine language for computers to run.

Language	Use Case
Python	Data science and machine learning
JavaScript	Interactive web content development
Java	Enterprise and mobile applications
C++	high-performance computing and game development
PHP	Server-side web development

Part 2: Introduction to Python

1. What is Python?

Python is a high-level, interpreted programming language known for its simple syntax, short learning curve and versatility. It's high adoption rate by developers is due to the fact that it allows expressing concepts in fewer lines of code as compared to C/C++. Python is a highly extensible language. Python also comes pre-installed in operating systems like Linux & its variants and MacOS.

2. Who developed Python and when?

Python was first conceived in late 1980's as a hobby project by Guido van Rossum from Netherlands. It was officially first released in 1991.

3. What are the main features of Python?

Python's key features include readable syntax, supports multiple programming styles, cross-platform support, and a vast library of tools.

4. Mention at least 3 areas where Python is widely used (e.g., data science, web development).

1. Data Science

Python is a dominant language in data analysis, used for processing data, statistical analysis, and visualisation (with libraries like pandas, NumPy, and Matplotlib) . Researchers and data scientists use Python to explore datasets and build data-driven insights.

2. Artificial Intelligence

Thanks to libraries such as TensorFlow, Keras, and PyTorch, Python is heavily used for machine learning models and AI research . Its simplicity allows AI

engineers to prototype and experiment quickly in areas like neural networks and data mining.

3. Automation Scripting

Python is often used to write small scripts to automate tasks like file management, testing, or system administration. For instance, a Python script can automatically generate reports or rename thousands of files at once. This ability to “glue” things together makes Python popular as a general scripting language.

5. What makes Python a beginner-friendly language?

Python’s syntax resembles natural English language. This makes it accessible to both newcomers and seasoned programmers. Its interactive shell allows rapid development and testing. Python has a very large support community which helps simplify troubleshooting any issues programmers come across. Python also has a continuously growing list of standard libraries, which programmers can simply import and start using, instead of re-inventing the wheel.

Part 3: Python IDEs

1. What is an IDE (Integrated Development Environment)?

An IDE, or Integrated Development Environment, is a software application that brings together all the essential tools needed for software development into a single, user-friendly application. It combines code editing, debugging, and execution tools into a single interface. It streamlines development by providing syntax highlighting, auto-completion, and version control integration. Just like writers use text editors and accountants use spreadsheets, programmers rely on IDEs to simplify their work.

2. List at least 3 popular IDEs or code editors used for Python development (e.g., PyCharm, VS Code, Jupyter Notebook).

1. PyCharm

A widely recognised IDE developed by JetBrains, specifically built for Python development, offering a comprehensive set of tools for coding, debugging, and version control. It’s available in both a free Community Edition and a more feature-rich paid Professional Edition.

2. VisualStudio Code (VS Code)

A lightweight yet powerful source code editor developed by Microsoft. While it supports numerous programming languages, it’s also excellent for Python development, especially with the installation of the Python extension. VS

Code is free to use and offers extensive customisation through its marketplace of extensions.

3. Jupyter Notebook

A web-based interactive platform that allows users to create and share documents containing live code, equations, visualisations, and narrative text. It's particularly favoured in the data science and machine learning communities. Jupyter Notebook is an open-source project developed by a non-profit foundation named Project Jupyter.

3. Mention one advantage and one disadvantage of each IDE you listed.

IDE	Advantage	Disadvantage
PyCharm	Robust debugging and integration with frameworks like Django	High memory usage for large projects
VS Code	Lightweight with customisable extensions for Python and Jupyter notebooks	Requires manual configuration and extensions for advanced features
Jupyter Notebook	Interactive environment ideal for data exploration and visualisation	Limited debugging tools compared to full IDEs

Part 4: Git and GitHub

1. What is Git?

Git is a Distributed Version Control System that tracks changes in code, helps manage project history, enabling developers to revert back to previous states and collaborate without conflicts.

2. Why do developers use Git?

Developers use Git to preserve project history. It facilitates in parallel development by offering features like branching, merging and contributions from multiple team members working on the same project. It also offers the capability to roll-back to a previous version, if needed.

3. What is GitHub?

GitHub is an online platform by Microsoft that hosts both public and private Git repositories. It makes it easy to for developers and organisations to share code and collaborate with others.

4. What is the difference between Git and GitHub?

Git is the tool for tracking code changes, while GitHub is a cloud platform that hosts Git repositories, offering collaboration features like pull requests and issue tracking. Git is a free and open-source software, while GitHub is a closed-source cloud platform which offers both free and paid hosting of Git repositories.

5. How does GitHub help in collaboration?

GitHub allows developers to fork repositories, propose changes via pull requests, and review code collaboratively. Its issue tracker manages bugs and feature requests, streamlining team workflows. It also offers many other value-added services like GitHub Actions, CodeSpaces, Copilot, etc.

6. What is version control and why is it important in software development?

Version control is a system for managing changes to code over time. It's important because it prevents data loss, documents changes, and resolves conflicts in collaborative projects, ensuring stability and accountability.

Part 5: Introduction to Google Colab

1. What is Google Colab?

Google Colab is a cloud-based IDE for writing and executing Python code via Jupyter Notebooks. It provides free access to GPUs and requires no local setup.

2. How is Google Colab different from other IDEs?

Unlike traditional IDEs, Colab runs entirely in browsers, stores notebooks on Google Drive, and offers pre-installed machine learning libraries like TensorFlow.

3. What are its key features (e.g., free GPU access, cloud-based notebooks)?

- **Free GPU/TPU Access:** Accelerates training of deep learning models.
- **Real-Time Collaboration:** Multiple users edit notebooks simultaneously.
- **Integration with Google Services:** Directly import datasets from Google Drive or BigQuery.

4. What are the common use cases of Google Colab, especially in machine learning or data analysis?

1. **Machine Learning:** Prototyping models with TensorFlow or PyTorch and Training AI models with free power.
2. **Data Analysis:** Playing with data and making visualising datasets using Pandas and Matplotlib.
3. **Education:** Learning to code without installing anything and sharing interactive coding tutorials with students.