**PYTHON ESSENTIALS – 1**

**Internship Report Submitted in partial fulfilment of the requirement for undergraduate degree of**

**BACHELOR OF TECHNOLOGY**

In

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**Koppisetti Charan Siva Sai**

**HU22CSEN0100397**

Under the guidance of

**Mrs. C. Sudha**

**Associate Professor**



**Department Of Computer Science and Engineering**

**GITAM School of Technology**

**GITAM (Deemed to be University)**

**Hyderabad-502329**

**March 2025**



**GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(GITAM)**

**HYDERABAD CAMPUS**

**DECLARATION**

I hereby declare that this summer internship report, titled *"* ***Python Essentials – 1*** *"* is my original work, carried out in the Department of Computer Science and Engineering at GITAM School of Technology, GITAM (Deemed to be University). This report is submitted as part of the partial fulfilment of the requirements for the Bachelor of Technology degree in Computer Science and Engineering.

I also confirm that this work has not been submitted to any other college or university for the award of any degree or diploma.

**Place:** Hyderabad **Koppisetti Charan Siva Sai**

**Date:** 07-03-2025 **HU22CSEN0100397**



**GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**(GITAM)**

**HYDERABAD CAMPUS**

**CERTIFICATE**

This is to certify that the internship report titled *"* ***Python Essentials – 1*** *"* is a genuine record of work carried out by **Koppisetti Charan Siva Sai (HU22CSEN0100397)**. This report has been submitted as part of the partial fulfilment of the requirements for the award of the Bachelor of Technology degree in Computer Science and Engineering.

**Mrs. C. Sudha Dr. Sheikh Mahaboob Basha**

Assistant Professor Professor and HOD

Dept. of CSE Dept. of CSE

**CERTIFICATE OF COMPLETION**

****

**ACKNOWLEDGEMENT**

The success of this internship is not just the result of my efforts but also the guidance and support of many individuals. I am truly grateful for their encouragement throughout this journey.

First and foremost, I would like to express my heartfelt gratitude to **Dr. D. SambaSiva Rao**, Pro Vice Chancellor, GITAM Hyderabad, and **Dr. N. Seetharamaiah**, Principal, GITAM Hyderabad, for their invaluable support.

A special thanks to **Shaik Mahaboob Basha**, Head of the Computer Science and Engineering Department, for giving me this wonderful opportunity to enhance my knowledge in my field and for providing the necessary guidance to prepare this internship report. This experience has truly helped me understand the practical significance of what we study.

I am also deeply thankful to **Mrs. C. Sudha**, whose guidance and support played a crucial role in making this internship a success.

Lastly, I extend my sincere appreciation to my friends for their continuous help in keeping my work well-organized and structured till the very end. Their support made this journey much smoother.

Koppisetti Charan Siva Sai

HU22CSEN0100397

**ABSTRACT**

This report documents my learning experience during the completion of the ***Python Essentials - 1*** certification course at Cisco Networking Academy as part of my internship. The course provided a strong foundation in Python programming, covering fundamental concepts such as data types, operators, control structures, functions, and error handling. Through interactive exercises and hands-on coding challenges, I gained practical experience in writing efficient and structured Python code.

The course also introduced essential programming paradigms, problem-solving techniques, and best practices, reinforcing my ability to develop real-world applications. Successfully completing this certification has enhanced my proficiency in Python and strengthened my understanding of core programming concepts, laying a solid groundwork for advanced studies and professional development in software development and data science.

**TABLE OF CONTENTS**

**Module -1: Introduction to Python and Computer Programming**

1. Introduction to Programming
2. Introduction to Python
3. Downloading and Installing Python

**Module – 2: Python Data types, Variables, Operators, and Basic I/O Operations**

1. Python Literals
2. Operators – data manipulation tools
3. Variables
4. Comments
5. Interaction with the user

**Module -3: Boolean Values, Conditional Execution, Loops, Lists and List Processing, Logical and Bitwise Operations**

1. Making decisions in Python
2. Loops in Python
3. Logic and bit operations in Python
4. Lists
5. Sorting simple lists: bubble sort algorithm
6. Operations on lists
7. Lists in advanced applications

**Module -4: Functions, Tuples, Dictionaries, Exceptions and Data Processing**

1. Functions
2. How functions communicate with their environment
3. Returning a result from a function
4. Scopes in Python
5. Creating multi-parameter functions
6. Tuples and dictionaries
7. Exceptions

**Conclusion**

**Module 1: Introduction to Python and Computer Programming**

**1. Introduction to Programming**

Programming is the process of giving a computer a set of instructions to perform specific tasks. It allows us to create software, automate processes, and solve complex problems efficiently. In this section, we explore the fundamental concepts of programming, including how computers interpret code, the role of algorithms, and the importance of logic in writing programs. Understanding these basics helps in developing a problem-solving mindset, which is essential for any programmer.

**2. Introduction to Python**

Python is a powerful yet beginner-friendly programming language known for its simplicity, readability, and versatility. It is widely used in fields such as web development, data science, artificial intelligence, and automation. This section introduces Python’s key features, including its clean syntax, dynamic typing, and extensive libraries, making it an ideal choice for both beginners and experienced developers.

**3. Downloading and Installing Python**

Before diving into coding, setting up Python on a computer is the first step. This section walks through the process of downloading Python from the official website and installing it on different operating systems like Windows, macOS, and Linux. It also covers setting up an Integrated Development Environment (IDE) like IDLE, PyCharm, or VS Code to write and execute Python programs efficiently. Proper installation ensures a smooth coding experience and helps in getting started with hands-on programming.

**Module 2: Python Data Types, Variables, Operators, and Basic I/O Operations**

**1. Python Literals**

Literals are the constant values that we use directly in Python programs. They represent fixed values like numbers, characters, or boolean values. Python supports different types of literals, including:

* **Numeric literals** (e.g., 10, 3.14, 0b101)
* **String literals** (e.g., "Hello, Python!", 'Single quotes work too')
* **Boolean literals** (True and False)
* **Special literal** (None, which represents the absence of a value)

Literals help define the basic building blocks of a program by providing values that the code can process.

**2. Operators – Data Manipulation Tools**

Operators are special symbols or keywords that allow us to perform operations on variables and values. Python provides various types of operators, including:

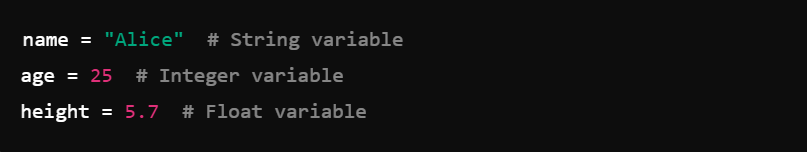
* **Arithmetic Operators** (+, -, \*, /, %, //, \*\*) – Used for mathematical calculations.
* **Comparison Operators** (==, !=, >, <, >=, <=) – Used for comparing values.
* **Logical Operators** (and, or, not) – Used to combine conditions.
* **Assignment Operators** (=, +=, -=, \*=, etc.) – Used to assign values to variables.

Operators are essential for performing calculations and making decisions within a program.

**3. Variables**

Variables act as containers that store data in a program. Unlike some other programming languages, Python doesn’t require explicit data type declarations—the type of a variable is determined dynamically.

**For example:**



Variables make programs flexible, allowing us to store and manipulate data efficiently.

**4. Comments**

Comments are non-executable lines in a program that help developers document their code. They improve readability and make it easier to understand what each part of the code does. Python supports:

* **Single-line comments** (using #):



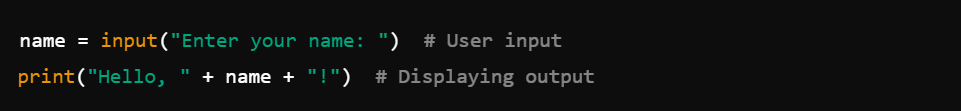
* **Multi-line comments** (using triple quotes):



Good commenting practices make code more maintainable and easier to debug.

**5. Interaction with the User**

Python allows programs to interact with users by taking input and displaying output. The input() function is used to get user input, while print() is used to display information on the screen.  
**Example:**



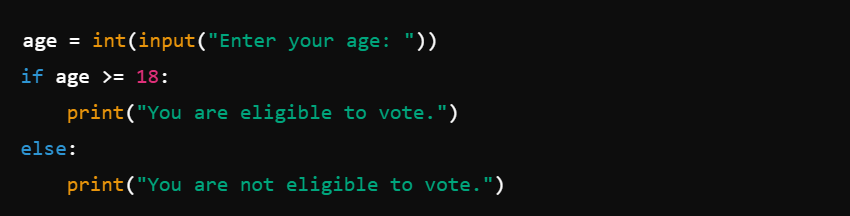
This feature is crucial for creating interactive applications where users can provide data dynamically, making programs more engaging and useful.

**Module 3: Boolean Values, Conditional Execution, Loops, Lists, and Logical Operations**

**1. Making Decisions in Python**

In programming, decision-making is essential to control the flow of execution. Python provides conditional statements like if, elif, and else to make logical decisions based on conditions.

**Example:**



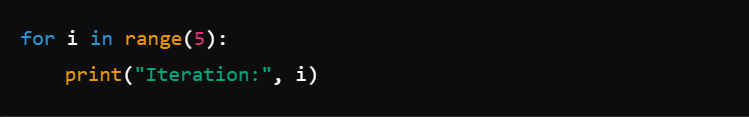
Conditional execution allows programs to respond dynamically to different situations.

**2. Loops in Python**

Loops help execute a block of code multiple times. Python supports:

* **for loop** – Used when the number of iterations is known.
* **while loop** – Used when looping depends on a condition.

**Example of a for loop:**



**Example of a while loop:**

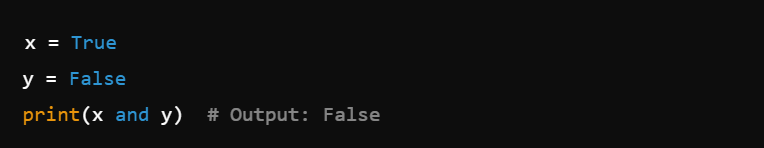


Loops help automate repetitive tasks, making programs efficient.

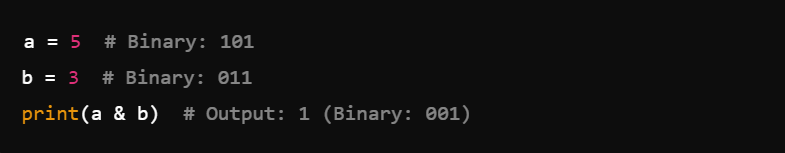
**3. Logic and Bit Operations in Python**

Python provides **logical operators** (and, or, not) for decision-making and **bitwise operators** (&, |, ^, ~, <<, >>) for manipulating binary data.

**Example of logical operations:**



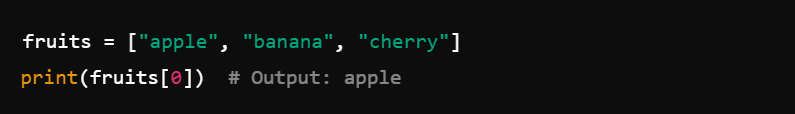
**Example of bitwise operations:**

These operations are useful for working with Boolean logic and low-level data processing.

**4. Lists**

A list is a collection of **ordered, mutable** elements. Lists can store multiple values in a single variable.

**Example:**

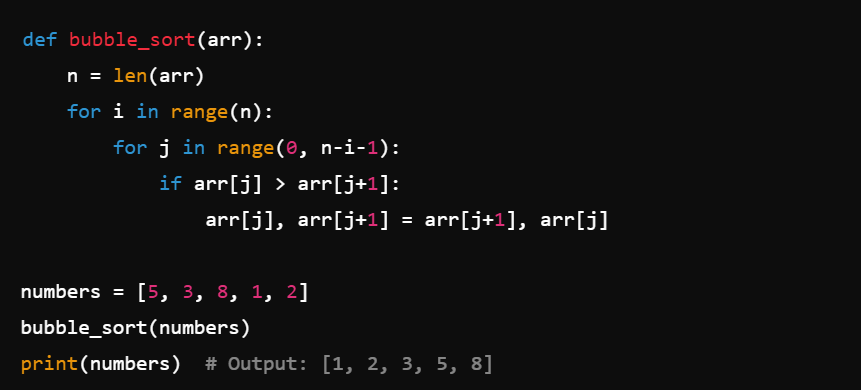


Lists allow easy storage and manipulation of data.

**5. Sorting Simple Lists: Bubble Sort Algorithm**

Bubble Sort is a basic sorting algorithm that repeatedly swaps adjacent elements if they are in the wrong order.

**Example:**

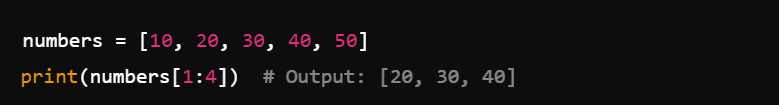


Sorting helps organize data for quick access and efficient processing.

**6. Operations on Lists**

Python provides various operations to modify lists:

* **Adding elements**: append(), insert(), extend()
* **Removing elements**: remove(), pop(), clear()
* **Slicing and indexing**:

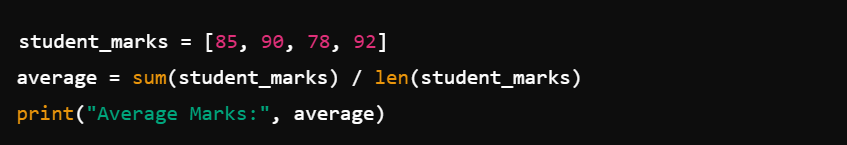
****

**7. Lists in Advanced Applications**

Lists are widely used in real-world applications such as:

* **Data analysis** – Storing and processing datasets.
* **Machine learning** – Handling feature sets.
* **Game development** – Managing player scores or inventory.

**Example of using lists in a real-world scenario:**



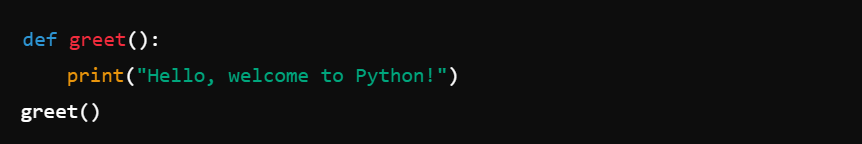
Lists make handling large amounts of data efficient and convenient in programming.

**Module 4: Functions, Tuples, Dictionaries, Exceptions, and Data Processing**

**1. Functions**

Functions are reusable blocks of code that perform a specific task. They help in organizing code, improving readability, and reducing redundancy. In Python, functions are defined using the def keyword.

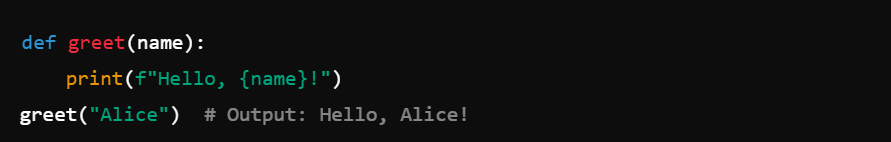
**Example:**

Functions make programs modular and easier to maintain.

**2. How Functions Communicate with Their Environment**

Functions interact with their environment through **parameters (inputs) and return values (outputs)**. They can modify variables, take inputs, and process data from external sources.

**Example:**

This allows functions to be dynamic and reusable.

**3. Returning a Result from a Function**

A function can return a value using the return statement, which allows the calling code to use the computed result.

**Example:**

Returning values makes functions more powerful by allowing further computations.

**4. Scopes in Python**

Scope determines the visibility of variables. There are **local**, **global**, and **nonlocal** scopes in Python.

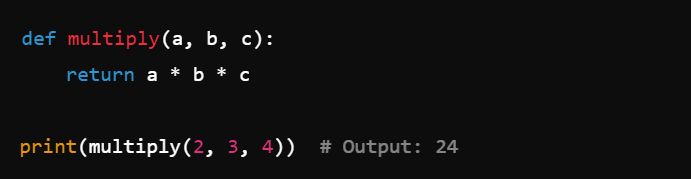
**Example:**

Understanding scope prevents unintended modifications and errors in variable usage.

**5. Creating Multi-Parameter Functions**

Functions can accept multiple parameters, making them more flexible.

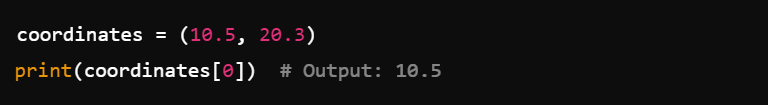
**Example:**

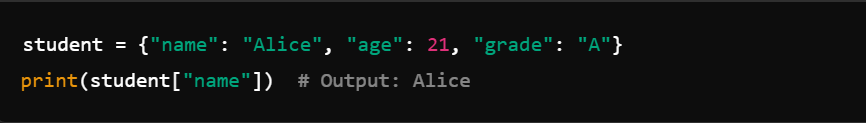


Multi-parameter functions allow complex operations with multiple inputs.

**6. Tuples and Dictionaries**

* **Tuples** are **immutable** sequences used to store multiple items.

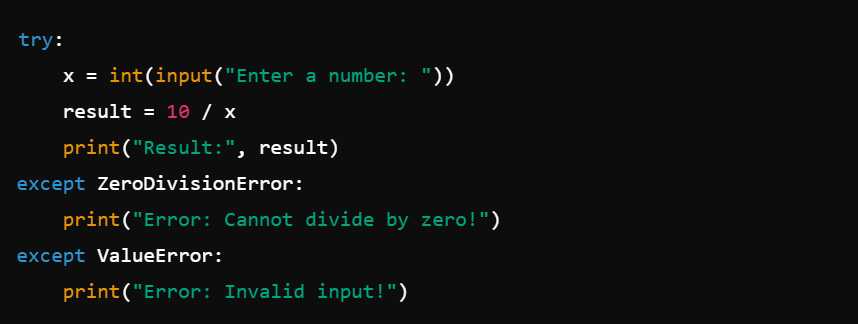
**Dictionaries** store key-value pairs, making data retrieval efficient.

Tuples are useful for fixed data structures, while dictionaries are great for mapping relationships.

**7. Exceptions**

Exceptions handle errors that might occur during program execution, preventing crashes. Python provides try-except blocks for error handling.

**Example:**

Using exceptions makes programs more robust and user-friendly by handling unexpected errors.

**CONCLUSION**

The ***Python Essentials - 1*** course provided a comprehensive introduction to Python programming, covering fundamental concepts essential for writing efficient and structured code.

* **Module 1** introduced the basics of programming, Python’s features, and the installation process, laying the foundation for hands-on coding.
* **Module 2** explored data types, variables, operators, and user interaction, enabling effective data manipulation and input handling.
* **Module 3** focused on decision-making, loops, list operations, and logical/bitwise operations, enhancing problem-solving and algorithmic thinking.
* **Module 4** covered functions, scopes, tuples, dictionaries, and exception handling, reinforcing structured programming and data processing skills.

By completing this course, I have gained a solid understanding of Python’s core principles and its real-world applications. The hands-on exercises and coding challenges helped develop my logical thinking and problem-solving abilities, preparing me for more advanced topics in programming and software development.