# UNIT-1

## Puzzles

1. Mystery Variable Puzzle

   - Puzzle: A variable `x` is initialized to 10. Through a series of operations, its value changes several times. Can you determine the final value of `x`?

   - Steps:

     - `x = 10`

     - `x = x + 5`

     - `x = x  2`

     - `x = x - 8`

   - Solution: `x = 24`

2. Conditional Maze

   - Puzzle: You’re navigating through a maze where each turn is decided by a Python conditional statement. At each step, you choose a direction based on whether a condition is true or false. Can you find the correct path through the maze?

   - Example:

     - `if x > 5:`

     - `turn right`

     - `else:`

     - `turn left`

   - Challenge: Determine the correct series of conditions and decisions to exit the maze.

3. Loop Countdown Puzzle

   - Puzzle: Write a loop that counts down from 10 to 1, but only prints numbers that are divisible by 2.

   - Solution: The loop should print 10, 8, 6, 4, 2.

4. Data Type Detective

   - Puzzle: You have a list of values. Identify the correct data type for each value without running the code.

   - Values: `42`, `"Hello"`, `3.14`, `[1, 2, 3]`, `True`

- Solution: `int`, `str`, `float`, `list`, `bool`

5. Variable Swap Challenge

- Puzzle: Given two variables `a` and `b`, swap their values without using a third variable.

- Solution: `a, b = b, a`

**Multiple Choice Questions (MCQs)**

1. What is the correct way to initialize a variable in Python?

- a) `int x = 10`

- b) `x: int = 10`

- c) `x = 10`

- d) `x := 10`

- Answer: c) `x = 10`

2. Which of the following is a mutable data type in Python?

- a) `int`

- b) `tuple`

- c) `str`

- d) `list`

- Answer: d) `list`

3. What will be the output of the following code?

```python
x = 5
if x > 3:
    x = x  2
else:
    x = x + 2
print(x)
```

- a) `5`

- b) `7`

- c) `10`

- d) `3`

- Answer: c) `10`

4. Which of the following loops will execute the block of code exactly 5 times?

- a) `for i in range(6):`

- b) `for i in range(1, 5):`

- c) `for i in range(0, 5):`

- d) `while i < 5:`

- Answer: c) `for i in range(0, 5):`

5. Which operator is used to check if two values are equal in Python?

- a) `==`

- b) `=`

- c) `!=`

- d) `<>`

- Answer: a) `==`

6. What is the output of the following code?

```python
x = "Hello"
print(x  3)
```

- a) `HelloHelloHello`

- b) `Hello 3`

- c) `Hello, Hello, Hello`

- d) `Hello  3`

- Answer: a) `HelloHelloHello`

7. Which of the following is not a valid data type in Python?

- a) `str`
- b) `float`
- c) `real`
- d) `bool`
- Answer: c) `real`

8. What is the output of the following loop?

```python
for i in range(5, 10):
    if i % 2 == 0:
        print(i, end=" ")
```

- a) `5 6 7 8 9`
- b) `6 8`
- c) `5 7 9`
- d) `6 8 10`
- Answer: b) `6 8`

9. How would you write a comment in Python?

- a) `// This is a comment`
- b) ` This is a comment`
- c) `/ This is a comment /`
- d) `-- This is a comment`
- Answer: b) ` This is a comment`

10. Which statement will correctly check if `a` is not equal to `b`?

- a) `a != b`
- b) `a =! b`
- c) `a <> b`
- d) `a = b`

- Answer: a) `a != b`

## Workbook Content

Chapter 1: Introduction to Python

- Overview: Introduction to Python, its features, and applications in engineering.
- Exercise: Write a Python script that prints "Welcome to Engineering with Python!".

Chapter 2: Variables and Data Types

- Overview: Understanding variables, data types (int, float, str, list, bool), and basic operations.
- Exercise: Define variables of different data types and perform basic arithmetic operations.

Chapter 3: Basic Operations

- Overview: Explore arithmetic, comparison, and logical operations.
- Exercise: Create a calculator program that takes two numbers and an operator as input and performs the corresponding operation.

Chapter 4: Conditional Statements

- Overview: Introduction to `if`, `elif`, and `else` statements.
- Exercise: Write a Python program that checks whether a number is positive, negative, or zero.

Chapter 5: Loops

- Overview: Understanding `for` and `while` loops, and their applications.
- Exercise: Write a Python program that prints the first 10 Fibonacci numbers using a loop.

Chapter 6: Practical Applications in Engineering

- Overview: Apply the concepts learned to solve basic engineering problems using Python.
- Exercise: Create a Python program that calculates the factorial of a number using a loop.

Chapter 7: Case Studies

- Overview: Real-world engineering case studies where Python can be applied.

- Exercise: Develop a Python script to automate a simple engineering task, like unit conversion or calculating resistance in a circuit.

## Chapter 8: Review and Practice

- Overview: Review key concepts and practice with additional exercises and challenges.

- Exercise: Solve a set of mixed problems involving variables, data types, operations, conditional statements, and loops.

## Chapter 9: Project

- Overview: Combine all the learned concepts to build a mini-project.

- Exercise: Develop a Python-based application, like a student grade calculator or a simple engineering tool.