**Python Basics Questions Assignment 1 (Data-Science and Generative AI Module 2)**

**Q1) What is Python, and why is it popular?**

**Python** is a **high-level, interpreted programming language** known for its **simplicity, readability, and versatility**. It was created by **Guido van Rossum** and released in **1991**.

Python supports multiple programming paradigms, including:

- **Procedural programming**
- **Object-oriented programming (OOP)**
- **Functional programming**

Python's popularity has soared due to several key factors:

1) Easy to Learn and Read
2) Versatile & General Purpose
3) Large Standard Library & Community
4) Dominant in Data Science and AI
5) Cross-platform Compatibility

**Q2) What is an interpreter in Python?**

An interpreter in Python is a program that reads and executes Python code line by line. It translates the high-level human-readable Python code into machine code that your computer can understand and run one line at a time.

The **Python interpreter**:

1. Reads the line: print("Hello, world!")
2. Converts it into low-level instructions the machine understands
3. Executes it immediately
4. Moves on to the next line (if any)

Q3) **What are pre-defined keywords in Python?**

Pre-defined keywords in Python are **reserved words** that have **special meaning** to the Python interpreter. They are used to define the **syntax and structure** of the language, such as control flow, functions, loops, logic, and exception handling. These keywords **cannot be used as variable names**, function names, or identifiers. Python keywords are **case-sensitive** and always written in lowercase (except True, False, and None).

**Q4) Can keywords be used as variable names?**

**No**, Python **keywords cannot be used as variable names**. These keywords are **reserved** by the Python language because they have **special meaning** like controlling flow (if, else, for), defining functions (def), handling exceptions (try, except), and so on.

**Q5. What is Mutability in Python?**

**Mutability** in Python refers to whether or not an object's **value can be changed** after it is created.

- **Mutable objects**: Their content **can be changed** without changing their identity.
- **Immutable objects**: Their content **cannot be changed** after creation. Any change results in a **new object** being created.

**Q6. Why are lists mutable, but tuples are immutable?**

**Lists** are designed to **hold a sequence of items that might need to change** — add, remove, or update elements.

**Tuples**, on the other hand, are meant to **represent fixed collections of items**, like coordinates or records that **shouldn't change**.

**Q7) What is the difference between "==" and "is" operators in Python?**

The == operator **checks if two variables have the same value**.

It compares the **contents** of the objects.

**Q8) What are logical operators in Python?**

**Logical operators** in Python are used to perform **logical operations** on **boolean expressions**. They return either **True** or **False** based on the logic applied.

**Q9) What is Type Casting in Python?**

**Type casting** in Python is the process of **converting the data type** of a value from one type to another. It is used to **change the type** of a variable so it can be used in a context where a different type is required (e.g., converting a string to an integer).

**Q10) What is the difference between implicit and explicit type casting?**

**1. Implicit Type Casting**

- **Done automatically** by Python.
- Python **converts data types internally** when no data loss occurs.

**2. Explicit Type Casting**

- **Done manually** by the programmer using built-in functions.
- Required when Python **cannot automatically convert types**.

**Q11) What is the purpose of conditional statements in Python**

The purpose of conditional statements in Python is to control the flow of the program based on certain conditions. They allow the program to make decisions and execute different blocks of code depending on whether a condition is True or False.

- To perform different actions based on different inputs or situations
- To make the program dynamic and logical
- To avoid executing unnecessary code

**Q12) Describe a scenario where a while loop is more suitable than a for loop.**

A while loop is more suitable when you don't know in advance how many times the loop needs to run  you just want to keep repeating until a certain condition becomes false.

**Q12) How does the elif statement work?**

The elif (short for else if) statement in Python is used to check multiple conditions one by one after an initial if condition. It allows your program to execute different blocks of code based on multiple conditions, without writing many separate if statements.

- elif lets you **chain multiple conditions**.
- Python **executes only the first True condition**, then exits the block.
- An optional else handles the case when **no condition is True**.

**Q13) What is the difference between for and while loops?**

Both for and while loops are used to repeat a block of code, but they differ in how and when they repeat.

1. **for Loop:**

- Used when you know how many times you want to repeat the loop.
- Iterates over a sequence (like list, string, range, etc.)

2. **while Loop:**

- Used when you want to repeat a block until a condition becomes False.
- Best when the number of repetitions is not known in advance.