ASSESSMENT-1

G.Manohar reddy

2211CS010184

GROUP-8

1)Give me some idea on what is a Data type. What are Primitive Data types and Non-Primitive Data types. Explain with some examples.

What is a Data Type?

A **data type** is a classification that specifies which type of value a variable can hold in a programming language. It defines the size, type of operations, and memory storage required for a variable. Data types ensure that the correct type of data is assigned to variables, preventing type-related errors in a program.

Primitive Data Types

Primitive data types are the basic building blocks of data manipulation in programming. They represent single values and are generally predefined by the programming language.

Examples of Primitive Data Types:

Data Type	Description	Example (Java/Python)
Integer (int)	Stores whole numbers	int x = 10; (Java) x = 10 (Python)
Floating Point (float, double)	Stores decimal numbers	float pi = 3.14f; (Java) pi = 3.14 (Python)
Character (char)	Stores a single character	char letter = 'A'; (Java) letter = 'A' (Python uses str for characters)
Boolean (boolean)	Stores true or false values	boolean flag = true; (Java) flag = True (Python)

Non-Primitive Data Types

Non-primitive data types (also called **reference types**) are more complex than primitive types. They refer to objects and can store multiple values or a combination of different data types.

Examples of Non-Primitive Data Types:

Data Type	Description	Example (Java/Python)
String	Stores a sequence of characters	String name = "John"; (Java) name = "John" (Python)
Array	Stores multiple values of the same data type	int[] arr = {1, 2, 3}; (Java) arr = [1, 2, 3] (Python uses lists)
Class	A blueprint for creating objects	class Car {} (Java) class Car: (Python)
Interface (Java)	Defines methods that a class must implement	interface Vehicle {}
List	Dynamic array-like structure	List<Integer> nums = new ArrayList<>(); (Java) nums = [1, 2, 3] (Python)

Key Differences:

Feature	Primitive Data Type	Non-Primitive Data Type
Storage	Stores single values	Stores multiple values or objects
Size	Fixed	Can grow dynamically
Operations	Simple operations (arithmetic, comparison)	Advanced operations (methods, iteration)
Examples	int, float, char, boolean	String, Array, List, Dictionary, Class