**Assignment**

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

A data type is an attribute of a variable which tells the compiler or interpreter how the programmer intends to use the variable. It defines the operations that can be done on the data and what type of values can be stored. According to the properties they possess, data types are divided into two groups:

1. Primitive Datatype
2. Non-Primitive Datatype

**Primitive Data Types:** A primitive data type is pre-defined by the programming language. The size and type of variable values are specified, and it has no additional methods.

**Primitive Datatypes can be:**

1. **Integer(int):** Used to store whole numbers.

* Example: int x = 10;

1. **Float/Double:** Used to store decimal numbers.

* Example: float pi = 3.14;

double z = 3.14159;

1. **Character(char):** Used to store a single character.

* Example: char grade = 'A';

1. **Boolean(bool):** Used to store true or false values.

* Example: bool java = true;

**Example for Primitive Datatypes:-**

class PrimitiveDatatypes {

public static void main(String args[]) {

int x = 10;

float pi = 3.14f;

double z = 3.14159;

char grade = 'A';

boolean java = true;

System.out.println("Integer Value: "+x);

System.out.println("Float Value: "+pi);

System.out.println("Double Value: "+z);

System.out.println("Character type: "+grade);

System.out.println("Boolean type: "+java);

}

}

**Output:**

Integer Value: 10

Float Value: 3.14

Double Value: 3.14159

Character type: A

Boolean type: true

**Characteristics of Primitive Data Types:**

* Simple structure.
* Fixed memory allocation.
* Directly operated upon by the processor.

**Non-Primitive Data Types:** These data types are not actually defined by the programming language but are created by the programmer. They are also called “reference variables” or “object references” since they reference a memory location which stores the data.

**Non-Primitive Datatypes can be:**

1. **Arrays:** A collection of elements of the same type.

* Example: int[] numbers = {1, 2, 3, 4};

1. **Classes and Objects:** Used in object-oriented programming to model real-world entities.

* Example:

class Person {

String name;

int age;

}

Person p = new Person();

p.name = "John";

p.age = 25;

1. **Strings:** Can be treated as non-primitive if implemented as objects.

* Example: String greeting = "Hello!";

1. **Collections:** Data structures like lists, sets, and maps.

* Example:

List: List<String> fruits = new ArrayList<>();

Map: Map<String, Integer> inventory = new HashMap<>();

**Characteristics of Primitive Data Types:**

* Complex structures.
* Can hold multiple values or objects.
* Memory allocation varies depending on the data type.