**\*\*Java Primitives Exercises\*\***

**\*\*Exercise 1: Working with Primitive Data Types\*\***

Create a Java program that declares and initializes variables of the following types:

- `byte`, `short`, `int`, `long`

- `float`, `double`

- `char`, `boolean`

Print their values and experiment with \*\*type conversion (casting)\*\*.

**\*\*Exercise 2: Type Conversion and Casting\*\***

Write a Java program that:

- Converts an `int` to `double`

- Converts a `double` to `int` (using casting)

- Converts a `char` to `int` (ASCII value)

-----------------------------------

**Exercise: Creating a Simple “Car” Class.**

**Instructions**  
1. **Create a Java class** named Car with the following **attributes (fields)**:  
• String brand (e.g., “Toyota”)  
• String model (e.g., “Corolla”)  
• int year (e.g., 2022)  
• double price (e.g., 20000.0)  
2. **Define a constructor** that allows initializing all attributes.  
3. **Create a method** named displayCarInfo() that prints all the car details.  
4. **Write a main method** in a CarTest class to:  
• Create **two Car objects**.  
• Call displayCarInfo() to show their details.

**Challenge 1: Add a Method to Compare Car Prices**

**Task:** Add a method **isMoreExpensiveThan(Car otherCar)** that compares the price of the current car with another car and returns true if it is more expensive, otherwise false.

**Challenge 2: Implement Encapsulation**

**Task:**

• **Make attributes private** and provide **getter** and **setter** methods.

• Ensure the **price cannot be negative** in the setter.

**Challenge 3: Add a Method to Apply Discount**

**Task:** Create a method applyDiscount(double percentage) that reduces the car price by the given percentage.

**Challenge 4: Add a Static Attribute to Count Created Cars**

**Task:**

• Add a **static variable** carCount that keeps track of how many Car objects are created.

• Increment carCount in the constructor.