**Task 1 – Dog class**

**Algorithm**1. Define a class named Dog.

2. Declare two class-level static attributes:

- species = "Canis familiaris"

- numLegs = 4

3. Declare instance-level attributes: name, breed, and age.

4. Define a method bark() that prints "Woof!".

5. In the main method, create a Dog object and call bark().

**Psudocode**

CLASS Dog

STATIC ATTRIBUTE species = "Canis familiaris"

STATIC ATTRIBUTE numLegs = 4

INSTANCE ATTRIBUTES: name, breed, age

METHOD bark()

PRINT "Woof!"

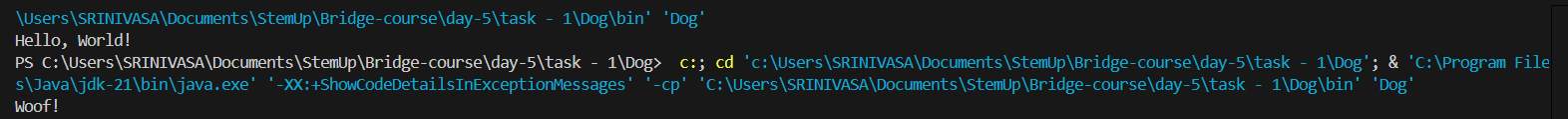
MAIN METHOD

CREATE Dog object

CALL bark() on object

**Code(java)  
**

**Output (test case 1)**



**Observations**- Demonstrates how to use both static (class-level) and instance (object-level) attributes.

- Method bark() belongs to each object and prints a message.

- A good starting point to understand class structure and behavior in Java.

**Task 2 – Book class**

**Algorithm**1. Define a class named Book.

2. Declare instance attributes: title, author, numPages, and isOpen.

3. Define the method openBook() that sets isOpen to true.

4. Define the method closeBook() that sets isOpen to false.

5. In the main method, create a Book object, open and close it, and print its state.

**Psudocode**

CLASS Book

INSTANCE ATTRIBUTES: title, author, numPages, isOpen

METHOD openBook()

SET isOpen = true

METHOD closeBook()

SET isOpen = false

MAIN METHOD

CREATE Book object

CALL openBook()

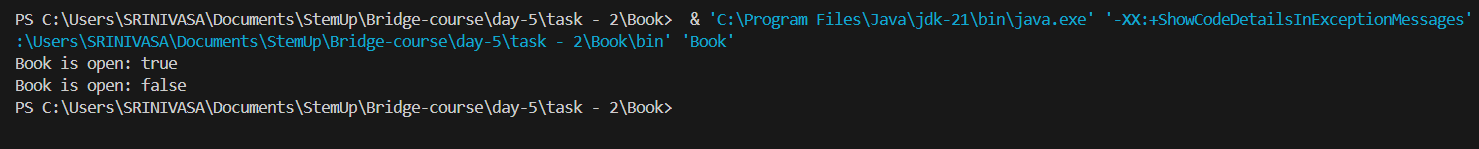
PRINT isOpen status

CALL closeBook()

PRINT isOpen status

**Code(java)  
**

**Output (test case 1)**



**Observations**- Demonstrates class with multiple instance attributes and behavior through methods.

- openBook and closeBook control the internal state of isOpen.

- Reinforces object-oriented principles of encapsulating data and behavior in classes.

**Task 3 – Identify Class Elements for Car Class**

**Algorithm**1. Think of attributes that vary between car objects → instance attributes.

2. Think of actions a car object might perform → instance methods.

3. Identify one class-level attribute common to all car objects.

**Psudocode**

INSTANCE ATTRIBUTES (3–5):

- String brand

- String model

- int year

- String color

- double mileage

INSTANCE METHODS (2–3):

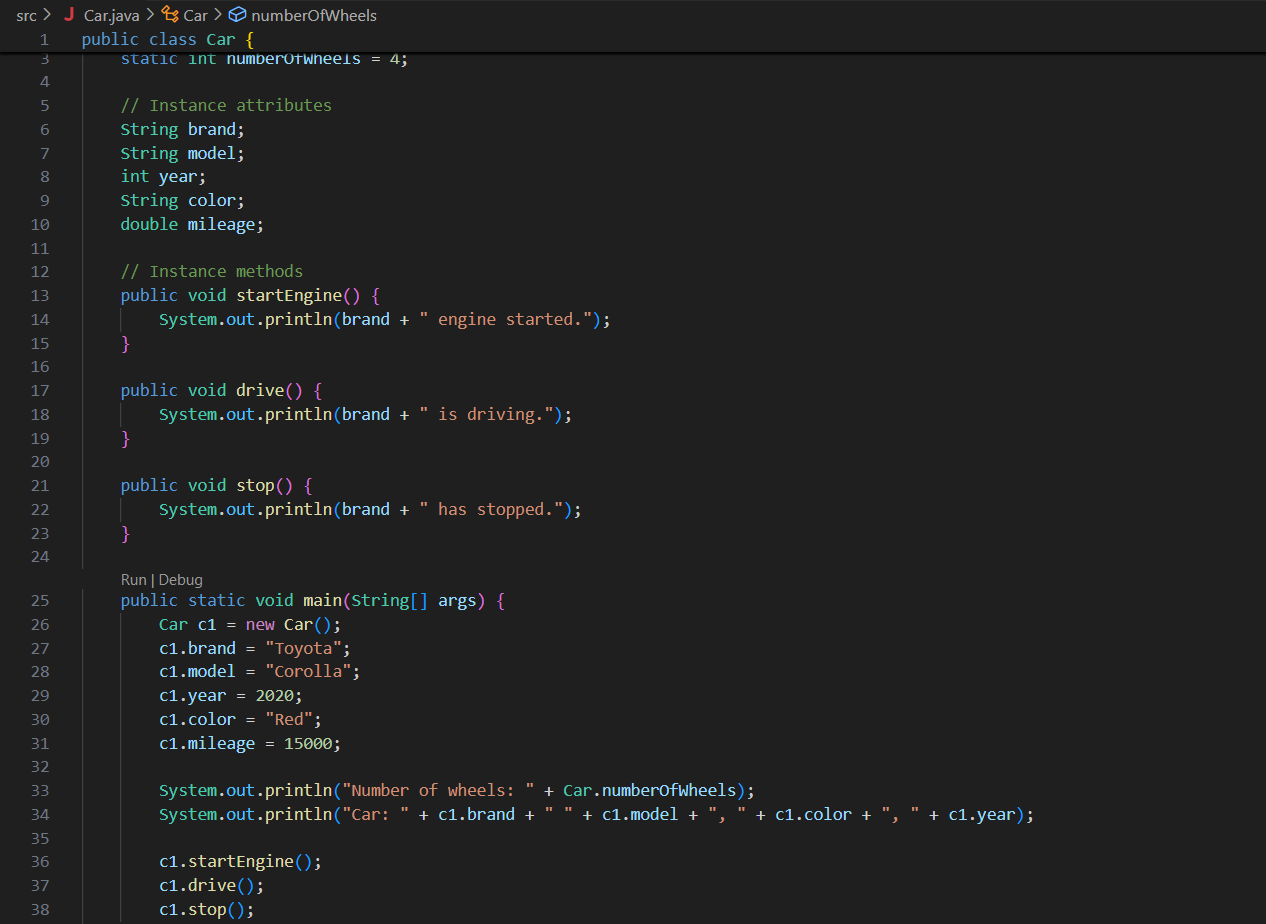
- void startEngine()

- void drive()

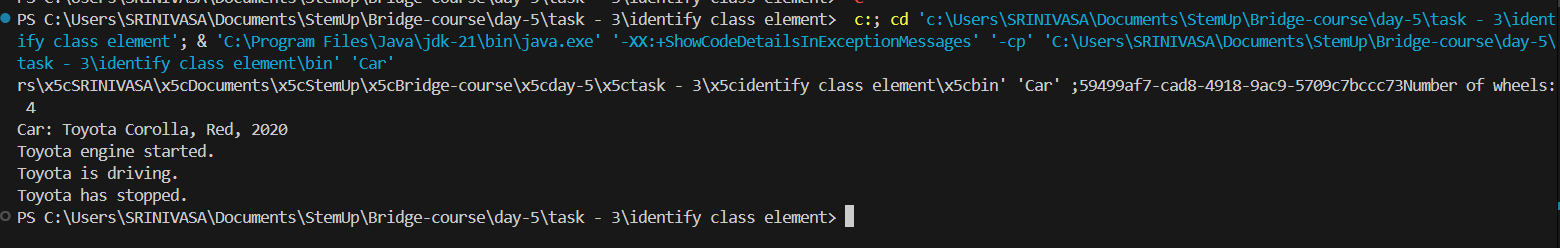
- void stop()

CLASS ATTRIBUTE (shared by all Car objects):

- static int numberOfWheels = 4

**Code(java)  
**

**Output (test case 1)**



**Observations**- Instance attributes define unique characteristics of each car.

- Instance methods define behaviors/actions that cars can perform.

- The class attribute numberOfWheels is shared across all car objects, assuming standard 4-wheeled vehicles.

- This helps practice class design by separating object-specific data from shared characteristics.

**Task 1 –**

**Algorithm**

**Psudocode**

**Code(java)**

**Output (test case 1)**

**Output (test case 2)**

**Output (test case 3)**

**Observations**