Simple Class and Object Creation

CODE:

public class Main {

public static void main(String[] args) {

Car car1 = new Car("Toyota", 2020);

car1.displayCarInfo();

}

}

class Car {

String brand;

int year;

public Car(String brand, int year) {

this.brand = brand;

this.year = year;

}

public void displayCarInfo() {

System.out.println("Car Brand: " + brand);

System.out.println("Car Year: " + year);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Class with Methods and Attributes

CODE:

public class Main {

public static void main(String[] args) {

Student student1 = new Student("John", 20);

student1.study();

student1.displayStudentInfo();

}

}

class Student {

String name;

int age;

public Student(String name, int age) {

this.name = name;

this.age = age;

}

public void study() {

System.out.println(name + " is studying.");

}

public void displayStudentInfo() {

System.out.println("Student Name: " + name);

System.out.println("Student Age: " + age);

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.

Using Constructor Overloading and Objects

CODE:

public class Main {

public static void main(String[] args) {

Rectangle rectangle1 = new Rectangle(5, 10);

Rectangle rectangle2 = new Rectangle(7);

rectangle1.displayArea();

rectangle2.displayArea();

}

}

class Rectangle {

int length;

int width;

public Rectangle(int length, int width) {

this.length = length;

this.width = width;

}

public Rectangle(int side) {

this.length = side;

this.width = side;

}

public void displayArea() {

System.out.println("Area of rectangle: " + (length \* width));

}

}

OUTPUT:

A screenshot of a computer

AI-generated content may be incorrect.