Java Chapter 7 OOP Inheritance

Contents

Time required: 60 minutes

Online Tutorial

Go through the following specific tutorials as a review.

Java OOP

- Java OOP
- Java Classes/Objects
- Java Class Attributes
- Java Class Methods
- Java Constructors
- Java Modifiers
- Java Encapsulation

Java OOP Inheritance

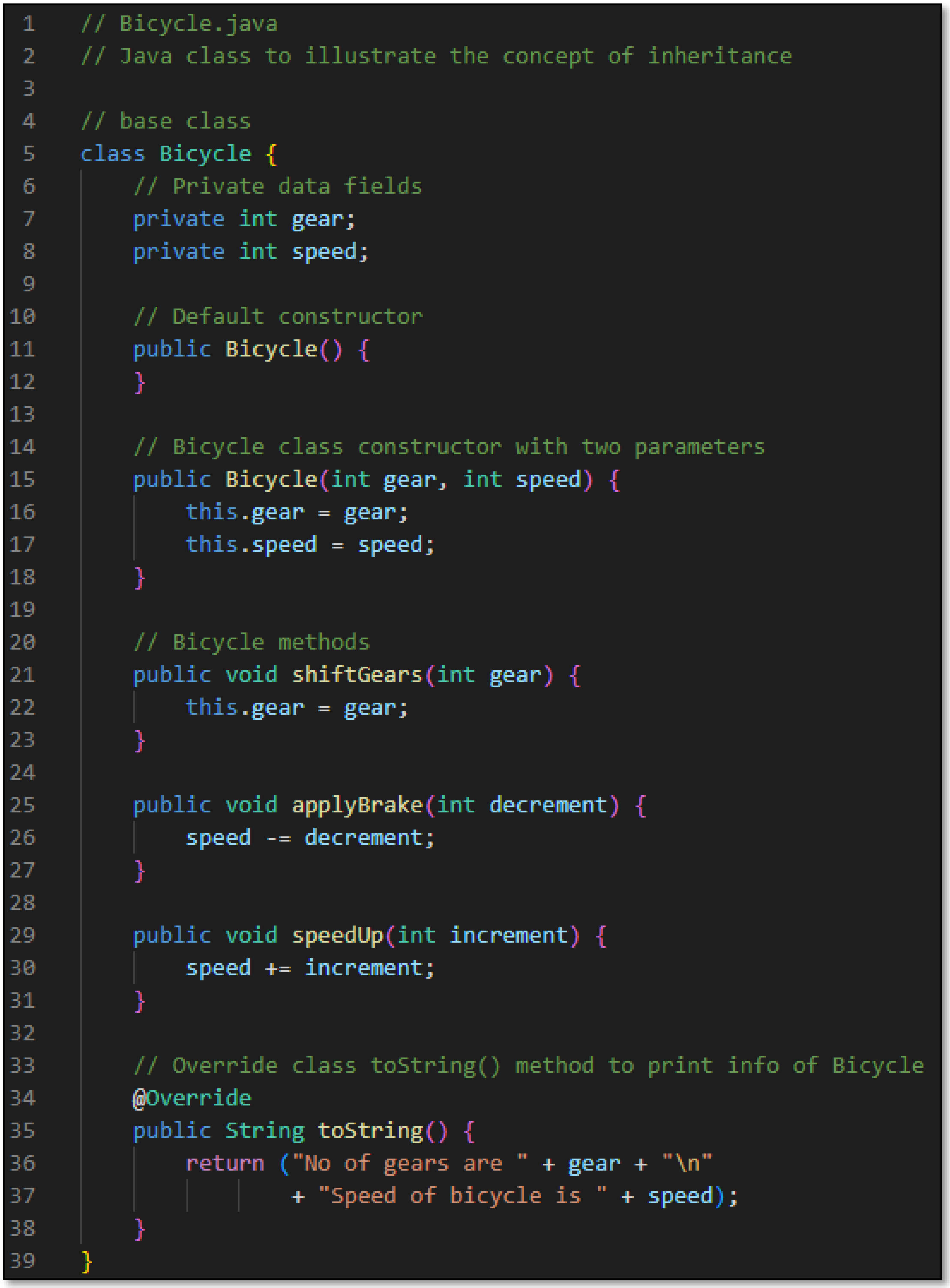
Go through the following tutorials as an introduction to OOP inheritance in Java.

- https://www.w3schools.com/java/java_inheritance.asp
- http://itec2140.gitlab.io/#_object_oriented_programming

- https://www.programiz.com/java-programming/inheritance

Tutorial 1: Bicycle Objects and Inheritance

The bicycle class has the data fields and methods for a regular bicycle.

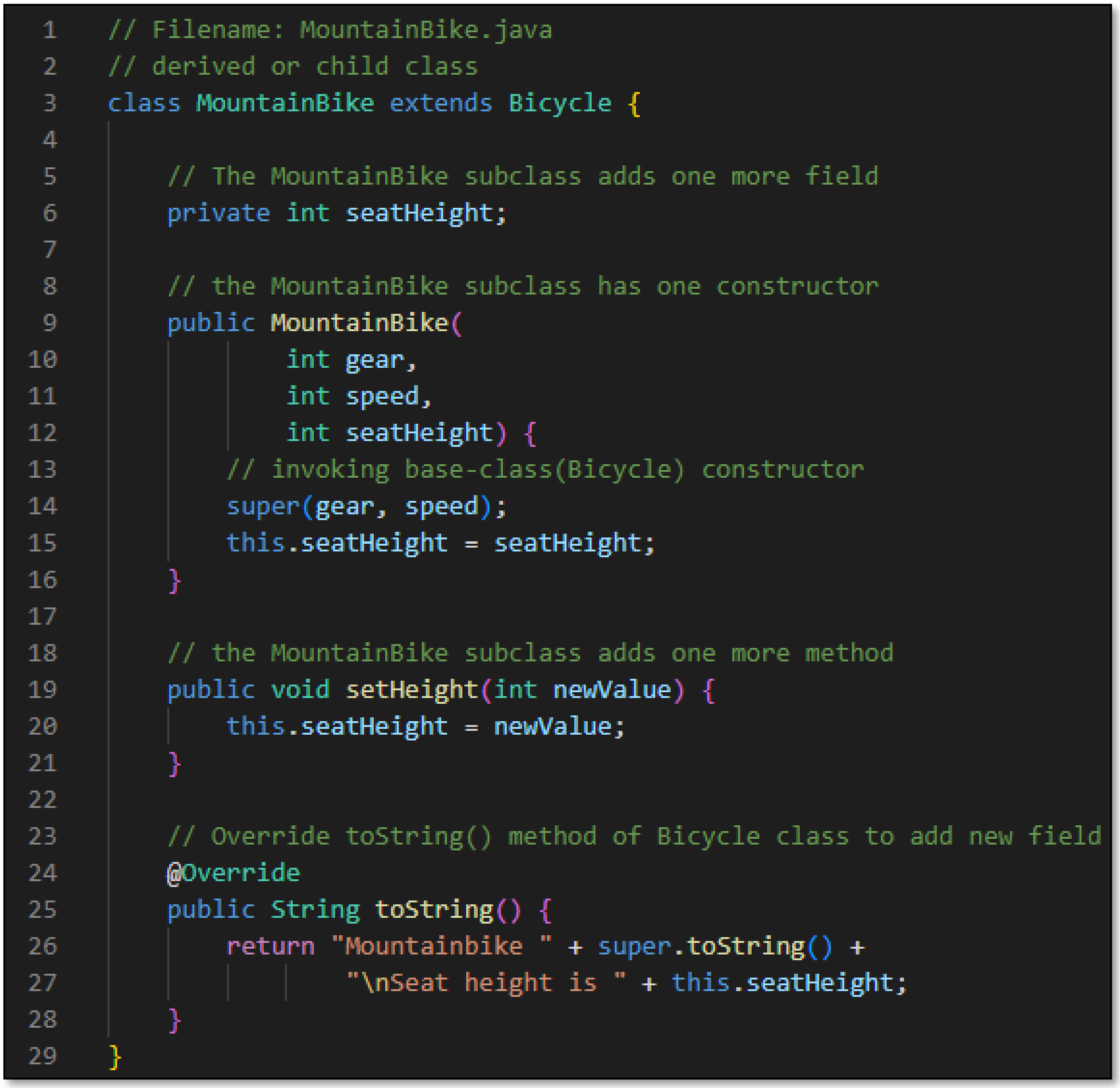


```java
// Bicycle.java
// Java class to illustrate the concept of inheritance

// base class
class Bicycle {
    // Private data fields
    private int gear;
    private int speed;

    // Default constructor
    public Bicycle() {
    }

    // Bicycle class constructor with two parameters
    public Bicycle(int gear, int speed) {
        this.gear = gear;
        this.speed = speed;
    }

    // Bicycle methods
    public void shiftGears(int gear) {
        this.gear = gear;
    }

    public void applyBrake(int decrement) {
        speed -= decrement;
    }

    public void speedUp(int increment) {
        speed += increment;
    }

    // Override class toString() method to print info of Bicycle
    @Override
    public String toString() {
        return ("No of gears are " + gear + "\n"
                + "Speed of bicycle is " + speed);
    }
}
```

Save this class as **Bicycle.java**

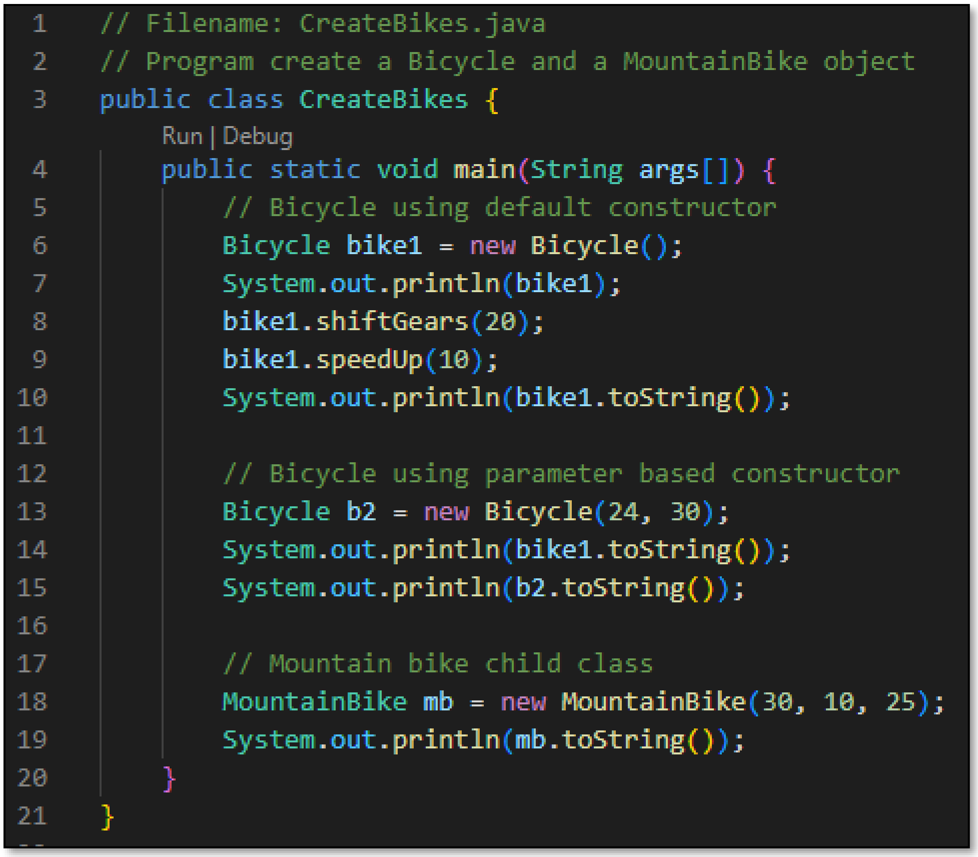
The MountainBike class inherits from the Bicycle class by using the "**extends**" keyword.

Save this class as **MountainBike.java**



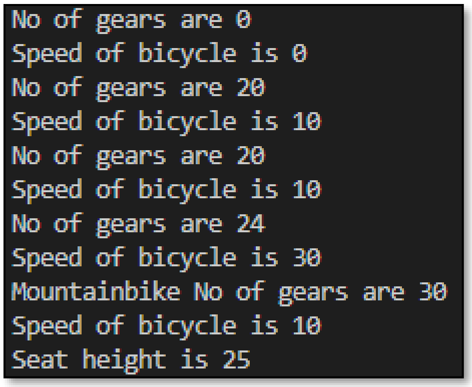
```java
// Filename: MountainBike.java
// derived or child class
class MountainBike extends Bicycle {

    // The MountainBike subclass adds one more field
    private int seatHeight;

    // the MountainBike subclass has one constructor
    public MountainBike(
            int gear,
            int speed,
            int seatHeight) {
        // invoking base-class(Bicycle) constructor
        super(gear, speed);
        this.seatHeight = seatHeight;
    }

    // the MountainBike subclass adds one more method
    public void setHeight(int newValue) {
        this.seatHeight = newValue;
    }

    // Override toString() method of Bicycle class to add new field
    @Override
    public String toString() {
        return "Mountainbike " + super.toString() +
                "\nSeat height is " + this.seatHeight;
    }
}
```

The **CreateBikes.java** program create a Bicycle and a MountainBike object.



```java
// Filename: CreateBikes.java
// Program create a Bicycle and a MountainBike object
public class CreateBikes {
    public static void main(String args[]) {
        // Bicycle using default constructor
        Bicycle bike1 = new Bicycle();
        System.out.println(bike1);
        bike1.shiftGears(20);
        bike1.speedUp(10);
        System.out.println(bike1.toString());

        // Bicycle using parameter based constructor
        Bicycle b2 = new Bicycle(24, 30);
        System.out.println(bike1.toString());
        System.out.println(b2.toString());

        // Mountain bike child class
        MountainBike mb = new MountainBike(30, 10, 25);
        System.out.println(mb.toString());
    }
}
```

Example run:



```
No of gears are 0
Speed of bicycle is 0
No of gears are 20
Speed of bicycle is 10
No of gears are 20
Speed of bicycle is 10
No of gears are 24
Speed of bicycle is 30
Mountainbike No of gears are 30
Speed of bicycle is 10
Seat height is 25
```

Assignment 1: Bake Your Own Cookies

Create and demonstrate your own class and inherited class.

An example might be a cookie class. Chocolate Chip Cookies extends the Cookie class. Ask ChatGPT for ideas for an interesting parent and child class. (No code.) Bake your own cookies.

Assignment Submission

Attach the pseudocode document and the .java files to the assignment in Blackboard.