# 2.1 BIKE PROJECT

## BIKE:

```
package bikeproject;

public class Bike {
        private String handleBars, frame, tyres, seatType;
        private int NumGears;
        private final String make;
        public Bike(){
                this.make = "Oracle Cycles";
        }//end constructor
        public Bike(String handleBars, String frame, String tyres, String seatType, int numGears) {
                this.handleBars = handleBars;
                this.frame = frame;
                this.tyres = tyres;
                this.seatType = seatType;
                NumGears = numGears;
                this.make = "Oracle Cycles";
        }//end constructor
        protected void printDescription()
        {
                System.out.println("\n" + this.make + "\n"
                                        + "This bike has " + this.handleBars + " handlebars on a "
                                        + this.frame + " frame with " + this.NumGears + " gears."
                                        + "\nIt has a " + this.seatType + " seat with " + this.tyres + "
tyres.");
        }//end method printDescription

        // Main method to test the Bike class
        public static void main(String[] args) {
                // Creating a bike using the default constructor
                Bike defaultBike = new Bike();
                defaultBike.printDescription();

                // Creating a bike using the parameterized constructor
                Bike customBike = new Bike("Drop", "Carbon", "Road", "Racing", 18);
                customBike.printDescription();
        }

}//end class Bike
```

## BIKE DRIVER:

```
package bikeproject;

public class BikeDriver {

        public static void main(String[] args) {
```

```
        RoadBike bike1 = new RoadBike();
        RoadBike bike2 = new RoadBike("drop", "tourer", "semi-grip", "comfort", 14, 25,
18);
        MountainBike bike3 = new MountainBike();
        Bike bike4 = new Bike();

        bike1.printDescription();
        bike2.printDescription();
        bike3.printDescription();
        bike4.printDescription();
    }//end method main

}//end class BikeDriver
```

**<u>MOUNTAIN BIKE:</u>**

```
package bikeproject;

public class MountainBike extends Bike {

    private String suspension, type;
    private int frameSize;

    // Default constructor
    public MountainBike() {
        // Calls the parameterized constructor with default values
        this("Bull Horn", "Hardtail", "Maxxis", "dropper", 27, "RockShox XC32", "Pro", 19);
    }//end constructor

    // Parameterized constructor
  public MountainBike(String handleBars, String frame, String tyres, String seatType, int numGears,
            String suspension, String type, int frameSize) {
        super(handleBars, frame, tyres, seatType, numGears);
        this.suspension = suspension;
        this.type = type;
        this.frameSize = frameSize;
    }//end constructor

    // Method to print the bike description
    @Override
    public void printDescription() {
        super.printDescription();
        System.out.println("This mountain bike is a " + this.type + " bike and has a " +
this.suspension + " suspension and a frame size of " + this.frameSize + " inches.");
    }//end method printDescription

    // Main method for testing the MountainBike class
    public static void main(String[] args) {
        // Create a MountainBike using the default constructor
        MountainBike defaultBike = new MountainBike();
        defaultBike.printDescription();

        // Create a MountainBike using the parameterized constructor
        MountainBike customBike = new MountainBike("Flat", "Full Suspension", "Kenda",
"Gel", 21, "Fox Float", "Downhill", 17);
        customBike.printDescription();
    }//end method main

}//end class MountainBike
```

## **ROAD BIKE:**

```
package bikeproject;

public class RoadBike extends Bike {

    private int tyreWidth, postHeight;

    // Default constructor
    public RoadBike() {
        // Calls the parameterized constructor with default values
        this("drop", "racing", "treadless", "razor", 19, 20, 22);
    }//end constructor

    // Constructor with postHeight only
    public RoadBike(int postHeight) {
        // Calls the parameterized constructor with a custom postHeight
        this("drop", "racing", "treadless", "razor", 19, 20, postHeight);
    }//end constructor

    // Parameterized constructor
    public RoadBike(String handleBars, String frame, String tyres, String seatType, int
numGears,
            int tyreWidth, int postHeight) {
        super(handleBars, frame, tyres, seatType, numGears);
        this.tyreWidth = tyreWidth;
        this.postHeight = postHeight;
    }//end constructor

    // Method to print the bike description
    public void printDescription() {
        super.printDescription();
        System.out.println("This RoadBike has " + this.tyreWidth + "mm tyres and a post
height of " + this.postHeight + " inches.");
    }//end method printDescription

    public static void main(String[] args) {
        RoadBike defaultBike = new RoadBike();
        defaultBike.printDescription();
        RoadBike customPostHeightBike = new RoadBike(25);
        customPostHeightBike.printDescription();

        RoadBike customBike = new RoadBike("flat", "carbon", "slick", "aero", 18, 23, 24);
        customBike.printDescription();
    }//end method main

}//end class RoadBike
```