

```
1 package revision;
2
3 public class Car extends Vehicle{
4     private int fuelCapacity;
5     Engine engine;
6
7     Car(){
8
9     }
10
11     Car(int fuelCapacity, Engine engine, int speed, String color){
12         super(speed, color);
13         this.fuelCapacity = fuelCapacity;
14         this.engine = engine;
15     }
16
17     Car(Car origin){
18         this.fuelCapacity = origin.fuelCapacity;
19         this.engine = origin.engine;
20     }
21
22     public Car(int fuelCapacity, Engine engine) {
23         this.fuelCapacity = fuelCapacity;
24         this.engine = engine;
25     }
26
27     public void printFuelRemain(){
28         System.out.println("Fuel Remaining : "+ fuelCapacity); // TODO
29     }
30
31     @Override
32     public String toString() {
33         return super.toString() + "Car{" +
34             "fuelCapacity=" + fuelCapacity +
35             ", engine=" + engine +
36             '}';
37     }
38 }
39
```

```
1 package revision;
2
3 import java.util.*;
4
5 public class Boat extends Vehicle{
6     public static final int MAXSPEED = 60;
7     public List<Engine> engines = new ArrayList<Engine>(3);
8     private int numCrew;
9
10    Boat(){
11
12    }
13
14
15    public Boat(int speed, String color, List<Engine> engines, int numCrew) {
16        super(speed, color);
17        this.engines = engines;
18        this.numCrew = numCrew;
19    }
20
21    Boat(Boat original){
22        // for list use addAll method instead of for loop
23        super(original);
24        this.engines.addAll(original.engines);
25        this.numCrew = original.numCrew;
26    }
27
28    public void printCrewNumber(){
29        System.out.println("Crew num: " + numCrew);
30    }
31
32    @Override
33    public String toString() {
34        return super.toString() + "Boat{" +
35            "engines=" + engines +
36            ", numCrew=" + numCrew +
37            '}';
38    }
39 }
40
```

```
1 package revision;
2
3 import java.util.*;
4
5 public class Main {
6
7     public static void main(String[] args) {
8         Scanner scan = new Scanner(System.in);
9         System.out.print("Enter Car color: ");
10        String color = scan.nextLine();
11        System.out.print("Enter Car speed: ");
12        int speed = scan.nextInt();
13        System.out.print("Enter Car Fuel capacity: ");
14        int fuelCap = scan.nextInt();
15
16        Engine engine = new Engine(1600, 210);
17
18        Car c = new Car(fuelCap, engine, speed, color);
19
20        System.out.println("Car = " + c.toString());
21    }
22
23    public static void forExample(){
24        int[] array = {1,2,3,4,5};
25        // example for for-loop
26        for(int i=0; i<array.length; i++){
27            System.out.println(array[i]);
28            // or to add in array
29            array[i] = i;
30        }
31        //enhanced for-loops
32        for(int num : array){
33            System.out.println(num);
34        }
35    }
36 }
37
38
39 }
40
41
42
```

```
1 package revision;
2
3 public class Engine {
4
5     private int CC;
6     private int hp;
7
8     public float calcTorq(Engine engine){
9         return (this.CC * this.hp) * (5/3);
10    }
11
12    public int getCC() {
13        return CC;
14    }
15
16    public void setCC(int CC) {
17        this.CC = CC;
18    }
19
20    public int getHp() {
21        return hp;
22    }
23
24    public void setHp(int hp) {
25        this.hp = hp;
26    }
27
28    public Engine(int CC, int hp) {
29        this.CC = CC;
30        this.hp = hp;
31    }
32
33    @Override
34    public String toString() {
35        return super.toString() + "Engine{" +
36            "CC=" + CC +
37            ", hp=" + hp +
38            '}';
39    }
40 }
41
```

```
1 package revision;
2
3 public class Printer {
4     public static void vehiclePrinter(Vehicle v){
5         // use instanceof to compare objects and their types
6         // in generic methods question
7         if(v instanceof Boat){
8             ((Boat) v).printCrewNumber();
9         }
10        else if(v instanceof Car){
11            ((Car) v).printFuelRemain();
12        }
13        else{
14            System.out.println("Unsupported type of vehicle");
15        }
16    }
17 }
18
```

```
1 package revision;
2
3 public class Scooter extends Vehicle implements IRechargeable{
4     private int batterySize;
5
6     public Scooter(int batterySize) {
7         this.batterySize = batterySize;
8     }
9
10
11
12     public int getBatterySize() {
13         return batterySize;
14     }
15
16     public void setBatterySize(int batterySize) {
17         this.batterySize = batterySize;
18     }
19
20     @Override
21     public int calcRechargeTime(int rechargingRate, int batterySize) {
22         return rechargingRate * batterySize;
23     }
24 }
25
```

```
1 package revision;
2
3 public abstract class Vehicle {
4     private int speed;
5     protected String color;
6
7     public Vehicle() {
8     }
9
10    public Vehicle(Vehicle original){
11        this.speed = original.speed;
12        this.color = original.color;
13    }
14
15    public Vehicle(int speed, String color) {
16        this.speed = speed;
17        this.color = color;
18    }
19
20    @Override
21    public String toString() {
22        return "Vehicle{" +
23            "speed=" + speed +
24            ", color='" + color + '\'' +
25            '}';
26    }
27 }
28
```

```
1 package revision;
2
3 public interface IRechargeable {
4     int rechargingRate = 50;
5
6     int calcRechargeTime(int rechargingRate, int batterySize);
7 }
8
```