File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Car.java

```java
package revision;

public class Car extends Vehicle{
    private int fuelCapacity;
    Engine engine;

    Car(){

    }

    Car(int fuelCapacity, Engine engine, int speed, String color){
        super(speed, color);
        this.fuelCapacity = fuelCapacity;
        this.engine = engine;
    }

    Car(Car origin){
        this.fuelCapacity = origin.fuelCapacity;
        this.engine = origin.engine;
    }

    public Car(int fuelCapacity, Engine engine) {
        this.fuelCapacity = fuelCapacity;
        this.engine = engine;
    }

    public void printFuelRemain(){
        System.out.println("Fuel Remaining : "+ fuelCapacity); // TODO
    }

    @Override
    public String toString() {
        return super.toString() + "Car{" +
                "fuelCapacity=" + fuelCapacity +
                ", engine=" + engine +
                '}';
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Boat.java

```java
package revision;

import java.util.*;

public class Boat extends Vehicle{
    public static final int MAXSPEED = 60;
    public List<Engine> engines = new ArrayList<Engine>(3);
    private int numCrew;

    Boat(){

    }

    public Boat(List<Engine> engines, int numCrew) {
        this.engines = engines;
        this.numCrew = numCrew;
    }

    Boat(Boat original){
        // for list use addAll method instead of for loop
        this.engines.addAll(original.engines);
        this.numCrew = original.numCrew;
    }

    public void printCrewNumber(){
        System.out.println("Crew num: " + numCrew);
    }

    @Override
    public String toString() {
        return super.toString() + "Boat{" +
                "engines=" + engines +
                ", numCrew=" + numCrew +
                '}';
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Main.java

```java
package revision;

import java.util.*;

public class Main {

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter Car color: ");
        String color = scan.nextLine();
        System.out.print("Enter Car speed: ");
        int speed = scan.nextInt();
        System.out.print("Enter Car Fuel capacity: ");
        int fuelCap = scan.nextInt();

        Engine engine = new Engine(1600, 210);

        Car c = new Car(fuelCap, engine, speed, color);

        System.out.println("Car = " + c.toString());

    }

    public static void forExample(){
        int[] array = {1,2,3,4,5};
        // example for for-loop
        for(int i=0; i<array.length; i++){
            System.out.println(array[i]);
            // or to add in array
//            array[i] = i;
        }
        //enhanced for-loops
        for(int num : array){
            System.out.println(num);
        }
    }


}
```

```java
package revision;

public class Engine {

    private int CC;
    private int hp;

    public float calcTorq(Engine engine){
        return (this.CC * this.hp) * (5/3);
    }

    public int getCC() {
        return CC;
    }

    public void setCC(int CC) {
        this.CC = CC;
    }

    public int getHp() {
        return hp;
    }

    public void setHp(int hp) {
        this.hp = hp;
    }

    public Engine(int CC, int hp) {
        this.CC = CC;
        this.hp = hp;
    }

    @Override
    public String toString() {
        return super.toString() + "Engine{" +
                "CC=" + CC +
                ", hp=" + hp +
                '}';
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Printer.java

```java
package revision;

public class Printer {
    public static void vehiclePrinter(Vehicle v){
        // use instanceof to compare objects and their types
        // in generic methods question
        if(v instanceof Boat){
            ((Boat) v).printCrewNumber();
        }
        else if(v instanceof Car){
            ((Car) v).printFuelRemain();
        }
        else{
            System.out.println("Unsupported type of vehicle");
        }
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Scooter.java

```java
package revision;

public class Scooter extends Vehicle implements IRechargable{
    private int batterySize;

    public Scooter(int batterySize) {
        this.batterySize = batterySize;
    }



    public int getBatterySize() {
        return batterySize;
    }

    public void setBatterySize(int batterySize) {
        this.batterySize = batterySize;
    }

    @Override
    public int calcRechargeTime(int rechargingRate, int batterySize) {
        return rechargingRate * batterySize;
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\Vehicle.java

```java
package revision;

public abstract class Vehicle {
    private int speed;
    protected String color;

    public Vehicle() {
    }

    public Vehicle(Vehicle original){
        this.speed = original.speed;
        this.color = original.color;
    }

    public Vehicle(int speed, String color) {
        this.speed = speed;
        this.color = color;
    }

    @Override
    public String toString() {
        return "Vehicle{" +
                "speed=" + speed +
                ", color='" + color + '\'' +
                '}';
    }
}

```

File - K:\UNI\uni\S6\oop\oop-revision\code\exam2019\src\revision\IRechargable.java

```java
package revision;

public interface IRechargable {
    int rechargingRate = 50;

    int calcRechargeTime(int rechargingRate, int batterySize);
}

```