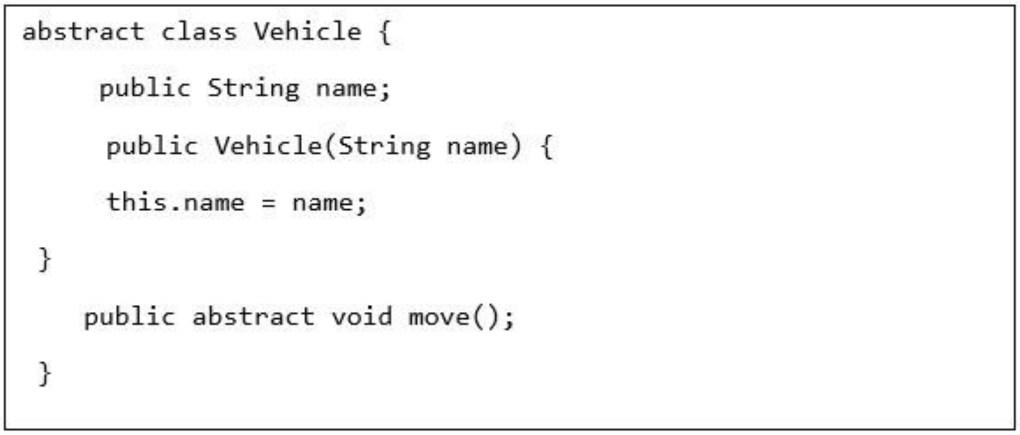
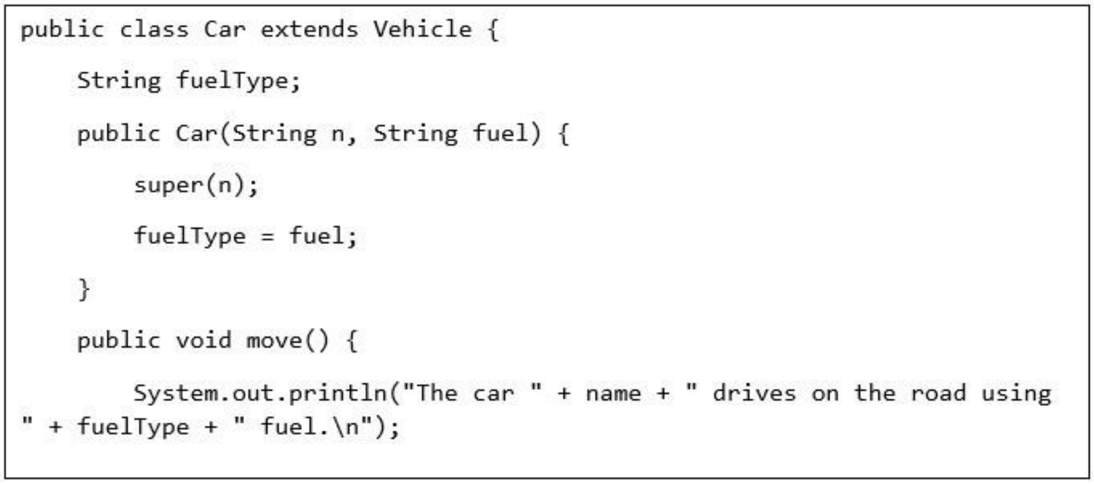
# LAB 3: POLYMORPHISM Lab Activities

1. The following program shows examples of abstract class and abstract method.

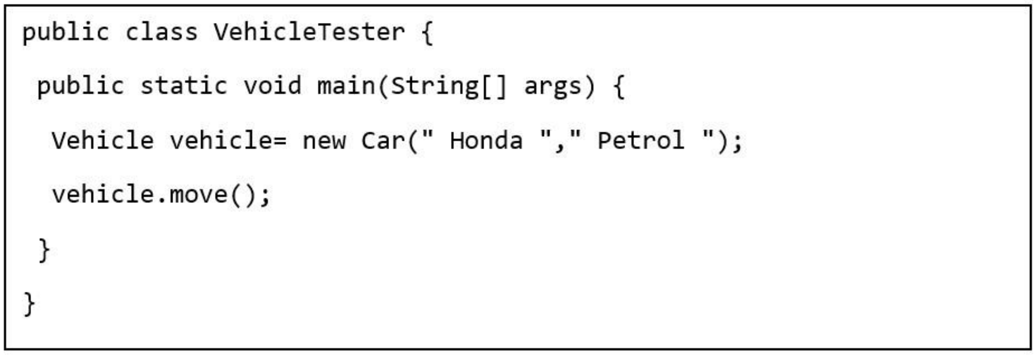
    a. Write, compile and run the following program of three classes. Give the output.



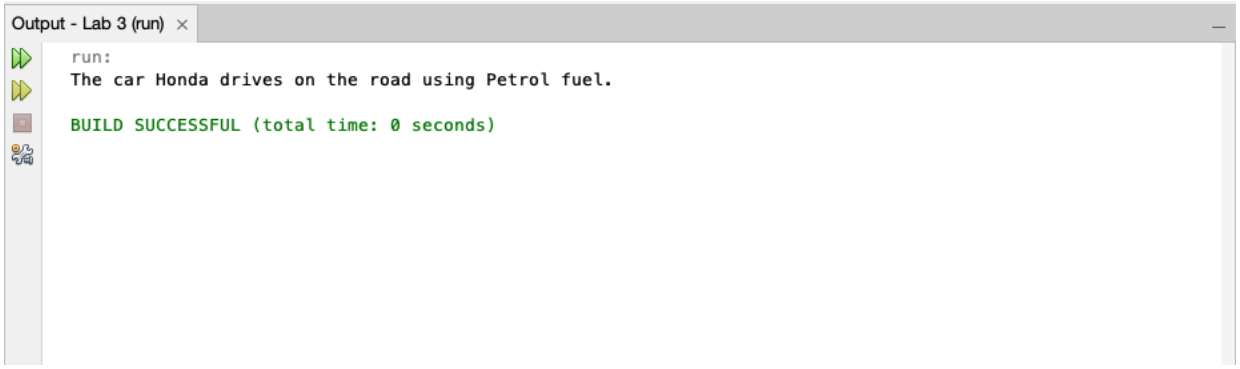
```
abstract class Vehicle {
     public String name;
      public Vehicle(String name) {
      this.name = name;
 }
    public abstract void move();
 }
```



```
public class Car extends Vehicle {
    String fuelType;
    public Car(String n, String fuel) {
        super(n);
        fuelType = fuel;
    }
    public void move() {
        System.out.println("The car " + name + " drives on the road using
" + fuelType + " fuel.\n");
```



```
public class VehicleTester {
 public static void main(String[] args) {
  Vehicle vehicle= new Car(" Honda "," Petrol ");
  vehicle.move();
 }
}
```



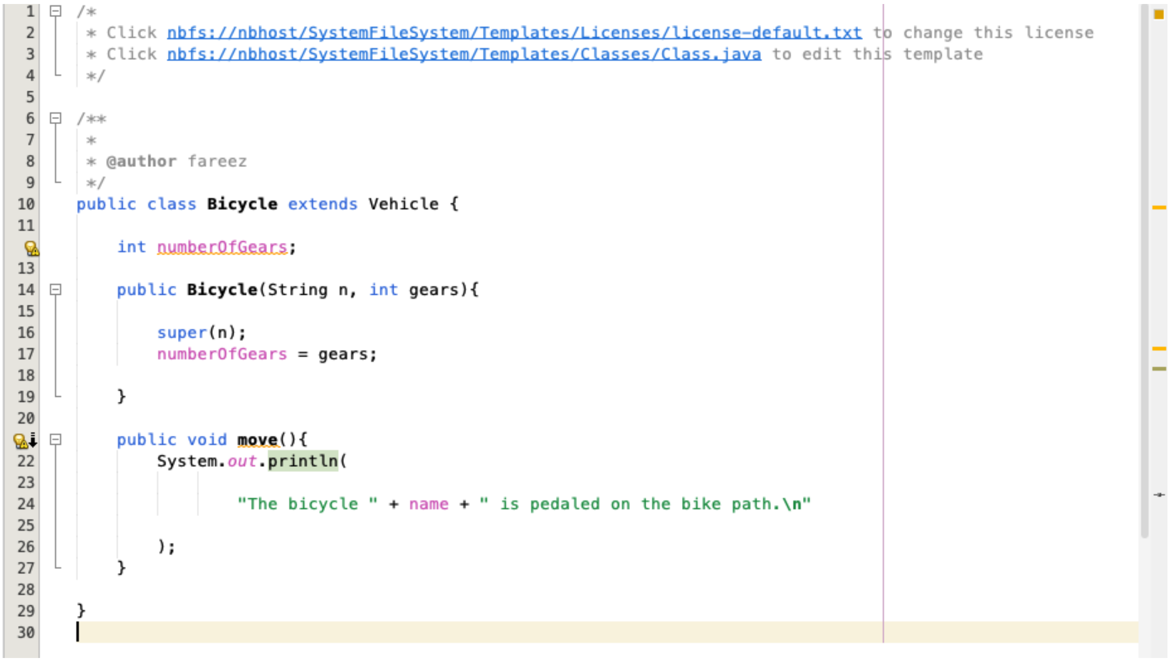
```
run:
The car Honda drives on the road using Petrol fuel.

BUILD SUCCESSFUL (total time: 0 seconds)
```

b. Add **Bicycle** class that inherits from abstract **Vehicle** class as following skeleton.

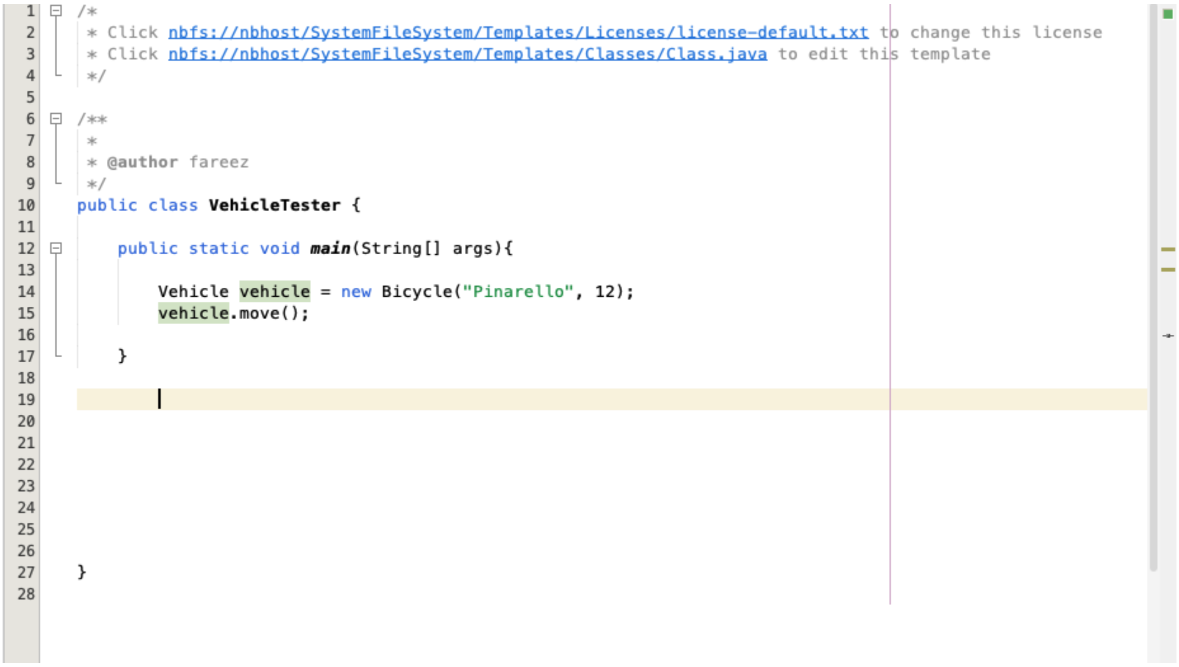
```
public class Bicycle extends Vehicle {

 int numberOfGears;

    public Bicycle(String n, int gears) {

        _____________________ // call superclass constructor

        numberOfGears = gears;

    }

public void move() {

  System.out.println("The bicycle " + name + " is pedaled on the 
bike path.\n");

 }

}
```

c. Instantiate a Bicyle object and assign to move() method in the VehicleTester class.

d. Compile and Run. Give the output.



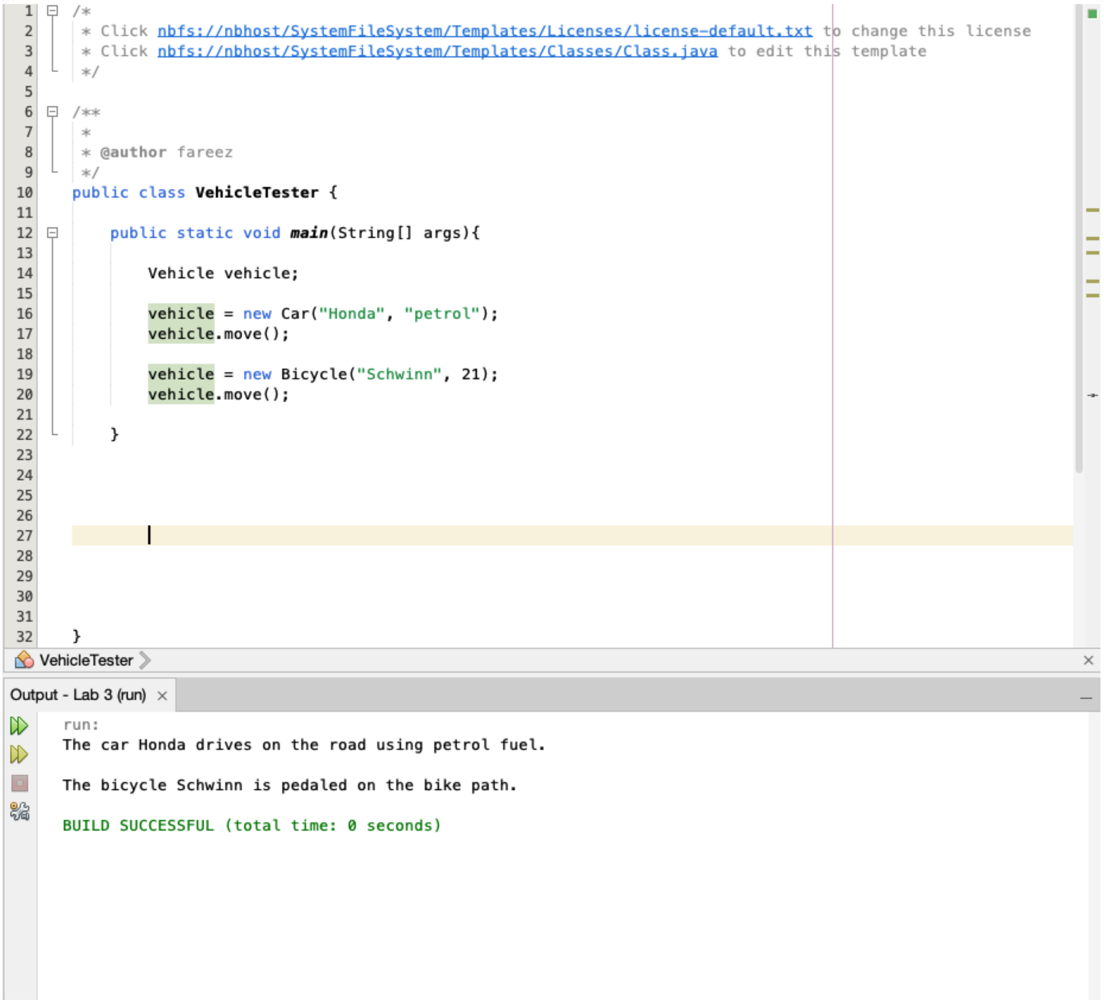
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class Bicycle extends Vehicle {

    int numberOfGears;

    public Bicycle(String n, int gears){

        super(n);
        numberOfGears = gears;

    }

    public void move(){
        System.out.println(

                "The bicycle " + name + " is pedaled on the bike path.\n"

        );
    }

}

```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class VehicleTester {

    public static void main(String[] args){

        Vehicle vehicle = new Bicycle("Pinarello", 12);
        vehicle.move();

    }









}

```

e. Rewrite your VehicleTester class with below program.

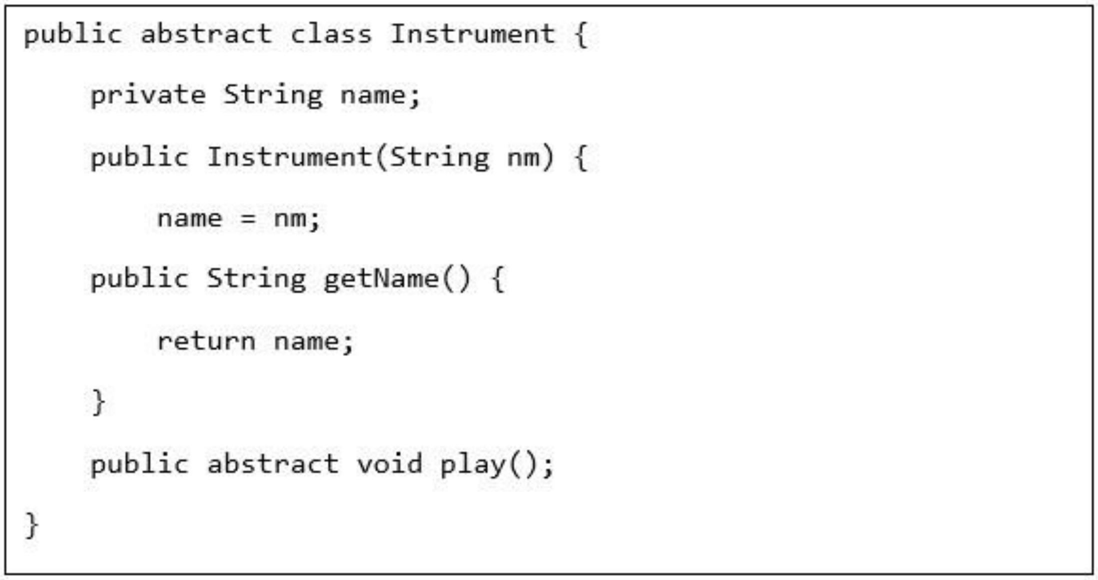
```
public class VehicleTester {

    public static void main(String[] args) {

        Vehicle vehicle;

        vehicle = new Car("Honda", "petrol");

        vehicle.move();

        vehicle = new Bicycle("Schwinn", 21);

        vehicle.move();

    }

}
```



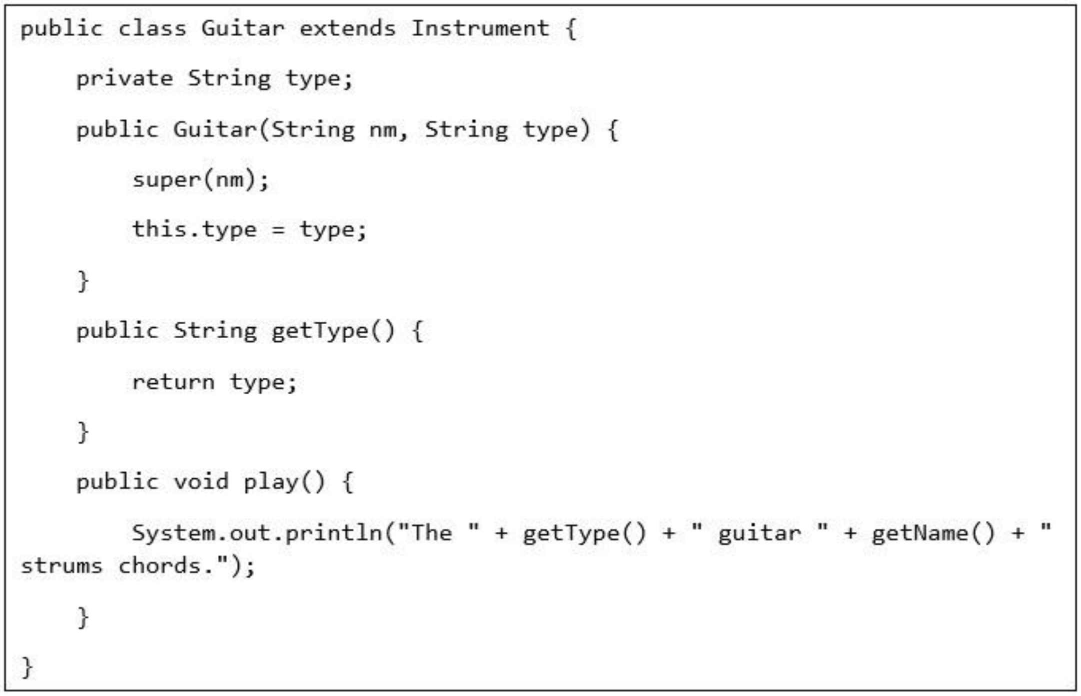
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class VehicleTester {

    public static void main(String[] args){

        Vehicle vehicle;

        vehicle = new Car("Honda", "petrol");
        vehicle.move();

        vehicle = new Bicycle("Schwinn", 21);
        vehicle.move();

    }









}
```

```
run:
The car Honda drives on the road using petrol fuel.

The bicycle Schwinn is pedaled on the bike path.

BUILD SUCCESSFUL (total time: 0 seconds)
```

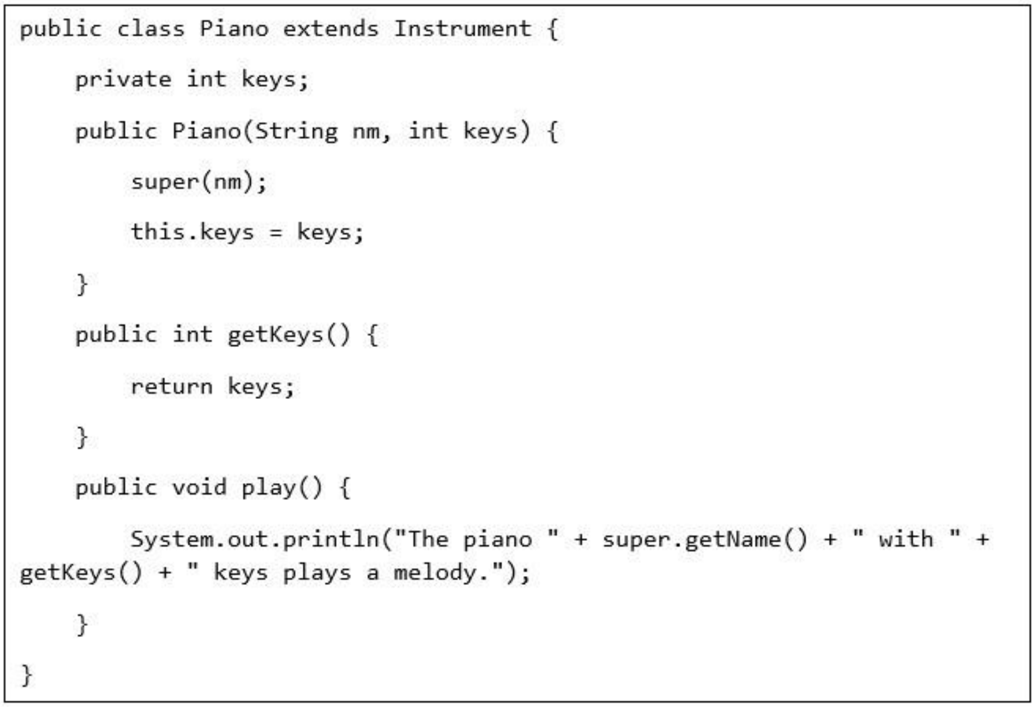
2. The following program demonstrates polymorphic behavior. Write, compile and run the following program. Give the output.



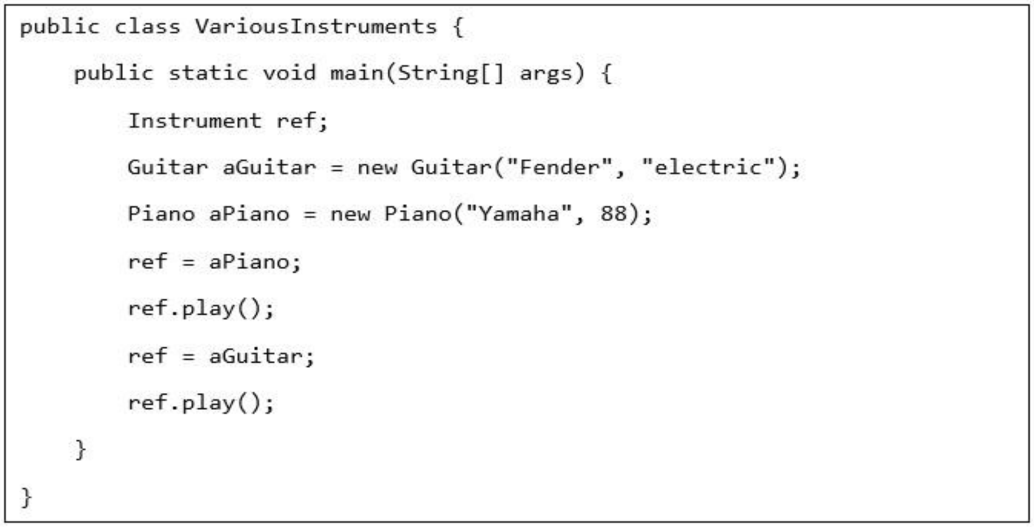
```
public abstract class Instrument {

    private String name;

    public Instrument(String nm) {

        name = nm;

    public String getName() {

        return name;

    }

    public abstract void play();

}
```



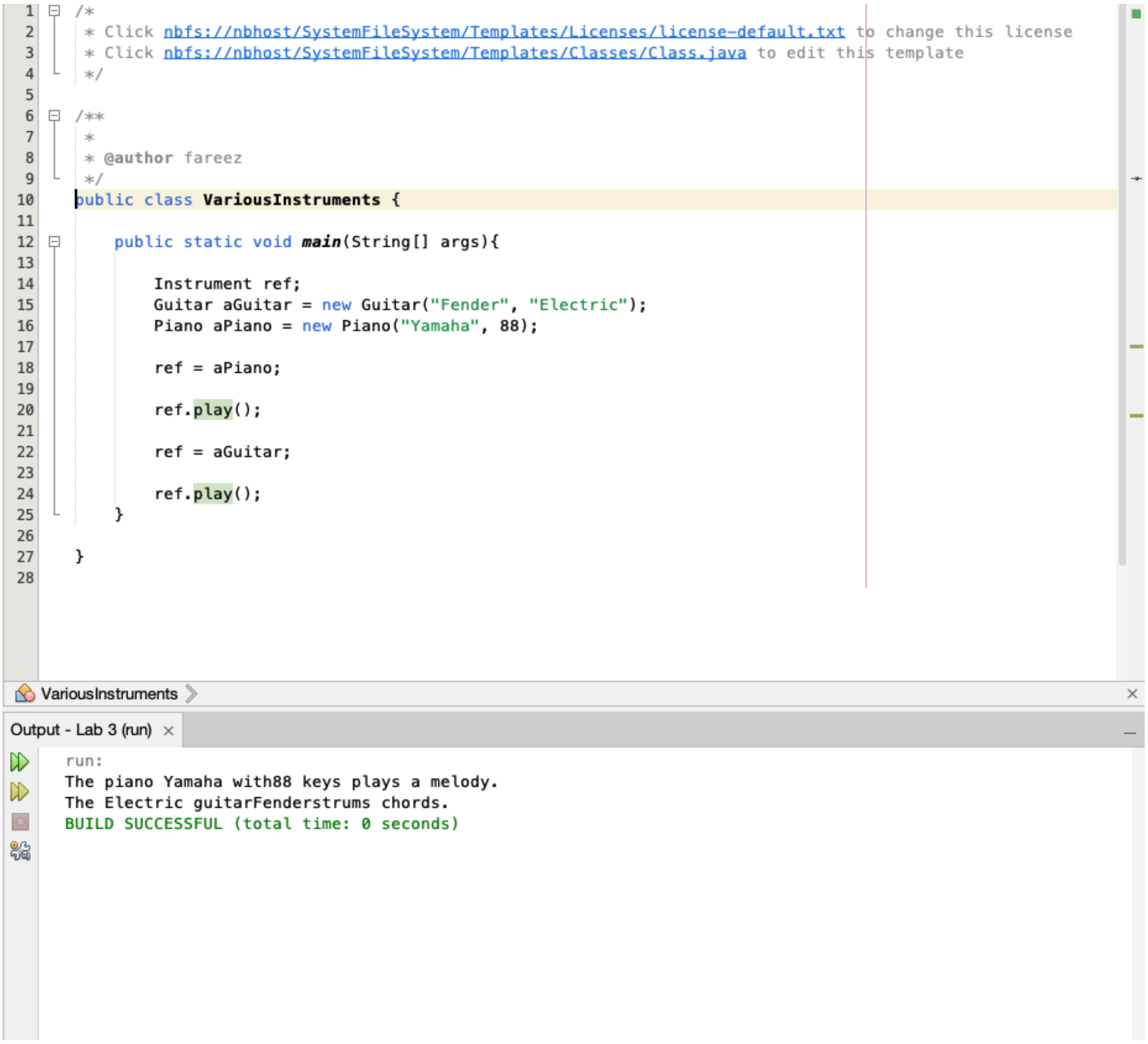
```
public class Guitar extends Instrument {

    private String type;

    public Guitar(String nm, String type) {

        super(nm);

        this.type = type;

    }

    public String getType() {

        return type;

    }

    public void play() {

        System.out.println("The " + getType() + " guitar " + getName() + "
strums chords.");

    }

}
```



```java
public class Piano extends Instrument {

    private int keys;

    public Piano(String nm, int keys) {

        super(nm);

        this.keys = keys;

    }

    public int getKeys() {

        return keys;

    }

    public void play() {

        System.out.println("The piano " + super.getName() + " with " +
getKeys() + " keys plays a melody.");

    }

}
```



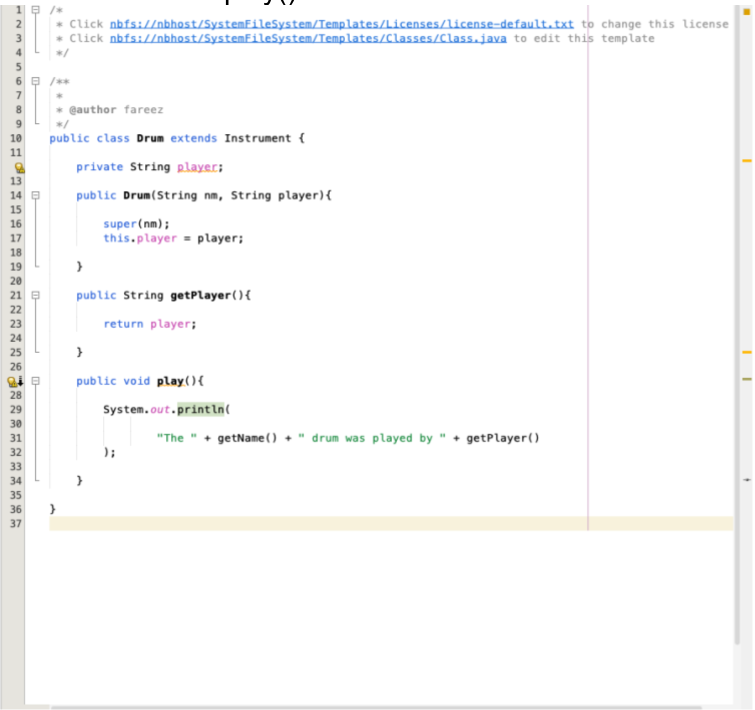
```java
public class VariousInstruments {

    public static void main(String[] args) {

        Instrument ref;

        Guitar aGuitar = new Guitar("Fender", "electric");

        Piano aPiano = new Piano("Yamaha", 88);

        ref = aPiano;

        ref.play();

        ref = aGuitar;

        ref.play();

    }

}
```



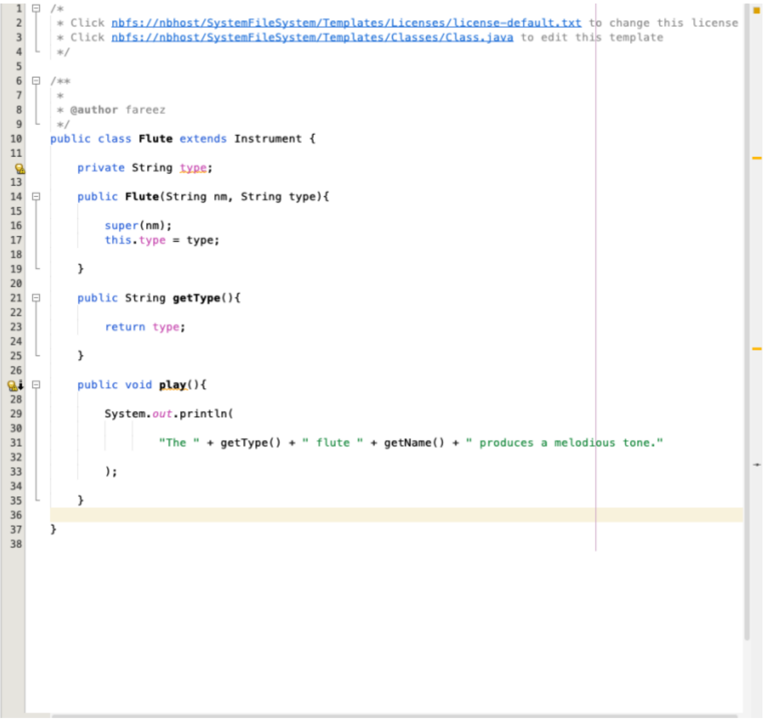
```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class VariousInstruments {

    public static void main(String[] args){

        Instrument ref;
        Guitar aGuitar = new Guitar("Fender", "Electric");
        Piano aPiano = new Piano("Yamaha", 88);

        ref = aPiano;

        ref.play();

        ref = aGuitar;

        ref.play();
    }

}
```

```
run:
The piano Yamaha with88 keys plays a melody.
The Electric guitarFenderstrums chords.
BUILD SUCCESSFUL (total time: 0 seconds)
```

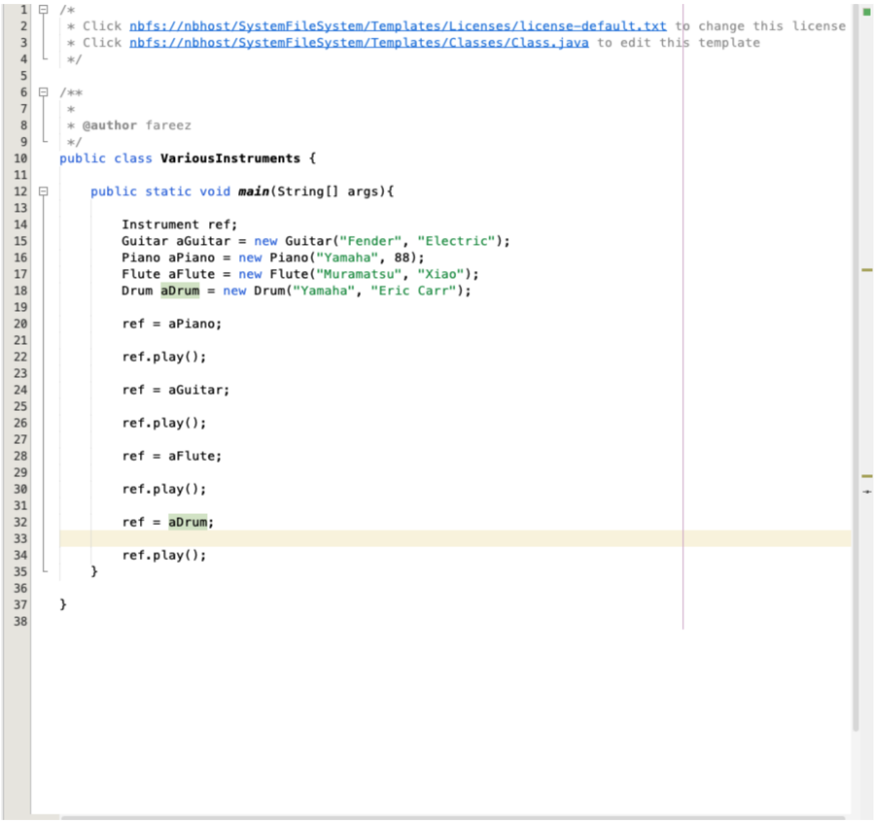
3. Extend the program in exercise 2 by adding two more subclasses of Instrument i.e. Flute and Drum where each has a method play().



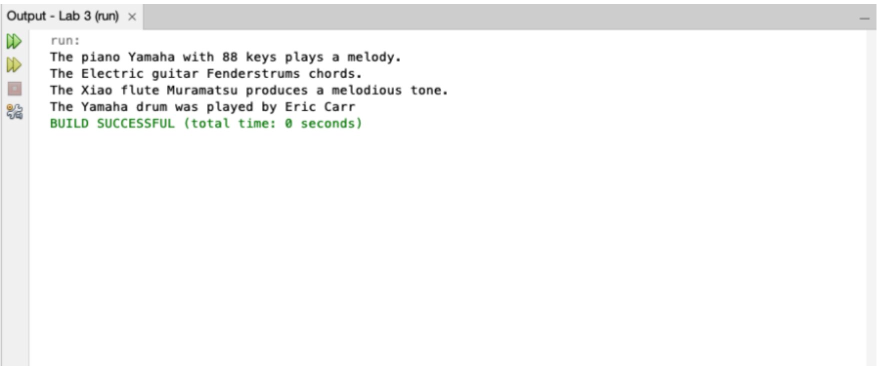
```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class Drum extends Instrument {

    private String player;

    public Drum(String nm, String player){

        super(nm);
        this.player = player;

    }

    public String getPlayer(){

        return player;

    }

    public void play(){

        System.out.println(

                "The " + getName() + " drum was played by " + getPlayer()
        );

    }

}
```



```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class Flute extends Instrument {

    private String type;

    public Flute(String nm, String type){

        super(nm);
        this.type = type;

    }

    public String getType(){

        return type;

    }

    public void play(){

        System.out.println(

                "The " + getType() + " flute " + getName() + " produces a melodious tone."

        );

    }

}
```

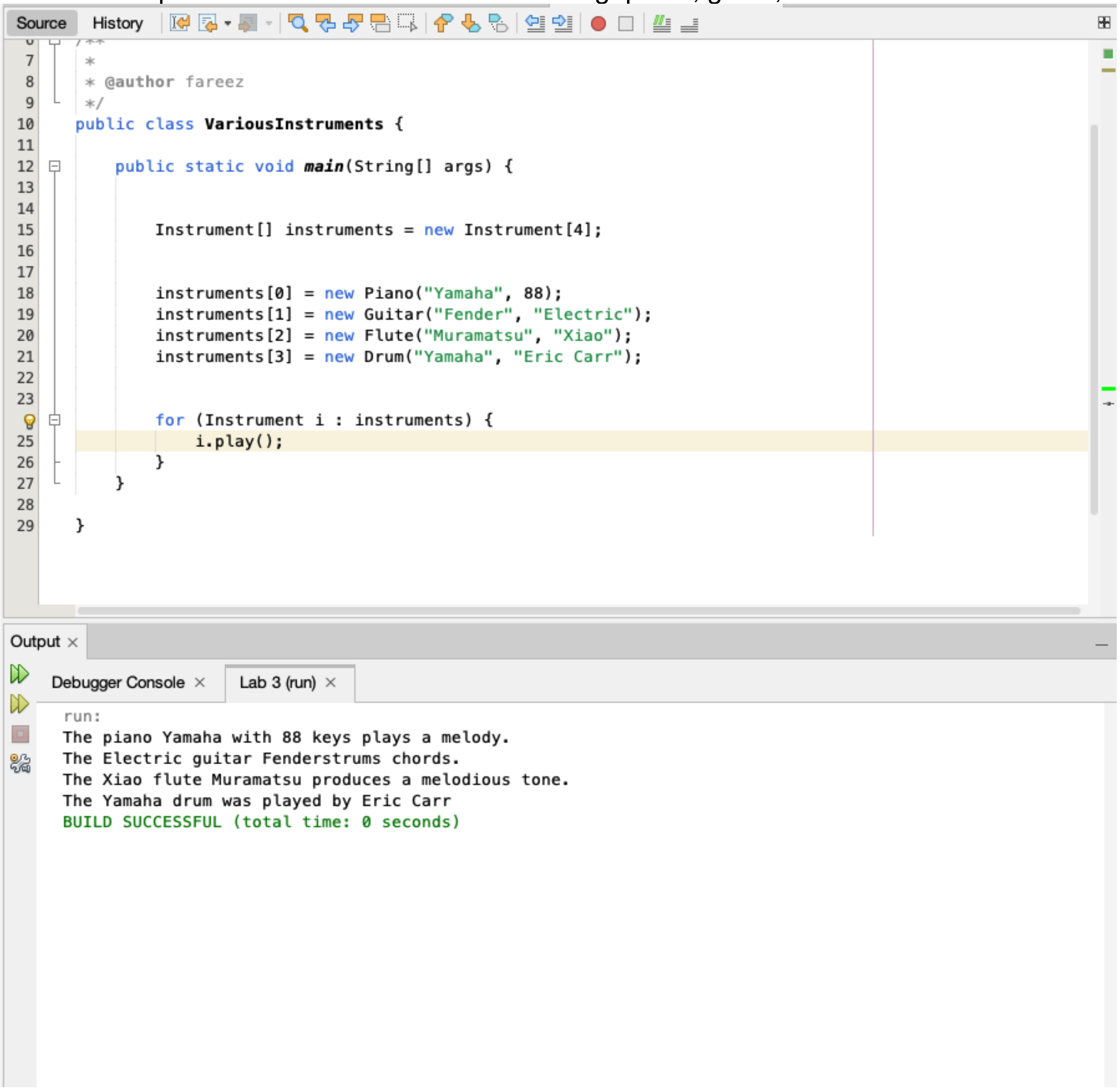


```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */

/**
 *
 * @author fareez
 */
public class VariousInstruments {

    public static void main(String[] args){

        Instrument ref;
        Guitar aGuitar = new Guitar("Fender", "Electric");
        Piano aPiano = new Piano("Yamaha", 88);
        Flute aFlute = new Flute("Muramatsu", "Xiao");
        Drum aDrum = new Drum("Yamaha", "Eric Carr");

        ref = aPiano;

        ref.play();

        ref = aGuitar;

        ref.play();

        ref = aFlute;

        ref.play();

        ref = aDrum;

        ref.play();
    }

}
```



```
run:
The piano Yamaha with 88 keys plays a melody.
The Electric guitar Fenderstrums chords.
The Xiao flute Muramatsu produces a melodious tone.
The Yamaha drum was played by Eric Carr
BUILD SUCCESSFUL (total time: 0 seconds)
```

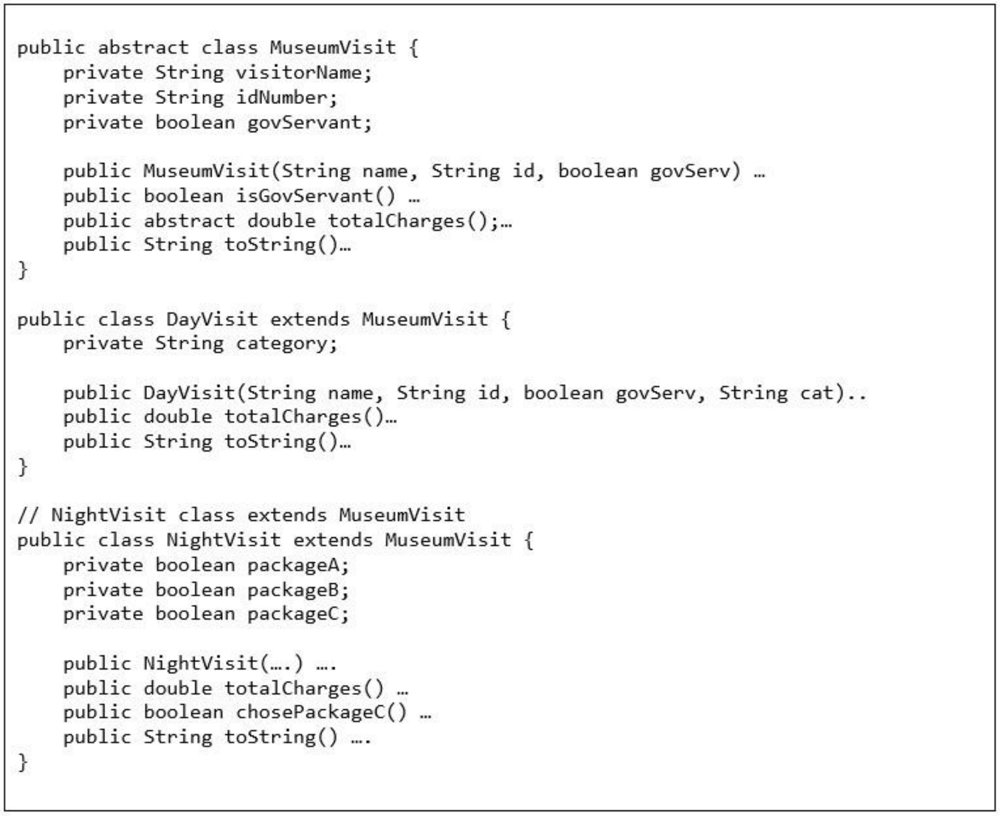
4. Modify the extended program created in Exercise 3 in this lab by creating an array of superclass objects and manipulate it by invoking the appropriate method for each subclass member to print the sound of each instrument eg. piano, guitar, flute and drum.



```java
/**
 *
 * @author fareez
 */
public class VariousInstruments {

    public static void main(String[] args) {


        Instrument[] instruments = new Instrument[4];


        instruments[0] = new Piano("Yamaha", 88);
        instruments[1] = new Guitar("Fender", "Electric");
        instruments[2] = new Flute("Muramatsu", "Xiao");
        instruments[3] = new Drum("Yamaha", "Eric Carr");


        for (Instrument i : instruments) {
            i.play();
        }
    }

}
```

```
run:
The piano Yamaha with 88 keys plays a melody.
The Electric guitar Fenderstrums chords.
The Xiao flute Muramatsu produces a melodious tone.
The Yamaha drum was played by Eric Carr
BUILD SUCCESSFUL (total time: 0 seconds)
```

# Postlab Exercise



```
public abstract class MuseumVisit {
    private String visitorName;
    private String idNumber;
    private boolean govServant;

    public MuseumVisit(String name, String id, boolean govServ) …
    public boolean isGovServant() …
    public abstract double totalCharges();…
    public String toString()…
}

public class DayVisit extends MuseumVisit {
    private String category;

    public DayVisit(String name, String id, boolean govServ, String cat)..
    public double totalCharges()…
    public String toString()…
}

// NightVisit class extends MuseumVisit
public class NightVisit extends MuseumVisit {
    private boolean packageA;
    private boolean packageB;
    private boolean packageC;

    public NightVisit(….) ….
    public double totalCharges() …
    public boolean chosePackageC() …
    public String toString() ….
}
```

From the above class outlines, the Museum is having a promotion to the visitor, one is to visit the Museum during a **normal** day time and the second is to visit the Museum during the **night** time. For the day time visit, there is no restriction to the visitor. But for the night time visit, it is only suit for adult and the visitors have three packages to choose with different prices as shown by Table below:

| Package | Price |
|---|---|
| A | 50 |
| B | 75 |
| C | 100 |

Visitors will be given 15% discount from the total charges if the visitor is a government servant. For day time visit, an adult will be charged RM25.00 per person and the children will be charged RM15.00 per head.

a. Write the code fragment for the abstract methods in both subclasses.
b. Write a TestMuseum class which have main program and apply the concept of **polymorphism** to:
    1) Declare an array of objects to store 20 data information. Get input from user regarding all the information required. The number of data to be stored and information on each visitor is given by the user.
    2) Calculate and display the number of visitors for each category (day and night visit) and the total income for the museum.
    3) Find and display all visitor details that choose package C for the night visit promotion.

```
public abstract class MuseumVisit {
   private String visitorName;
   private String idNumber;
   private boolean govServant;

   public MuseumVisit(String name, String id, boolean govServ) {
      this.visitorName = name;
      this.idNumber = id;
      this.govServant = govServ;
   }

   public boolean isGovServant() {
      return govServant;
   }

   public String getVisitorName() {
      return visitorName;
   }

   public String getIdNumber() {
      return idNumber;
   }

   public abstract double totalCharges();

   public String toString() {
      return "Visitor Name: " + visitorName +
            "\nID Number: " + idNumber +
            "\nGovernment Servant: " + govServant;
   }
}
```

```
public class DayVisit extends MuseumVisit {

    private String category;

    public DayVisit(String name, String id, boolean govServ, String cat) {
        super(name, id, govServ);
        category = cat;
    }

    public double totalCharges() {
        double charge = 0;
        if (category.equalsIgnoreCase("Adult")) {
            charge = 25.0;
        } else if (category.equalsIgnoreCase("Child")) {
            charge = 15.0;
        }

        if (isGovServant()) {
            charge = charge * 0.85;
        }
        return charge;
    }

    public String toString() {
        return "Day Visit - Name: " + getVisitorName() + ", ID: " + getIdNumber() +
                ", Category: " + category + ", Charges: RM" + totalCharges();
    }
}
```

```
public class NightVisit extends MuseumVisit {

    private boolean packageA;
    private boolean packageB;
    private boolean packageC;

    public NightVisit(String name, String id, boolean govServ, boolean a, boolean b, boolean c) {
        super(name, id, govServ);
        packageA = a;
        packageB = b;
        packageC = c;
    }

    public boolean chosePackageC() {
        return packageC;
    }

    public double totalCharges() {
        double charge = 0;
        if (packageA) {
            charge = 50.0;
        } else if (packageB) {
            charge = 75.0;
        } else if (packageC) {
            charge = 100.0;
        }

        if (isGovServant()) {
            charge = charge * 0.85;
        }
        return charge;
    }

    public String toString() {
        String selectedPackage = "";
        if (packageA) selectedPackage = "Package A";
        else if (packageB) selectedPackage = "Package B";
        else if (packageC) selectedPackage = "Package C";

        return "Night Visit - Name: " + getVisitorName() + ", ID: " + getIdNumber() +
                ", Selected: " + selectedPackage + ", Charges: RM" + totalCharges();
    }
}
```

```
import java.util.Scanner;

public class TestMuseum {
  public static void main(String[] args) {
    Scanner input = new Scanner(System.in);

    MuseumVisit[] visitors = new MuseumVisit[20];
    int numVisitors = 0;

    int numDayCount = 0;
    int numNightCount = 0;
    double totalIncome = 0;

    System.out.print("Enter number of visitors to input: ");
    numVisitors = input.nextInt();
    input.nextLine(); // consume newline

    for (int i = 0; i < numVisitors; i++) {
      System.out.println("\nVisitor " + (i + 1));
      System.out.print("Enter visitor name: ");
      String name = input.nextLine();
      System.out.print("Enter ID number: ");
      String id = input.nextLine();
      System.out.print("Is the visitor a government servant? (true/false): ");
      boolean govServ = input.nextBoolean();
      input.nextLine(); // consume newline

      System.out.print("Day or Night visit? (D/N): ");
      String visitType = input.nextLine();

      if (visitType.equalsIgnoreCase("D")) {
        System.out.print("Enter category (Adult/Child): ");
        String category = input.nextLine();
        visitors[i] = new DayVisit(name, id, govServ, category);
        numDayCount++;
      } else if (visitType.equalsIgnoreCase("N")) {
        System.out.print("Choose package (only one true): ");
        System.out.print("Package A (true/false): ");
        boolean packA = input.nextBoolean();
        System.out.print("Package B (true/false): ");
        boolean packB = input.nextBoolean();
        System.out.print("Package C (true/false): ");
        boolean packC = input.nextBoolean();
        input.nextLine(); // consume newline

        visitors[i] = new NightVisit(name, id, govServ, packA, packB, packC);
        numNightCount++;
      }
```

```
            totalIncome += visitors[i].totalCharges();
        }

        System.out.println("\n\nSummary:");
        System.out.println("Day visit total visitors: " + numDayCount);
        System.out.println("Night visit total visitors: " + numNightCount);
        System.out.printf("Total income: RM%.2f\n", totalIncome);

        System.out.println("\nVisitors who chose Package C:");
        for (int i = 0; i < numVisitors; i++) {
            if (visitors[i] instanceof NightVisit) {
                NightVisit nv = (NightVisit) visitors[i];
                if (nv.chosePackageC()) {
                    System.out.println(nv.toString());
                }
            }
        }
    }
}
```

```
run:
Enter number of visitors to input: 5

Visitor 1
Enter visitor name: Abu
Enter ID number: 1234
Is the visitor a government servant? (true/false): true
Day or Night visit? (D/N): D
Enter category (Adult/Child): Adult

Visitor 2
Enter visitor name: Lisa
Enter ID number: 4124
Is the visitor a government servant? (true/false): false
Day or Night visit? (D/N): N
Choose package (only one true): Package A (true/false): false
Package B (true/false): false
Package C (true/false): true

Visitor 3
Enter visitor name: ubai
Enter ID number: 1234
Is the visitor a government servant? (true/false): true
Day or Night visit? (D/N): D
Enter category (Adult/Child): Child

Visitor 4
Enter visitor name: Shila
Enter ID number: 2412
Is the visitor a government servant? (true/false): false
Day or Night visit? (D/N): N
Choose package (only one true): Package A (true/false): true
Package B (true/false): false
Package C (true/false): false

Visitor 5
Enter visitor name: Ben Stiller
Enter ID number: 1344
Is the visitor a government servant? (true/false): true
Day or Night visit? (D/N): D
Enter category (Adult/Child): Adult
```

```
Summary:
Day visit total visitors: 3
Night visit total visitors: 2
Total income: RM205.25

Visitors who chose Package C:
Night Visit - Name: Lisa , ID: 4124, Selected: Package C, Charges: RM100.0
BUILD SUCCESSFUL (total time: 2 minutes 5 seconds)
```