

LAB CAT-II

Time: 10.45 AM-12.15 PM

marks)

```
import java.io.*;
import java.lang.*;
import java.util.*;

class MusicalComposition
{
    String music_title;
    String music_composer;
    String music_yearwritten;

    MusicalComposition()
    {
        this.music_title = "";
        this.music_composer = "";
        this.music_yearwritten = "";
    }

    MusicalComposition(String title, String composer, String yearwritten)
    {
        this.music_title = title;
        this.music_composer = composer;
        this.music_yearwritten = yearwritten;
    }

    void display()
    {
        System.out.println("\n\tClass : MusicalComposition");
        System.out.println("\t-----");
        System.out.println("TITLE           :   " + this.music_title);
        System.out.println("COMPOSER        :   " + this.music_composer);
        System.out.println("YEAR WRITTEN    :   " + this.music_yearwritten);
        System.out.println("-----");
    }
}
```

```

}

class NationalAnthem extends MusicalComposition
{
    String music_title;
    String music_composer;
    String music_yearwritten;
    String anthemsnation;

    NationalAnthem()
    {
        this.music_title = "";
        this.music_composer = "";
        this.music_yearwritten = "";
        this.anthemsnation = "";
    }

    NationalAnthem(MusicalComposition obj,String nation)
    {
        this.music_title = obj.music_title;
        this.music_composer = obj.music_composer;
        this.music_yearwritten = obj.music_yearwritten;
        this.anthemsnation = nation;
    }

    void display()
    {
        System.out.println("\n\t Class : NationalAnthem");
        System.out.println("\t-----");
        System.out.println("TITLE           :   " + this.music_title);
        System.out.println("COMPOSER           :   " + this.music_composer);
        System.out.println("YEAR WRITTEN       :   " + this.music_yearwritten);
        System.out.println("ANTHEMS NATION     :   " + this.anthemsnation);
        System.out.println("-----");
    }
}

public class MusicDemo {

    public static void main(String[] Nithish) {

        System.out.print("\tName      : Nithish G \n\tReg No. : 19BCS0012\n");
        System.out.print("-----
\n\n");

        MusicalComposition object1 = new MusicalComposition("Jana Gana Mana",
        "Rabindranath Tagore", "1911");
        NationalAnthem object2 = new NationalAnthem(object1, "India");

        object1.display();

        object2.display();
    }
}

```

OUTPUT

The screenshot displays an IDE with two main panels. The left panel, titled 'Console', shows the output of a Java application named 'MusicDemo'. The output is as follows:

```
<terminated> MusicDemo [Java Application] C:\Program Files\Java\jdk-1.8.0_101\bin\java.exe
Name      : Nithish G
Reg No.   : 19BCS0012
-----
Class : MusicalComposition
-----
TITLE      : Jana Gana Mana
COMPOSER   : Rabindranath Tagore
YEAR WRITTEN : 1911
-----
Class : NationalAnthem
-----
TITLE      : Jana Gana Mana
COMPOSER   : Rabindranath Tagore
YEAR WRITTEN : 1911
ANTHEMS NATION : India
-----
```

The right panel shows the source code of the application, 'Arun.java', located in the 'src' directory. The code defines a 'NationalAnthem' class and a 'MusicDemo' class. The 'NationalAnthem' class has a constructor that initializes attributes and a 'display()' method that prints the attributes. The 'MusicDemo' class has a 'main' method that creates a 'NationalAnthem' object and calls its 'display()' method.

```
54
55 NationalAnthem(MusicalComposition mc) {
56     {
57         this.music_title = mc.music_title;
58         this.music_composer = mc.music_composer;
59         this.music_yearwritten = mc.music_yearwritten;
60         this.anthemsnation = mc.anthemsnation;
61     }
62
63 void display()
64 {
65     System.out.println("Title: " + music_title);
66     System.out.println("Composer: " + music_composer);
67     System.out.println("Year Written: " + music_yearwritten);
68     System.out.println("Anthem Nation: " + anthemsnation);
69     System.out.println("-----");
70     System.out.println("Class: " + getClass().getSimpleName());
71     System.out.println("-----");
72 }
73
74 }
75
76 public class MusicDemo {
77
78     public static void main(String[] args) {
79
80         NationalAnthem anthem = new NationalAnthem(new MusicalComposition("Jana Gana Mana", "Rabindranath Tagore", 1911, "India"));
81         System.out.print("\n");
82         System.out.print("-----");
83         anthem.display();
84     }
85 }
```

The bottom status bar shows the current line of code: 'Dwelling D - RV.display(Vehicle, Dwelling)'.

2. Create the classes named Vehicle and Dwelling. The Vehicle class contains data members such as a vehicle identification number, make, and number of miles the vehicle can travel on a gallon of gas. The Dwelling class contains data members such as number of bedrooms and area in square feet. Create a recreational vehicle class RV inherits from both Vehicle and Dwelling. RV is a vehicle that “is a” Vehicle (you drive it; it runs on gas), but also “is a” Dwelling (you sleep, cook, and live in it, at least during road trips). Create main function in RVDemo class to receive five arguments required for the RV class constructor; in which, three are passed to the Vehicle constructor and the other two are passed to the Dwelling constructor. Display the received values and the rental amount using RV class display()function.

(10 marks)

Source code:

```
interface V
{

}

interface H
{

}

class Vehicle implements V
{
    String Vehicle_ID;
    String make;
    double milespergal;
}

class Dwelling implements H
{
    int nofbedroom;
    double areasqft;
}

class RV implements V, H
{
    RV()
    {
```

```

    }

    RV(String Vehicle_ID, String make, double milespergal, int nofbedroom,
double areasqft)
    {
        Vehicle V = new Vehicle();
        V.Vehicle_ID = Vehicle_ID;
        V.make = make;
        V.milespergal = milespergal;

        Dwelling D = new Dwelling();
        D.nofbedroom = nofbedroom;
        D.areasqft = areasqft;

        RV obj = new RV();
        obj.display(V,D);
    }

    void display(Vehicle V, Dwelling D)
    {
        System.out.println("          From Class RV ---> Display()\n");
        System.out.println("VEHICLE ID          :    " + V.Vehicle_ID);
        System.out.println("VEHICLE MAKE          :    " + V.make);
        System.out.println("MILES PER GALON GAS   :    " + V.milespergal);
        System.out.println("NO.OF BEDROOMS        :    " + D.nofbedroom);
        System.out.println("AREA IN SQ.FEET       :    " + D.areasqft+"
sq.ft");
        System.out.println("RENT                  :    " + "Rs.
"+(D.nofbedroom*V.milespergal*10));
        System.out.println("-----");
    }
}

public class RVDemo {

    public static void main(String[] NITHISH) {

        System.out.print("\tName      : Nithish G \n\tReg No. : 19BCS0012\n");
        System.out.print("-----
\n\n");
    }
}

```

```

    RV object = new RV("A712", "AUDI", 15, 5, 250);
}

}

```

Output

The screenshot shows an IDE with two main panels. The left panel displays the output of a Java application, and the right panel shows the source code of the application.

Output Panel:

```

<terminated> RVDemo [Java Application] C:\Program Files\Java\jdk1.7
Name      : Nithish G
Reg No.   : 19BCS0012
-----
From Class RV ---> Display()
VEHICLE ID      : A712
VEHICLE MAKE    : AUDI
MILES PER GALON GAS : 15.0
NO.OF BEDROOMS  : 5
AREA IN SQ.FEET : 250.0 sq.ft
RENT            : Rs. 750.0
-----

```

Source Code Panel:

```

39 D.noBedroom
40 D.areaSqft =
41
42 RV obj = new
43 obj.display()
44 }
45
46 void display(Veh
47 {
48     System.out.p
49     System.out.p
50     System.out.p
51     System.out.p
52     System.out.p
53     System.out.p
54     System.out.p
55     System.out.p
56 }
57 }
58 public class RVDemo
59
60 public static vo
61
62     System.out.p
63     System.out.p
64     RV object =
65 }
66
67 }
68

```

The bottom status bar indicates the current file is `String[] NITHISH - RVDemo.main(String[] args)`.