









/* Priyanshu Jain */

Q 1) WAP to show that there always exists single copy of static member of a class in memory:



Main.java



Run

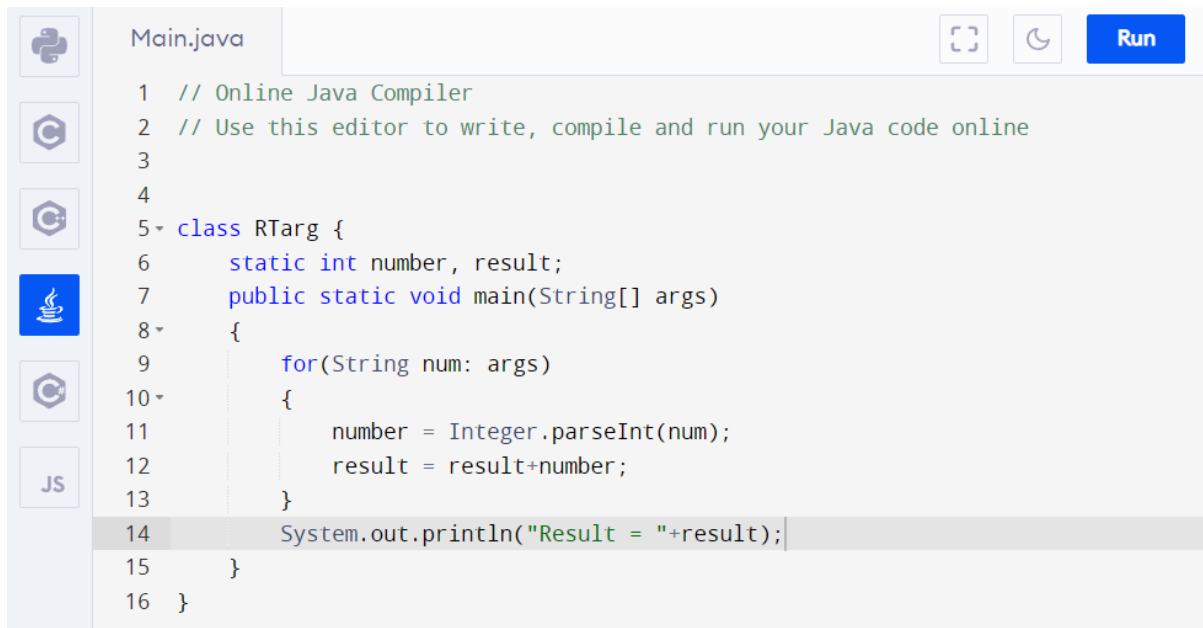
```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4
5 class MyClass {
6     static int staticDM = 0;
7     int nonStaticDM = 0;
8     public static void main(String[] args)
9     {
10         MyClass obj1 = new MyClass();
11         MyClass obj2 = new MyClass();
12
13         System.out.println(++obj1.nonStaticDM);
14         System.out.println(obj2.staticDM);
15         System.out.println(++obj1.staticDM);
16         System.out.println(obj2.staticDM);
17         System.out.println(MyClass.staticDM);
18         System.out.println(staticDM);
19     }
20 }
```

Output

Clear

```
java -cp /tmp/eg5XHU1eII MyClass
1
0
1
1
1
1
```

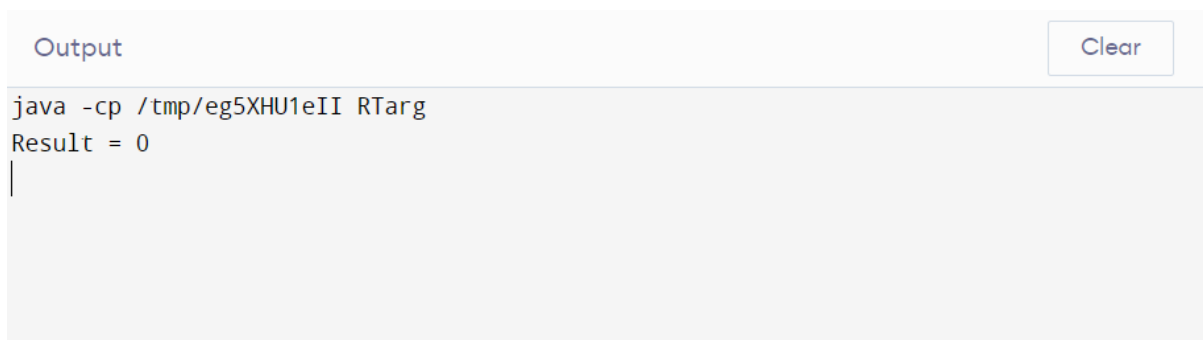
Q 2) WAP using for each loop to add numbers passed as command line argument:



The screenshot shows an online Java IDE with a file named 'Main.java'. The code is as follows:

```
1 // Online Java Compiler
2 // Use this editor to write, compile and run your Java code online
3
4
5 class RTarg {
6     static int number, result;
7     public static void main(String[] args)
8     {
9         for(String num: args)
10        {
11            number = Integer.parseInt(num);
12            result = result+number;
13        }
14        System.out.println("Result = "+result);
15    }
16 }
```

On the left side, there is a sidebar with icons for Python, C, C++, Java (selected), and JavaScript. On the right side of the editor, there are icons for a full-screen view, a moon theme, and a blue 'Run' button.



The screenshot shows the 'Output' window of the IDE. It contains the command used to run the program and its output:

```
java -cp /tmp/eg5XHU1eII RTarg
Result = 0
```

There is a 'Clear' button in the top right corner of the output window.

Q 3) WAP that accepts length and width of a triangle using scanner class and calculate its area:





Main.javaAreaOfRectangle.java+

```
1 import java.util.Scanner;
2
3 public class AreaOfRectangle {
4
5     public static void main(String[] args)
6     {
7         Scanner sc = new Scanner(System.in);
8         System.out.println("Enter the length of rectangle: ");
9         double length = sc.nextDouble();
10        System.out.println("Enter the width of rectangle: ");
11        double width = sc.nextDouble();
12        double area = length*width;
13        System.out.println("Area of Rectangle : " +area);
14    }
15 }
```

Ln: 3, Col: 29

RunShare

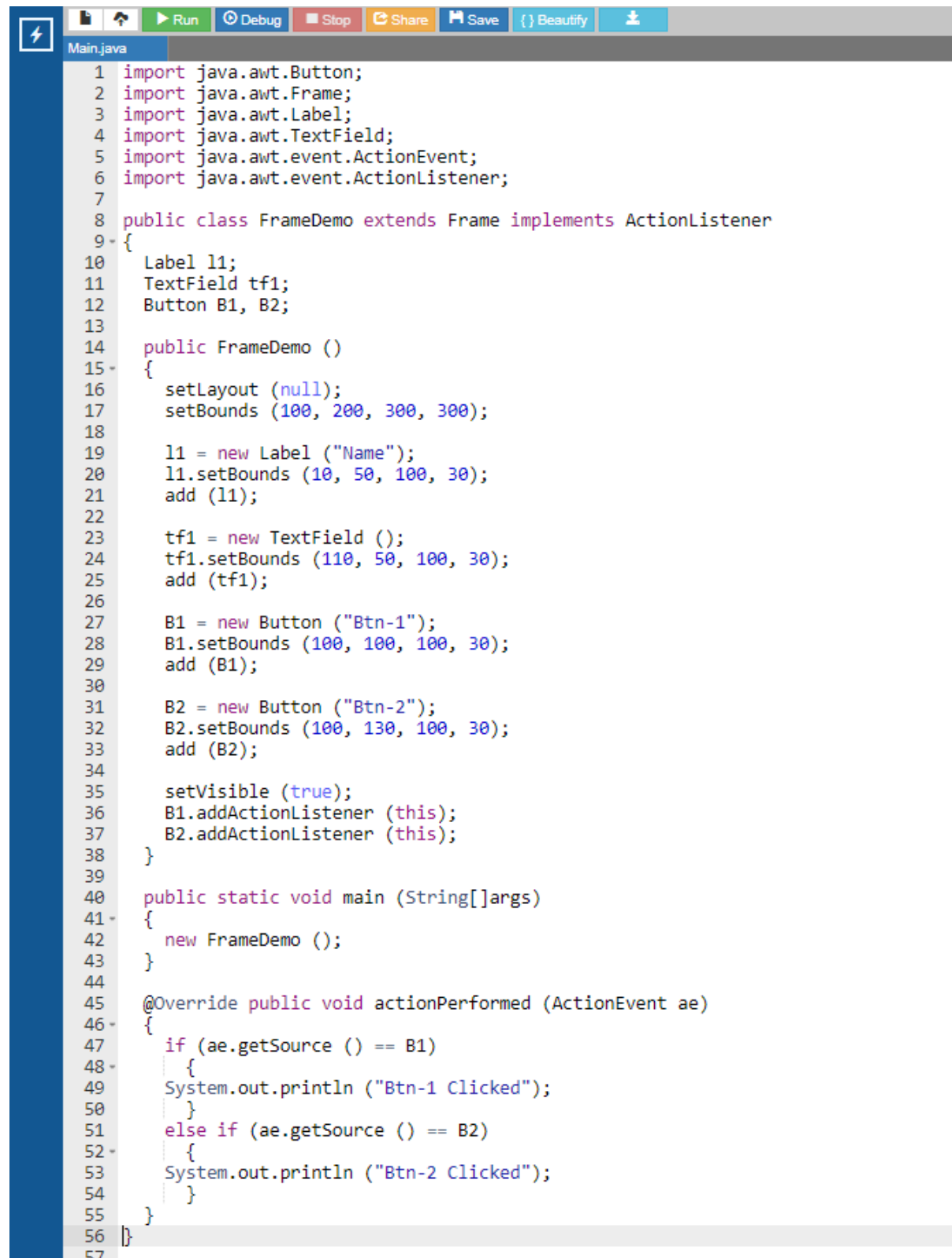
Command Line Arguments



Enter the length of rectangle:
4
Enter the width of rectangle:
5
Area of Rectangle : 20.0

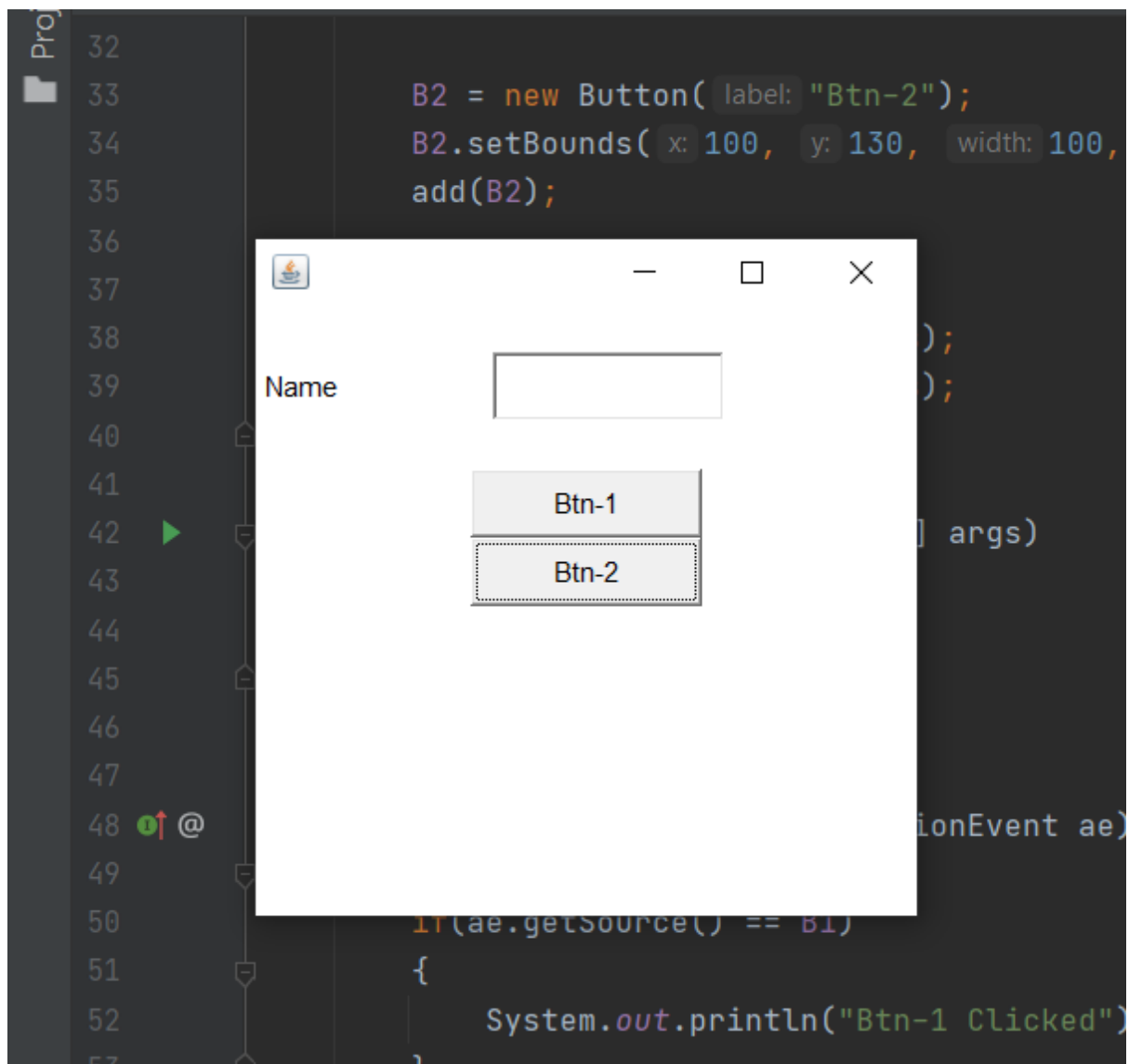
** Process exited - Return Code: 0 **

Q 5) Create a frame , add two buttons into it, and identify the buttons when they are clicked:

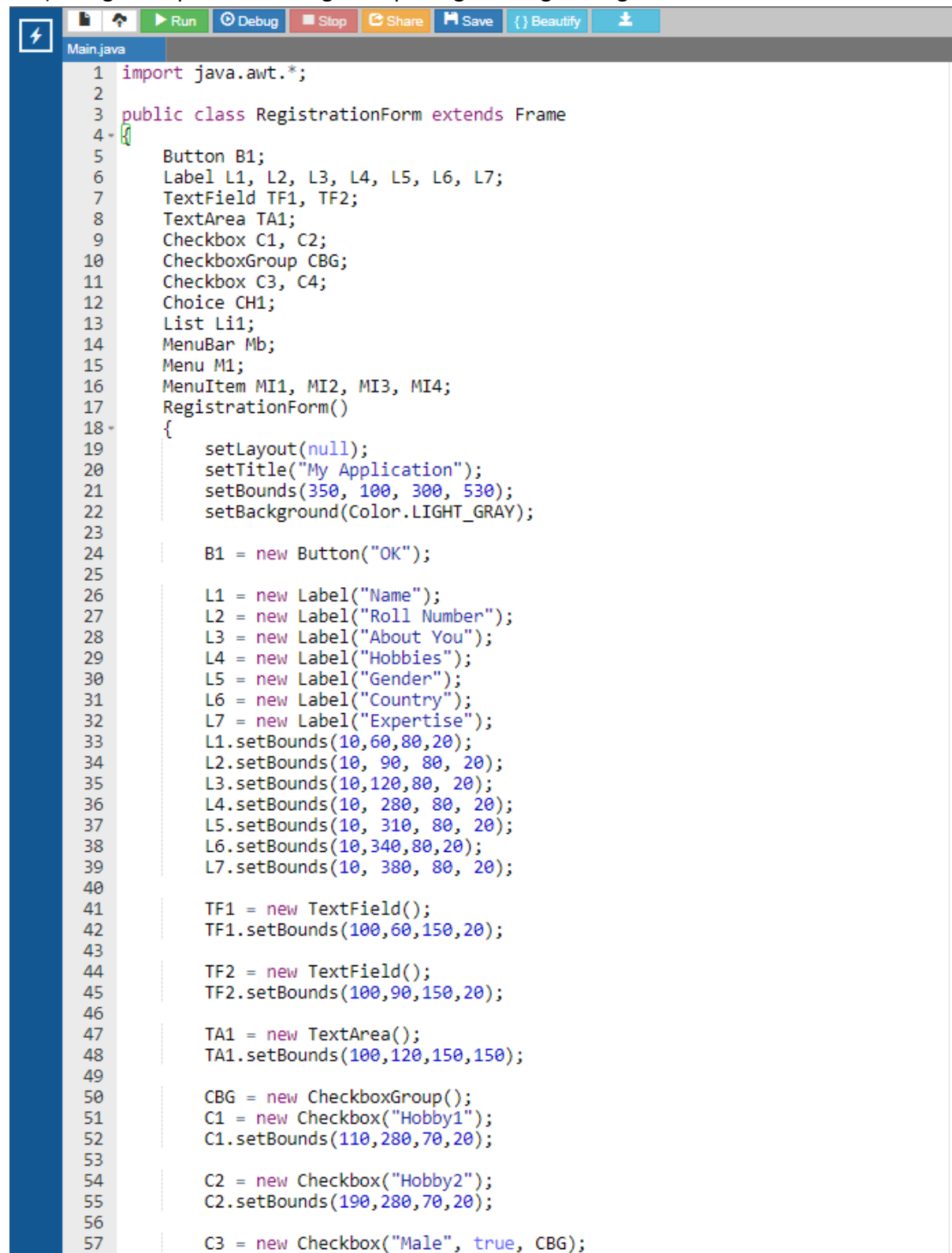
The image shows a screenshot of an IDE with a toolbar at the top containing icons for Run, Debug, Stop, Share, Save, Beautify, and a user icon. The file name 'Main.java' is visible in the editor's title bar. The code is a Java program that creates a window titled 'FrameDemo' and adds a text field and two buttons. The first button, 'Btn-1', is located at (100, 100) and the second button, 'Btn-2', is located at (100, 130). Both buttons have a width of 100 and a height of 30. The program also includes an ActionListener to handle button clicks, printing 'Btn-1 Clicked' and 'Btn-2 Clicked' to the console. The code is as follows:

```
1 import java.awt.Button;
2 import java.awt.Frame;
3 import java.awt.Label;
4 import java.awt.TextField;
5 import java.awt.event.ActionEvent;
6 import java.awt.event.ActionListener;
7
8 public class FrameDemo extends Frame implements ActionListener
9 {
10     Label l1;
11     TextField tf1;
12     Button B1, B2;
13
14     public FrameDemo ()
15     {
16         setLayout (null);
17         setBounds (100, 200, 300, 300);
18
19         l1 = new Label ("Name");
20         l1.setBounds (10, 50, 100, 30);
21         add (l1);
22
23         tf1 = new TextField ();
24         tf1.setBounds (110, 50, 100, 30);
25         add (tf1);
26
27         B1 = new Button ("Btn-1");
28         B1.setBounds (100, 100, 100, 30);
29         add (B1);
30
31         B2 = new Button ("Btn-2");
32         B2.setBounds (100, 130, 100, 30);
33         add (B2);
34
35         setVisible (true);
36         B1.addActionListener (this);
37         B2.addActionListener (this);
38     }
39
40     public static void main (String[]args)
41     {
42         new FrameDemo ();
43     }
44
45     @Override public void actionPerformed (ActionEvent ae)
46     {
47         if (ae.getSource () == B1)
48         {
49             System.out.println ("Btn-1 Clicked");
50         }
51         else if (ae.getSource () == B2)
52         {
53             System.out.println ("Btn-2 Clicked");
54         }
55     }
56 }
57
```

Output :-



Q 6) Design complete GUI using AWT package for registering a new user:



```
1 import java.awt.*;
2
3 public class RegistrationForm extends Frame
4 {
5     Button B1;
6     Label L1, L2, L3, L4, L5, L6, L7;
7     TextField TF1, TF2;
8     TextArea TA1;
9     Checkbox C1, C2;
10    CheckboxGroup CBG;
11    Checkbox C3, C4;
12    Choice CH1;
13    List Li1;
14    MenuBar Mb;
15    Menu M1;
16    MenuItem MI1, MI2, MI3, MI4;
17    RegistrationForm()
18    {
19        setLayout(null);
20        setTitle("My Application");
21        setBounds(350, 100, 300, 530);
22        setBackground(Color.LIGHT_GRAY);
23
24        B1 = new Button("OK");
25
26        L1 = new Label("Name");
27        L2 = new Label("Roll Number");
28        L3 = new Label("About You");
29        L4 = new Label("Hobbies");
30        L5 = new Label("Gender");
31        L6 = new Label("Country");
32        L7 = new Label("Expertise");
33        L1.setBounds(10,60,80,20);
34        L2.setBounds(10, 90, 80, 20);
35        L3.setBounds(10,120,80, 20);
36        L4.setBounds(10, 280, 80, 20);
37        L5.setBounds(10, 310, 80, 20);
38        L6.setBounds(10,340,80,20);
39        L7.setBounds(10, 380, 80, 20);
40
41        TF1 = new TextField();
42        TF1.setBounds(100,60,150,20);
43
44        TF2 = new TextField();
45        TF2.setBounds(100,90,150,20);
46
47        TA1 = new TextArea();
48        TA1.setBounds(100,120,150,150);
49
50        CBG = new CheckboxGroup();
51        C1 = new Checkbox("Hobby1");
52        C1.setBounds(110,280,70,20);
53
54        C2 = new Checkbox("Hobby2");
55        C2.setBounds(190,280,70,20);
56
57        C3 = new Checkbox("Male", true, CBG);
```

```
Main.java
58 C3.setBounds(110,310,70,20);
59
60 C4 = new Checkbox("female", true, CBG);
61 C4.setBounds(190,310,70,20);
62
63 CH1 = new Choice();
64 CH1.setBounds(110,340,140,20);
65
66 CH1.add("");
67 CH1.add("India");
68 CH1.addItem("Japan");
69 CH1.addItem("USA");
70 CH1.addItem("UK");
71
72 Li1 = new List(0, true);
73 Li1.setBounds(110,380,140,100);
74 Li1.add("Hardware");
75 Li1.add("Networking");
76 Li1.add("Java");
77 Li1.add("C-Sharp");
78 Li1.add("Chip Design");
79 Li1.addItem("Web Design");
80 Li1.addItem("Unix/Linux");
81 Li1.addItem("Production");
82 Li1.addItem("Machine Design");
83 Li1.addItem("Mobile comm.");
84
85 B1 = new Button("Ok");
86 B1.setBounds(100,490,100,30);
87
88 Mb = new MenuBar();
89 M1 = new Menu("Menu-1");
90
91 MI1 = new MenuItem("M-Item-1");
92 MI2 = new MenuItem("M-Item-2");
93 M1.add(MI1);
94 M1.add(MI2);
95 Mb.add(M1);
96 setMenuBar(Mb);
97 add(L1);
98 add(L2);
99 add(L3);
100 add(L4);
101 add(L5);
102 add(L6);
103 add(L7);
104 add(TF1);
105 add(TF2);
106 add(TA1);
107 add(C1);
108 add(C2);
109 add(C3);
110 add(C4);
111 add(CH1);
112 add(Li1);
113 add(B1);
114 setVisible(true);
```

```

    }

    public static void main(String[] args)
    {
        new RegistrationForm();
    }
}

```

Output :-

The screenshot shows a Java Swing window titled "My Application" with standard Windows-style window controls (minimize, maximize, close). The window contains a registration form with the following elements:

- Menu-1**: A header label for the form.
- Name**: A text input field.
- Roll Number**: A text input field.
- About You**: A large text area for a description.
- Hobbies**: Two checkboxes labeled "Hobby1" and "Hobby2".
- Gender**: Two radio buttons labeled "Male" (selected) and "female".
- Country**: A dropdown menu currently showing "India".
- Expertise**: A list box containing the following items: Hardware, Networking, Java, C-Sharp, Chip Design, Web Design, and Unix/Linux.
- Ok**: A button at the bottom center of the form.

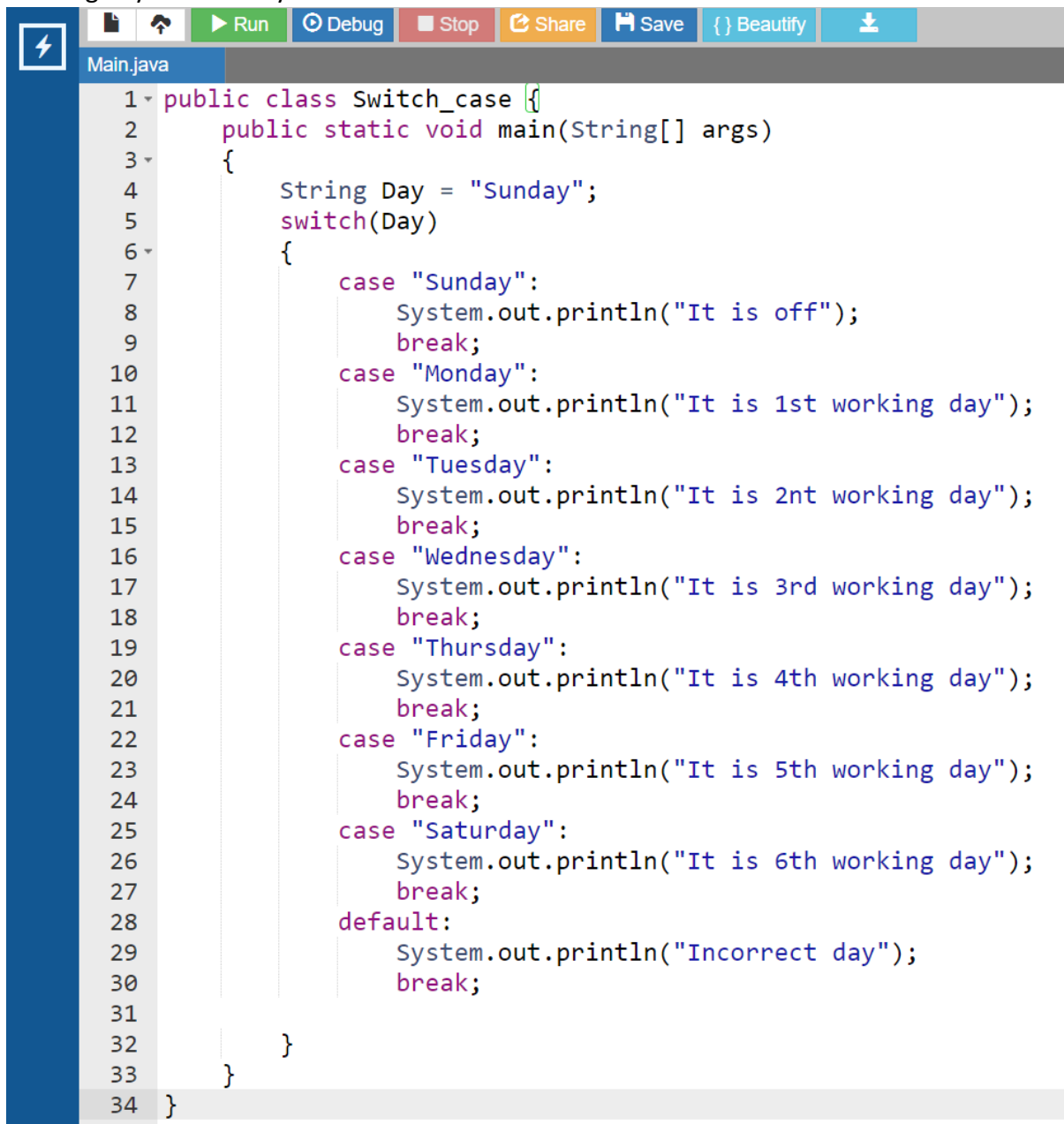
Q 7) WAP that accepts value of n ranging from 2 to 9 and generates the following pattern:

```
Main.java
3
4 public class TrianglePattern {
5
6     public static void main(String[] args)
7     {
8         int n = 7;
9         for(int i = 1; i<=n; i++)
10        {
11            int j;
12            for(j = 1; j<=n-i; j++)
13            {
14                System.out.print(" ");
15            }
16            int k = i;
17            for(; j<=n; j++)
18            {
19                System.out.print(k--);
20            }
21            k = 2;
22            for(; j<=n+i-1; j++)
23            {
24                System.out.print(k++);
25            }
26            System.out.println();
27        }
28    }
29 }
```

Output

```
java -cp /tmp/7LcB53Gi1s TrianglePattern
    1
   212
  32123
 4321234
543212345
65432123456
7654321234567
```

Q 8) WAP using switch-case construct that accepts name of a day and tells whether it is a working day or a holiday:



```
1 public class Switch_case {
2     public static void main(String[] args)
3     {
4         String Day = "Sunday";
5         switch(Day)
6         {
7             case "Sunday":
8                 System.out.println("It is off");
9                 break;
10            case "Monday":
11                System.out.println("It is 1st working day");
12                break;
13            case "Tuesday":
14                System.out.println("It is 2nt working day");
15                break;
16            case "Wednesday":
17                System.out.println("It is 3rd working day");
18                break;
19            case "Thursday":
20                System.out.println("It is 4th working day");
21                break;
22            case "Friday":
23                System.out.println("It is 5th working day");
24                break;
25            case "Saturday":
26                System.out.println("It is 6th working day");
27                break;
28            default:
29                System.out.println("Incorrect day");
30                break;
31        }
32    }
33 }
34 }
```

Output

```
java -cp /tmp/dOLPuR05TH Switch_case
It is off
```

Q 9) WAP, which proves that the first statement of any constructor always calls the superclass constructor:



```
2 class GrandParent extends Object {
3     GrandParent(int i) {
4         System.out.println("GrandParent class");
5     }
6 }
7
8 class Parent extends GrandParent
9 {
10     Parent()
11     {
12         super(10);
13         System.out.println("Parent class");
14     }
15 }
16
17 class Child extends Parent {
18     Child() {
19         System.out.println("Child Class");
20     }
21 }
22
23 class OtherClass {
24     public static void main(String[] args)
25     {
26         new Child();
27     }
28 }
```

Output

```
java -cp /tmp/NZSIUA52j9 OtherClass
GrandParent class
Parent class
Child Class
|
```