

## Q1. Checking if arguments are equal in generics

S.SRIHARI - 2018103601

The screenshot displays the Eclipse IDE interface. The main editor shows the source code for `CheckEqual.java` in the `lab9` package. The code defines a generic class `CheckEqual` with a static method `isEqualTo` and a `main` method. The `main` method uses a `Scanner` to take input from the user, including two decimal numbers and two `TeachingFaculty` objects. It then calls `isEqualTo` to check for equality and prints the results.

```
1 package lab9;
2 import labexam.TeachingFaculty;
3 import java.util.*;
4
5 public class CheckEqual {
6     static <T> void isEqualTo(T obj1, T obj2) {
7         System.out.println(obj1.equals(obj2));
8     }
9     public static void main(String[] args) {
10         // TODO Auto-generated method stub
11         float a, b;
12         Scanner sc = new Scanner(System.in);
13
14         System.out.println("Enter 2 decimal numbers");
15         a = sc.nextFloat();
16         b = sc.nextFloat();
17
18         System.out.println("Are these equal?");
19         isEqualTo(a, b);
20
21         System.out.println("Enter the department name and"
22             + " the faculty name of the teaching faculty 1");
23         TeachingFaculty obj1 = new TeachingFaculty(sc.next(), sc.next());
24
25         System.out.println("Enter the department name and"
26             + " the faculty name of the teaching faculty 2");
27         TeachingFaculty obj2 = new TeachingFaculty(sc.next(), sc.next());
28
29         System.out.println("Are these equal?");
30         isEqualTo(obj1, obj2);
31         obj2 = obj1;
32         System.out.println("Now assigned obj2 = obj1....Are these equal?");
33         isEqualTo(obj1, obj2);
34     }
35 }
36
37 }
```

The right-hand pane shows the console output of the program. It displays the execution flow, including user input and the program's output for each equality check.

```
<terminated> CheckEqual [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (Oct 17, 2020 11:09 AM)
Enter 2 decimal numbers
13.5
13.50
Are these equal?
true
Enter the department name and the faculty name of the teaching faculty 1
CSE
Srihari
Enter the department name and the faculty name of the teaching faculty 2
CSE
Srihari
Are these equal?
false
Now assigned obj2 = obj1....Are these equal?
true
```

The bottom status bar indicates the current state of the editor, showing "Writable", "Smart Insert", and the time "13:9:327". The system tray at the bottom right shows the date and time "11:09 AM 10/17/2020".

## Q2. Generics in java for pairs

The screenshot shows the Eclipse IDE interface. The left sidebar displays a project tree with a package named 'lab9'. The main editor window shows the source code for 'Generics.java'.

```
1 package lab9;
2
3 class Pair<F,S>{
4     private F first;
5     private S second;
6
7     public Pair(F a,S b) {
8         first = a;
9         second = b;
10    }
11
12    public F getFirst() {
13        return first;
14    }
15    public void setFirst(F a) {
16        first = a;
17    }
18
19    public S getSecond() {
20        return second;
21    }
22    public void setSecond(S b) {
23        second = b;
24    }
25 }
26
27 public class Generics {
28
29     public static void main(String[] args) {
30         // TODO Auto-generated method stub
31         Pair<Double,Boolean>temp = new Pair<Double,Boolean>(3.14,false);
32         System.out.println("First of object1:"+
33             temp.getFirst()+"\nSecond of object1:" + temp.getSecond());
34         Pair<String,Integer>temp2 = new Pair<String,Integer>("Srihari",2018103601);
35         System.out.println("First of object2:"+ temp2.getFirst()
36             +"\nSecond of object2:" + temp2.getSecond());
37     }
38 }
39 }
```

The right sidebar shows the 'Console' tab with the following output:

```
<terminated> Generics [Java Application] C:\Program Files\Java\jdk1.8.0_144\bin\javaw.exe (Oct 1
First of object1:3.14
Second of object1:false
First of object2:Srihari
Second of object2:2018103601
```

The bottom status bar indicates 'Writable', 'Smart Insert', and the time '17:6:229'.

### Q3. Guessing the random number

```
Q1.java TeachingFaculty.java Generics.java CheckEqual.java Guess.java ✕
1 package lab9;
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.*;
5
6 class Guess extends JFrame{
7     JTextField t1, t2, t3, t4;
8     JLabel j4;
9     ButtonListener bl1;
10    ButtonListener2 bl2;
11    ButtonListener3 bl3;
12
13    int rand=(int) (Math.random()*1000);
14    int count=0;
15
16    public Guess(){
17        Container c = getContentPane();
18        c.setLayout(null);
19        c.setBackground(Color.white);
20
21        JTextField j=new JTextField("I have a number between 1 and 1000. Can you guess my number?",20);
22        j.setForeground(Color.PINK);
23        j.setFont(new Font("Helvetica",Font.BOLD,24));
24        j.setSize(800,40);
25        j.setLocation(20,40);
26        j.setEditable(false);
27
28        t1=new JTextField(10);
29        t1.setSize(150,30);
30        t1.setLocation(250,140);
31
32        j4=new JLabel("Hey player guess the random number");
33        j4.setForeground(Color.ORANGE);
34        j4.setFont(new Font("Helvetica",Font.PLAIN,17));
```





Q1.java



TeachingFaculty.java



Generics.java



CheckEqual.java



Guess.java



```
35         j4.setSize(230,20);
36         j4.setLocation(290,190);
37
38         JButton b1=new JButton("Guess");
39         b1.setSize(150,40);
40         b1.setLocation(460,140);
41         b1=new ButtonListener();
42         b1.addActionListener(b1);
43
44
45         JButton b3=new JButton("Play Again");
46         b3.setSize(120,30);
47         b3.setLocation(330,270);
48         b13=new ButtonListener3();
49         b3.addActionListener(b13);
50
51
52         c.add(j4);
53         c.add(j);
54         c.add(t1);
55         c.add(b1);
56         c.add(b3);
57
58         setTitle("Number Guesser");
59
60         setSize(550,350);
61         setVisible(true);
62         setDefaultCloseOperation(EXIT_ON_CLOSE);
63     }
64
```

```
65 private class ButtonListener implements ActionListener{
66     public void actionPerformed(ActionEvent e){
67         int a = Integer.parseInt(t1.getText());
68         if(a<rand){
69             getContentPane().setBackground(Color.blue);
70             j4.setText("TOO LOW");
71         }
72         else if(a>rand){
73             getContentPane().setBackground(Color.red);
74             j4.setText("TOO HIGH");
75         }
76         else{
77             getContentPane().setBackground(Color.green);
78             j4.setText("CORRECT, YOU WIN!!!!");
79             j4.setForeground(Color.RED);
80             // j.setForeground(Color.RED);
81             t1.setEditable(false);
82         }
83         t1.requestFocus();
84         t1.selectAll();
85     }
86 }
87 private class ButtonListener2 implements ActionListener{
88     public void actionPerformed(ActionEvent e){
89         j4.setText(rand+" is the answer!");
90         t1.setText("");
91         t1.setEditable(false);
92     }
93 }
94
```

```
94
95⊖ private class ButtonListener3 implements ActionListener{
96⊖     public void actionPerformed(ActionEvent e) {
97         rand=(int) (Math.random()*1000);
98         t1.setText("");
99         j4.setText("Try and guess my number");
100        t1.setEditable(true);
101        t1.requestFocus();
102        getContentPane().setBackground(Color.white);
103    }
104 }
105
106⊖ public static void main(String[] args)
107 {
108     new Guess();
109 }
110
111
112 }
```

## Guessing the random number - output initially



```
lab5 98 t1.setText("");
lab6 99 j4.setText("Try and guess my number");
lab7 100 t1.setEditable(true);
lab9 101 t1.requestFocus();
102 getContentPane().setBackground(Color.white);
103 }
104 }
105
106 public static void main(String[] args)
107 {
108     new Guess();
109 }
110
111
112
```

## Guessing the random number entering higher value





## Guessing the random number entering a lower number



Number Guesser

I have a number between 1 and 1000. Can you guess my number?

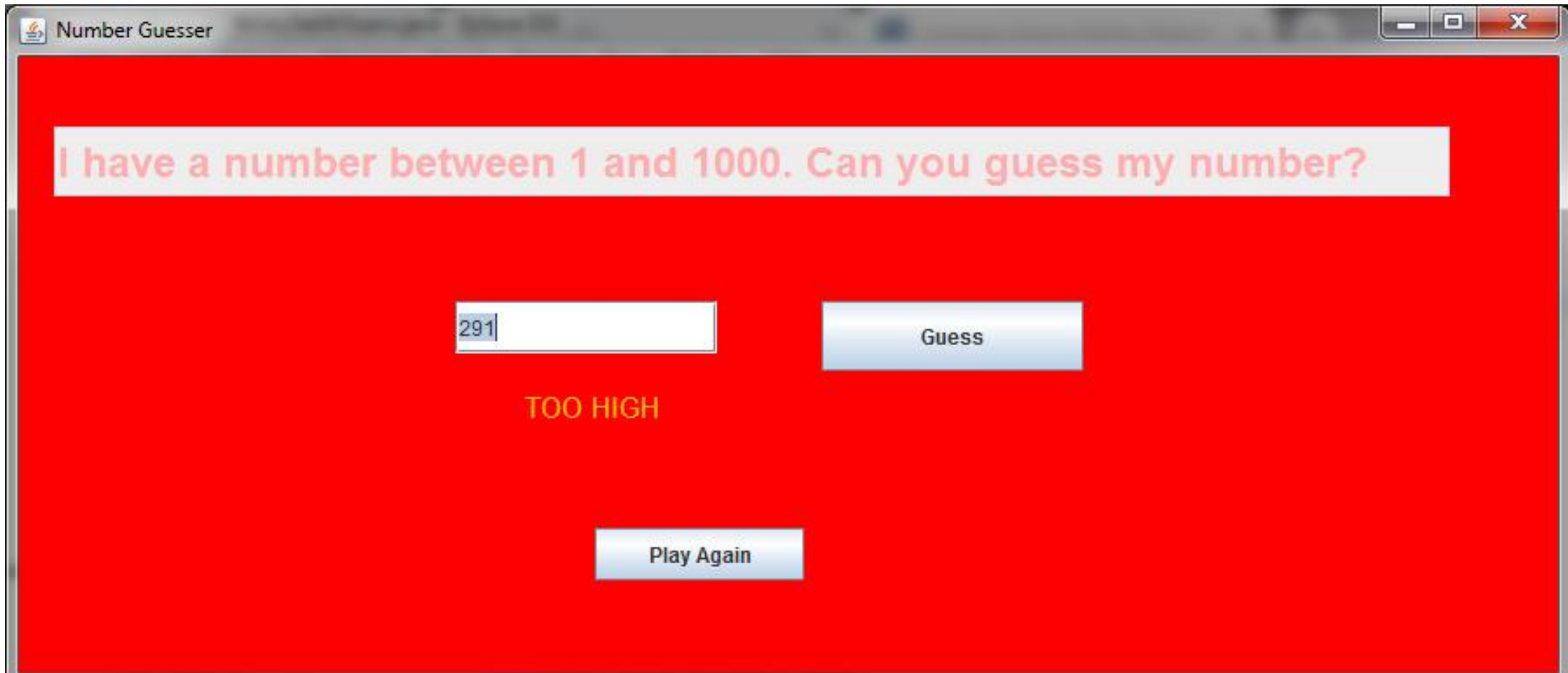
255

Guess

TOO LOW

Play Again

## Further guesses

A screenshot of a Java Swing window titled "Number Guesser". The window has a red background. At the top, a white text box contains the message "I have a number between 1 and 1000. Can you guess my number?". Below this, there is a text input field containing the number "291". To the right of the input field is a blue button labeled "Guess". Below the input field, the text "TOO HIGH" is displayed in orange. At the bottom center, there is another blue button labeled "Play Again".

Number Guesser

I have a number between 1 and 1000. Can you guess my number?

291

Guess

TOO HIGH

Play Again

Correct answer

The screenshot shows a window titled "Number Guesser" with a green background. At the top, a white banner contains the text "I have a number between 1 and 1000. Can you guess my number?". Below this, there is a text input field containing the number "289" and a "Guess" button. Centered on the screen is the text "CORRECT, YOU WIN!!!!" in red. At the bottom, there is a "Play Again" button.

Number Guesser

I have a number between 1 and 1000. Can you guess my number?

289

Guess

CORRECT, YOU WIN!!!!

Play Again

## Q4. Displaying the output - GUI Shown

The screenshot displays the Eclipse IDE interface. On the left, the Package Explorer shows a project structure with packages 'src', 'lab4', 'lab5', 'lab6', 'lab7', 'lab9', and 'labexam'. The 'labexam' package is expanded, showing files like 'Bonus.java', 'DepartmentFaculty.java', 'Gui.java', 'Inventory.java', 'NonTeachingFaculty.java', 'q2Tester.java', and 'TeachingFaculty.java'. The main editor window shows the code for 'Gui.java'.

```
1 package labexam;
2 import javax.swing.*;
3 import java.awt.*;
4
5 public class Gui extends JFrame {
6
7     public Gui(){
8         Container c = getContentPane();
9         c.setLayout(null);
10        Color color=new Color(224,224,224);
11        c.setBackground(color);
12
13        JButton b1=new JButton("Okay");
14        b1.setSize(90,30);
15        b1.setLocation(420,10);
16
17        JButton b2=new JButton("cancel");
18        b2.setSize(90,30);
19        b2.setLocation(420,50);
20
21        JButton b3=new JButton("setup");
22        b3.setSize(90,30);
23        b3.setLocation(420,90);
24
25        JButton b4=new JButton("help");
26        b4.setSize(90,30);
27        b4.setLocation(420,130);
28
29        JRadioButton r1=new JRadioButton("Selection",false);
30        r1.setSize(79,40);
31        r1.setLocation(240,55);
32        r1.setBackground(color);
33
34        JRadioButton r2=new JRadioButton("All",false);
35        r2.setSize(79,40);
36        r2.setLocation(240,85);
37        r2.setBackground(color);
38
39        JRadioButton r3=new JRadioButton("Image",false);
40        r3.setSize(79,40);
41        r3.setLocation(240,115);
42        r3.setBackground(color);
43
44        JRadioButton r4=new JRadioButton("Text",false);
45        r4.setSize(79,40);
46        r4.setLocation(240,145);
47        r4.setBackground(color);
48
49        JRadioButton r5=new JRadioButton("Code",false);
50        r5.setSize(79,40);
51        r5.setLocation(240,175);
52        r5.setBackground(color);
53
54        JRadioButton r6=new JRadioButton("Applet",false);
55        r6.setSize(79,40);
56        r6.setLocation(240,205);
57        r6.setBackground(color);
58
59        PrintDialog dialog=new PrintDialog(c);
60        dialog.setVisible(true);
61    }
62}
```

Overlaid on the code editor is a 'PrintWriter' dialog box. It features a title bar with standard window controls. The dialog contains a section titled 'Printer: MyPrinter' with a preview area. Below this, there are checkboxes for 'Image', 'Text', 'Code', and 'Applet'. A 'Print Quality' dropdown menu is set to 'high'. There is also a checkbox for 'Print to file'. On the right side of the dialog, there are five buttons: 'Okay', 'cancel', 'setup', and 'help'.

The bottom status bar of the IDE shows 'Writable', 'Smart Insert', and the time '16:9:379'. The system tray at the bottom right indicates the time '12:35 PM' and date '10/17/2020'.

