PCS SLO review

February 2018 answers

Questions 6-end

6. Errors

The code segment below is intended to print " Passing score" only when the value of the variable “testScore” is at least 70. Assume the variable “testScore” has been declared and initialized properly as an int.

```
if (testScore >= 70) {
    System.out.println(" Passing score" );
}
```

Modify the code to produce a syntax error.

```
if (testScore >= 70) {
    system.out-println(" Passing score" )
} // s is lower case in system
```

Modify the code to produce a logic error.

```
if (testScore >= 70) {
    System.out.println(" failing score" );
} // it runs but gives a bad answer
```

Modify the code to produce a run time error.

```
if (testScore >= 70/0) {
    System.out.println(" Passing score" );
} // you can’t divide by zero
```

7. Logic comparisons

Assume variables x and y in the logical expressions below have been declared and initialized properly. Describe the value(s) of these variable(s) that make each expression evaluate as true.

- x < 5 && y > 9 **any time that both: x is less than 5 AND y is greater than 9**
- x < 5 || y > 9 **any x less than 5 or any y greater than 9**
- x >= 5 && y < 9 **both x>=5 AND y<9 have to be true.**
- x > 5 || x <= 9 **either x is more than 5 or x is less than or equal to 9:**
  every number ie either more than 5 or less than 9 so always true
- 9 > 5 || x < 9 **always true because 9 is always > 5**
- 5 > 5 && x < 9 **always false because 5 is never > 5**

8. Operators	Description
Relational Operators	
= =	Equal to
!=	not equal to
>	greater
>=	greater than or equal to
<	Less than
<=	less than or equal to
Logical Operators	
&&	and
||	or
!	not
Simple Assignment Operator	
=	assigns a value to a variable
Compound Assignment Operators	
+=	Add and assign the result
-=	Subtract and assign a result
*=	Multiply and assign a result
/=	Divide and assign the result

9. Given the code answer the questions.

```
public class IfElseDemo {
 public static void main(String[] args) { >
  int testScore = 79;
  char grade;
  if (testScore >= 92) {
              grade  = 'A';
  } else if (testScore >= 84) {
              grade  = 'B';
  } else if (testScore >= 76) {
              grade  = 'C';
  } else if (testScore >= 68) {
              grade  = 'D';
  } else {
              grade  = 'F';
  }
  System.out.println("Grade  = " + grade);
  }
}
```

What is the lowest value the variable testScore can be assigned and still print a 'C' as a grade?

76

What is the highest value that can be assigned to the variable testScore and print B as a grade?

91

What is the highest value that can be assigned to the variable testScore and print F as a grade?

67

10 Explain the expressions that are **not** relational expressions.

For these expressions, identify the type of the expression or statement.(assignment statement, math expression or error)

Relational expressions == logic expressions == true or false

- x = = 3 _______________**relational**______________
- **x = 3 This is an assignment statement**
- x >= 3 _______________**relational**______________
- **x * 3 This is a math expression**
- 3 < x _______________**relational**______________
- x - 3 <= 10 _______________**relational**______________

11. Given the following code what is the last value of “i” to be printed? __________

```
int stop =  25;
int i;
for (i = 0; i  < stop; i++){
     System.out.println(   "The value of i: " + i );
}
```

12. What values entered for the int amount will cause the body of the loop to be executed?

```
Scanner sc = new  Scanner();
  int amount =  sc.nextInt();


  while ( amount >= 50 || amount < 60   ) {
      System.out.print( "Number  is out of range.”);
      amount = sc.nextInt(  );
```

- }
  Any value because all numbers are either greater than 50 or less than 60
- Rewrite the conditional expression in the previous problem so the body of the loop is entered only when the amount entered is at least 12 but less than 22
- amount >= 12 && amount < 22

Given the String *str* write the code to find:

- Its length ______________________________________
- A string with only its last character ______________________________________
- A string with only its first three characters ______________________________
- A string with only the last 4 characters _____________________________________
- A string with all but the first character in str _______________________________
- The index of the letter q in str or -1 if 'q' is not there __________________________

- Note: The first member of the array fish is fish[0]; The length of the array fish is fish.length
  The last member of the array fish is fish[fish.length -1]

The following code is intended to print the number of times the value of min changes as an array is searched for its minimum value.

```
• int [ ] miles = {250, 350, 150,  100, 325, 400, 290};
•   int min = Integer.MAX_VALUE;
•   int i = 0;
•   while (i <  miles.length){
•     int  count = 0;
•     if  ( miles[ i ] < min ){
•       min  = miles[ i ];
•       count++;
•     }
•     i++;
•   }
•   System.out.println("Changes:   " + count) ; // why is there an error here?
```

GUI Questions

Here is an example of how to create an empty box that the user can enter information into:

```
JTextField box1 = new JTextField("0", 4); // (start value, field width)
```

1. How many characters wide will this box be? ___________
   JTextField box2 = new JTextField("6", 5); // (start value, field width)

2. Write the Java to declare and instantiate a JTextField named *box* with a starting value of -1 that is 12 spaces wide.

3. Write the Java to convert the number 5 into a string and assign it to a String named *str* .

4. Write the Java to get the string that is in a JTextField named *studentName_txt* and assign it to a String named *student*.

5. Write the Java to get the string that is in a JTextField named *boxWeight_txt* and assign it to a double named *weight*.

1. 5
2. JTextField box = new JTextField(“-1”,12);
3. String str = 5 + “”;
4. Student = studentName_txt.getText();
5. Double weight = Double.parseDouble(boxWeight_txt.getText());

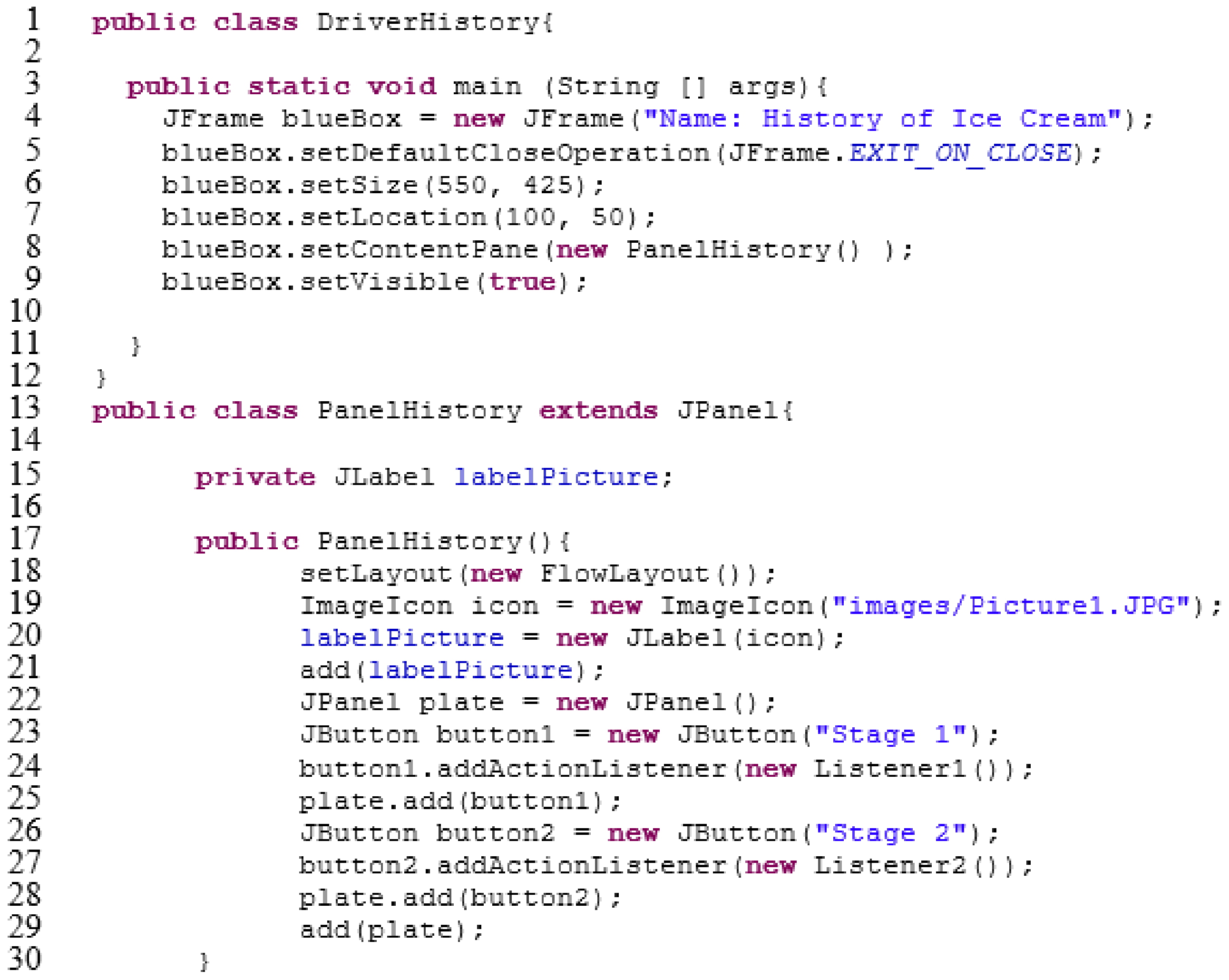
Questions refer to the code

6 **Based on line 4:**

- What type of object is constructed?
- **frame**
- What type of parameter does the constructor receive?
- **String (title of the frame)**

7 **Based on line 21:**

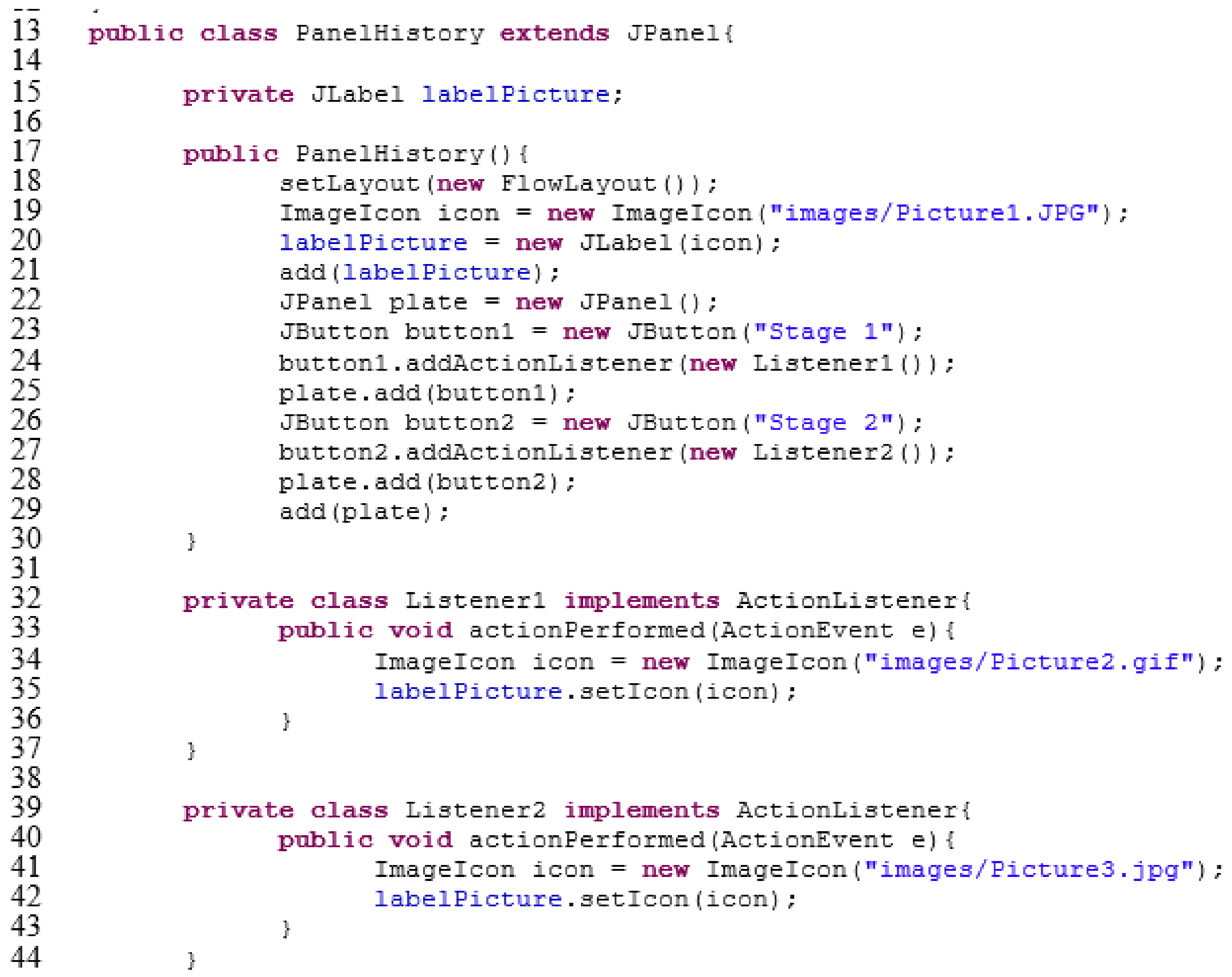
- What type of object is being added?
- **label**
- To what type of object it is added?
- **Panel (or Jpanel)**



```
1   public class DriverHistory{
2
3     public static void main (String [] args){
4       JFrame blueBox = new JFrame("Name: History of Ice Cream");
5       blueBox.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
6       blueBox.setSize(550, 425);
7       blueBox.setLocation(100, 50);
8       blueBox.setContentPane(new PanelHistory() );
9       blueBox.setVisible(true);
10
11    }
12  }
13  public class PanelHistory extends JPanel{
14
15        private JLabel labelPicture;
16
17        public PanelHistory(){
18              setLayout(new FlowLayout());
19              ImageIcon icon = new ImageIcon("images/Picture1.JPG");
20              labelPicture = new JLabel(icon);
21              add(labelPicture);
22              JPanel plate = new JPanel();
23              JButton button1 = new JButton("Stage 1");
24              button1.addActionListener(new Listener1());
25              plate.add(button1);
26              JButton button2 = new JButton("Stage 2");
27              button2.addActionListener(new Listener2());
28              plate.add(button2);
29              add(plate);
30        }
```

Questions refer to the code

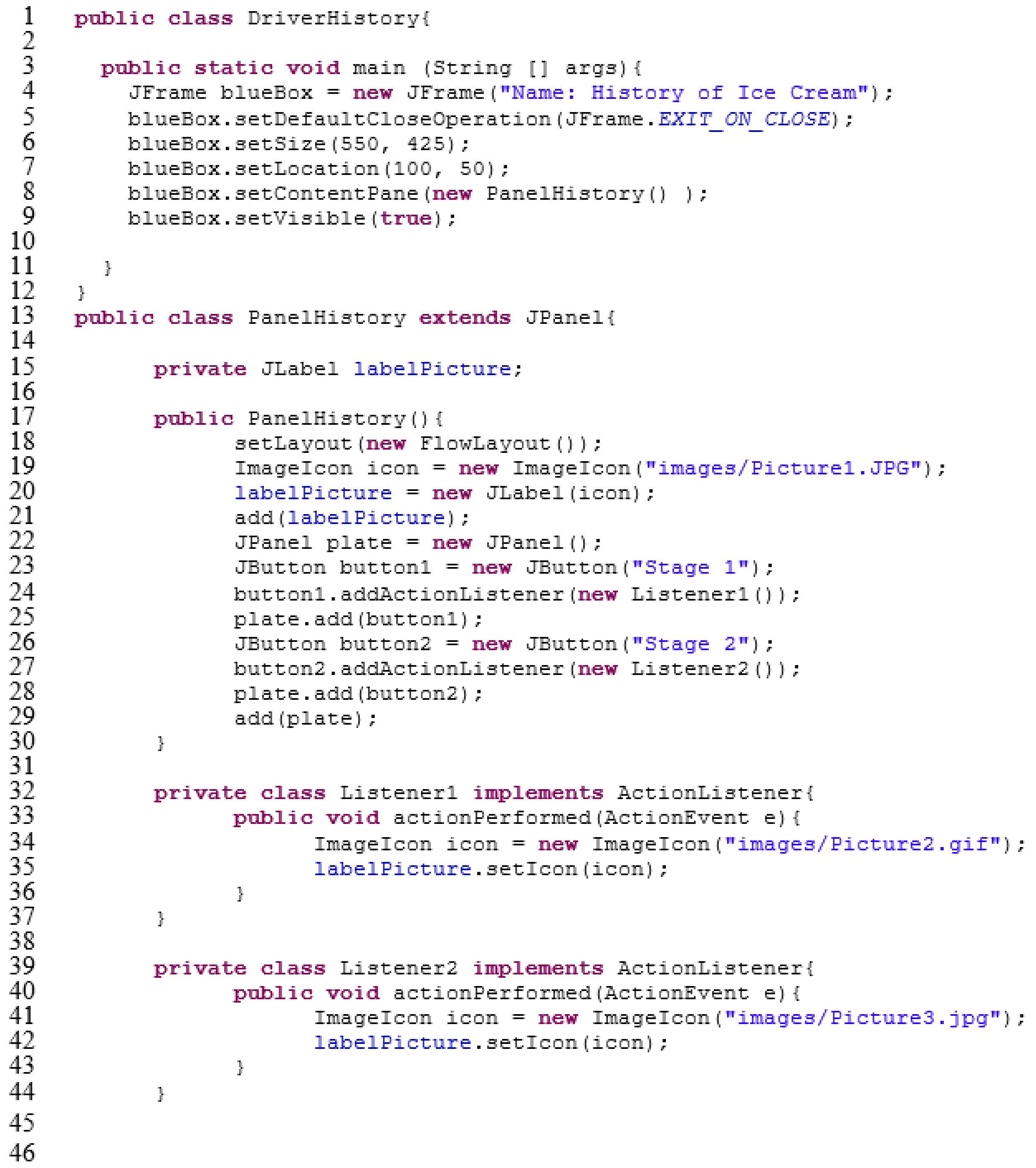
- On what two lines are listeners registered with JComponents?
- ________________
- 9. What is the name of the interface the Listeners implement?
- ________________
- 10. What method does that interface require?
- ________________
- 11. How does the function of a JButton change if a listener is not registered and implemented?
- ____________________________________
  ____



```
13  public class PanelHistory extends JPanel{
14
15          private JLabel labelPicture;
16
17          public PanelHistory(){
18                  setLayout(new FlowLayout());
19                  ImageIcon icon = new ImageIcon("images/Picture1.JPG");
20                  labelPicture = new JLabel(icon);
21                  add(labelPicture);
22                  JPanel plate = new JPanel();
23                  JButton button1 = new JButton("Stage 1");
24                  button1.addActionListener(new Listener1());
25                  plate.add(button1);
26                  JButton button2 = new JButton("Stage 2");
27                  button2.addActionListener(new Listener2());
28                  plate.add(button2);
29                  add(plate);
30          }
31
32          private class Listener1 implements ActionListener{
33                  public void actionPerformed(ActionEvent e){
34                          ImageIcon icon = new ImageIcon("images/Picture2.gif");
35                          labelPicture.setIcon(icon);
36                  }
37          }
38
39          private class Listener2 implements ActionListener{
40                  public void actionPerformed(ActionEvent e){
41                          ImageIcon icon = new ImageIcon("images/Picture3.jpg");
42                          labelPicture.setIcon(icon);
43                  }
44          }
```

What does it do?

- What does this code do?



```java
public class DriverHistory{

 public static void main (String [] args){
  JFrame blueBox = new JFrame("Name: History of Ice Cream");
  blueBox.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
  blueBox.setSize(550, 425);
  blueBox.setLocation(100, 50);
  blueBox.setContentPane(new PanelHistory() );
  blueBox.setVisible(true);

 }
}
public class PanelHistory extends JPanel{

   private JLabel labelPicture;

   public PanelHistory(){
      setLayout(new FlowLayout());
      ImageIcon icon = new ImageIcon("images/Picture1.JPG");
      labelPicture = new JLabel(icon);
      add(labelPicture);
      JPanel plate = new JPanel();
      JButton button1 = new JButton("Stage 1");
      button1.addActionListener(new Listener1());
      plate.add(button1);
      JButton button2 = new JButton("Stage 2");
      button2.addActionListener(new Listener2());
      plate.add(button2);
      add(plate);
   }

   private class Listener1 implements ActionListener{
      public void actionPerformed(ActionEvent e){
         ImageIcon icon = new ImageIcon("images/Picture2.gif");
         labelPicture.setIcon(icon);
      }
   }

   private class Listener2 implements ActionListener{
      public void actionPerformed(ActionEvent e){
         ImageIcon icon = new ImageIcon("images/Picture3.jpg");
         labelPicture.setIcon(icon);
      }
   }
```

Text fields

- 12. What type of value is returned when a JTextField's getText method is used?
- **String**
- 13. Write the Java to obtain a String from a JTextField named *lumens_txt,* convert it to an integer, and assign it to an int name *num*.
- **Int num = Integer.parseInt(***lumens_txt.getText());*
- 14. Write the Java to obtain a String from a JTextField named *lumens_txt,* convert it to a double and assign it to an double name *dec*.
- **double dec = num = Double.parseDouble(***lumens_txt.getText());*