You have 80 minutes to complete this quiz. For coding questions, you do not need to include comments, and you should assume that all necessary files have already been imported.

Good luck!

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Question 1</td>
<td>/30</td>
</tr>
<tr>
<td>Question 2</td>
<td>/20</td>
</tr>
<tr>
<td>Question 3</td>
<td>/30</td>
</tr>
<tr>
<td>Question 4</td>
<td>/20</td>
</tr>
<tr>
<td>Total</td>
<td>/100</td>
</tr>
</tbody>
</table>
Question 1. ArrayLists (30 points)

You are the director of a large cruise ship. There are two classes in your program: Passenger and CruiseShip. The Passenger class, which represents a single passenger on the ship, has three data members: the passenger’s age, name, and a boolean variable stating whether the passenger is a first class passenger or not. Passenger is defined as follows:

```java
package travel;
public class Passenger {
    String name;
    int age;
    boolean isFirstClass;
    //isFirstClass is true when the passenger is a first class passenger, false otherwise

    public Passenger(String n, int a, boolean c){
        name = n;
        age = a;
        isFirstClass = c;
    }
}
```

The CruiseShip class contains an ArrayList of Passengers, a constructor, and two additional methods.

//code continued on next page
package travel;
public class CruiseShip {
    ArrayList<Passenger> passengers;
    /*
    * The constructor simply initializes passengers
    * to an empty ArrayList.
    */
    public CruiseShip()
    {
    }

    /*
    * This method adds the inputted Passenger p
    * to the ArrayList passengers.
    */
    public void addPassenger(Passenger p)
    {
    }
    //code continued on next page
/*
 * This method returns an ArrayList of Passengers that meet the
 * criteria to have access to the cruise ship’s exclusive club.
 * These passengers must have first class status, and must have an
 * age of at least 21
 */

public ArrayList<Passenger> getExclusiveMembers() {

}
Question 2. Multiple Choice (20 points)

Circle all answers that are correct.

1. You always have to add a "throws" clause to a method signature if the method could throw an exception.

   TRUE  FALSE

2. Consider the following code:

   ```java
   public class EnclosingClass {
       int counter;
       private static class MysteryClass {
           public MysteryClass() {
               counter++;
           }
       }
   }
   ```

   a. Is MysteryClass an inner class or nested class

      Inner Class  Nested Class

   b. This code will compile.

      TRUE  FALSE

3. A concrete class that extends an abstract class must implement all abstract methods.

   TRUE  FALSE

4. A class can extend multiple classes.

   TRUE  FALSE

5. A subclass can access ________ fields of its superclass in the same package.

   Public  Protected  Private  Package  No
6. An abstract class may have

<table>
<thead>
<tr>
<th>Public fields</th>
<th>Protected fields</th>
<th>Private fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract methods</td>
<td>Concrete methods</td>
<td></td>
</tr>
</tbody>
</table>
Question 3. Swing (30 points)

Read the following program then answer the questions which involve drawing the JFrame QuirkyCalculator initially and after buttons are pressed. Assume all necessary classes have been imported.

```java
public class QuirkyCalculator extends JFrame implements ActionListener{
    JPanel panel1 = new JPanel();
    JLabel totalLabel = new JLabel("total: 0");
    JPanel panel2 = new JPanel();
    JPanel panel3 = new JPanel();
    int total=0;
    JButton a = new JButton("Button A");
    JButton b = new JButton("Button B");
    JButton c = new JButton("Button C");

    public QuirkyCalculator (String name) {
        super(name);
    }

    public void addComponentsToPane(final Container pane) {
        a.addActionListener(this);
        b.addActionListener(this);
        c.addActionListener(this);

        panel1.add(new JLabel("Pick one");
        panel2.add(a);
        panel2.add(b);
        panel2.add(c);
        panel3.add(totalLabel);

        pane.add(panel2, BorderLayout.CENTER);
        pane.add(panel1, BorderLayout.NORTH);
        pane.add(panel3, BorderLayout.SOUTH);
    }

    //code continues on next page
```
public void actionPerformed(ActionEvent e){
    if(e.getSource().equals(a)){
        total+=1;
    }
    if(e.getSource().equals(b)){
        total*=2;
        totalLabel.setText("total: "+total);
    }
    if(e.getSource().equals(c)){
        total-=1;
        totalLabel.setText("total: "+total);
    }
}

public static void main(String[] args) {
    QuirkyCalculator frame = new QuirkyCalculator("Quirky");
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.addComponentsToPane(frame.getContentPane());
    frame.pack();
}
1. Draw the JFrame as it initially appears. Be accurate about labels and layout. Indicate on your drawing the names of all the JComponents.

2. The user has clicked on Button C then Button B then Button A. Draw the JFrame as it now appears. Be accurate about labels and layout. This time, you do not need to indicate on your drawing all the JComponents.
Question 4: Recursion (20 points)

Read the following methods and write either the expression implemented by the method (e.g., $\frac{x}{y}$ or $\sqrt[y]{x}$) or the numerical value returned by the method call. Either will be counted as correct.

Assume that $x$ and $y$ are positive. Show work for partial credit.

1.  
   ```java
   public static int methodA(int x, int y){
       if(y == 1){
           return x;
       }
       else{
           return x + methodA(x, y - 1);
       }
   }
   
   methodA(5,100);
   ```
2. public static int methodB(int x, int y){
    if(x < y){
        return x;
    }
    else{
        return methodB(x-y, y);
    }
}

methodB(10000,3);