**SHORT ANSWER**

1. How have user interface components changed since the early versions of Java?

ANS:

In early versions of Java, components had simple names, such as Frame and Button. The components created from these original classes did not have a consistent appearance when used with different browsers and operating systems. When Java’s creators designed new, improved classes, they needed new names for the classes, so they used a *J* in front of each new class name. Hence, Swing components have names like JFrame, JButton, JScrollbar, JOptionPane, and so on.

PTS: 1 REF: 740

2. What are the four JFrame constructors?

ANS:

\* JFrame() constructs a new frame that initially is invisible and has no title.

\* JFrame(String title) creates a new, initially invisible JFrame with the specified title.

\* JFrame(GraphicsConfiguration gc) creates a JFrame in the specified GraphicsConfiguration of a screen device with a blank title.

\* JFrame(String title, GraphicsConfiguration gc) creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen.

PTS: 1 REF: 741-742

3. How can you customize the appearance of a JFrame?

ANS:

The appearance of the JFrame is provided by the operating system in which the program is running. For example, the coffee-cup icon in the frame’s title bar and the Minimize, Restore, and Close buttons look and act as they do in other Windows applications. The icon and buttons are known as window decorations; by default, window decorations are supplied by the operating system. However, you can request that Java’s look and feel provide the decorations for a frame. A look and feel is the default appearance and behavior of any user interface. Optionally, you can set a JFrame’s look and feel using the setDefaultLookAndFeelDecorated() method.

PTS: 1 REF: 744-745

4. How do you change the text of a JLabel? Provide an example.

ANS:

You can change the text in a JLabel by using the setText() methodwith the JLabel object and passing a String to it. For example, the following code changes the value displayed in the greeting JLabel:

greeting.setText("Howdy");

PTS: 1 REF: 750

5. How do you change the font of a JLabel?

ANS:

To give a JLabel object a new font, you can create a Font object, as in the following:

Font headlineFont = new Font("Monospaced", Font.BOLD, 36);

The typeface name is a String, so you must enclose it in double quotation marks when you use it to declare the Font object. Then you use the setFont() method to assign the Font to a JLabel with a statement such as:

greeting.setFont(headlineFont);

PTS: 1 REF: 751

6. Describe the FlowLayout manager.

ANS:

When you use a FlowLayout instead of a BorderLayout, components do not lie on top of each other. Instead, the flow layout manager places components in a row, and when a row is filled, it automatically spills components into the next row. By default, the components in each row are centered.

PTS: 1 REF: 753-754

7. What are some of the decisions you must make when extending a JFrame?

ANS:

When you extend a JFrame to create a new custom class, you must remember to make decisions as to which attributes you want to set within the class and which you want to leave to the applications that will use the class. For example, you can place the setVisible() statement within the JFrame child class constructor (using either an explicit or implied this reference), or you can allow the application to use a setVisible() statement (using the name of an instantiated object followed by a dot and the method name). Either one works, but if you fail to do either, the frame will not be visible.

PTS: 1 REF: 757-758

8. How do you modify whether or not a JTextField can be edited?

ANS:

When a JTextField has the capability of accepting keystrokes, the JTextField is editable. If you do not want the user to be able to enter data in a JTextField, you can use the setEditable() methodto change the editable status of a JTextField from its default value of true.

PTS: 1 REF: 760

9. Provide an example of an event-driven program.

ANS:

If you use a word-processing program, you have dozens of choices at your disposal at any moment in time. You can type words, select text with the mouse, click a button to change text to bold, click a button to change text to italic, choose a menu item, and so on. With each word-processing document you create, you choose options in any order that seems appropriate at the time. The word-processing program must be ready to respond to any event you initiate.

PTS: 1 REF: 765

10. What type of method must you implement when you register an event listener?

ANS:

When you register a component (such as a JFrame) to be a listener for events generated by another component (such as a JCheckBox), you must write a method that reacts to any generated event. You cannot choose your own name for the reacting methods—specific methods react to specific event types.

PTS: 1 REF: 776

11. Write the statement to create a JLabel named welcome that holds the words “Welcome Home”.

ANS:

JLabel greeting = new JLabel("Welcome Home");

PTS: 1 REF: 749

12. Write the statement to create a layout manager named myLayout that centers components in each row.

ANS:

FlowLayout myLayout = new FlowLayout(FlowLayout.CENTER);

PTS: 1 REF: 754

13. Write the statement to provide a JTextField object named myInfo that allows enough room for a user to enter 15 characters.

ANS:

JTextField myInfo = new JTextField(15);

PTS: 1 REF: 759

14. Write the statement to clear out a JTextField named myText.

ANS:

myText.setText("");

PTS: 1 REF: 760

15. Write the statement to create a JButton named submitButton with the label “Submit your data”.

ANS:

JButton submitButton = new JButton("Submit your data");

PTS: 1 REF: 761

**CASE**

1. Create the statements to construct two JFrames. One JFrame should be declared as welcomeFrame and have the title “Welcome”, and the second JFrame should be declared as noTitleFrame and have no title.

ANS:

JFrame welcomeFrame = new JFrame("Welcome");

JFrame noTitleFrame = new JFrame();

PTS: 1 REF: 742

2. Assume you have declared a JFrame named welcomeFrame. Write the statement to set the welcomeFrame object’s size to 300 pixels horizontally by 110 pixels vertically. Create a second statement to set the JFrame’s title to display the text “My Sized Frame”.

ANS:

welcomeFrame.setSize(300, 110);

welcomeFrame.setTitle("My Sized Frame");

PTS: 1 REF: 743

3. import javax.swing.\*;

public class JFrame1

{

public static void main(String[] args)

{

JFrame aFrame = new JFrame("First frame");

aFrame.setSize(300, 125);

aFrame.setVisible(true);

}

}

The three shaded statements in the main() method above are important. Explain the purpose of each statement and why they are necessary for creating a JFrame.

ANS:

The first shaded statement declares a JFrame named aFrame. After you instantiate aFrame, the second statement sets the JFrame size. If you do not set its size, you see only the title bar of the JFrame because the JFrame size is 0 0 by default. In the third shaded line, you set the JFrame visibility to true. If you do not use setVisible(true), you do not see the JFrame. The default state for a JFrame is invisible.

PTS: 1 REF: 743-744

4. import javax.swing.\*;

public class JFrameLook

{

public static void main(String[] args)

{

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JFrame myFrame = new JFrame("Look and Feel");

myFrame.setSize(300, 120);

myFrame.setVisible(true);

}

}

In the shaded line above, write the statement to set the JFrame’s look and feel using the setDefaultLookAndFeelDecorated() method.

ANS:

JFrame.setDefaultLookAndFeelDecorated(true);

PTS: 1 REF: 745

5. import javax.swing.\*;

public class JFrameLabel

{

public static void main(String[] args)

{

final int FRAME\_WIDTH = 300;

final int FRAME\_HEIGHT = 120;

JFrame myFrame = new JFrame("Frame with label");

myFrame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

myFrame.setVisible(true);

myFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

}

}

The above code shows an application in which a JFrame is created and its size, visibility, and close operation are set. In the blank lines provided, write the code to create a JLabel named thanks that holds the words “Thank you for your business”. Then, write the statement to add the JLabel to the JFrame.

ANS:

JLabel thanks = new JLabel("Thank you for your business");

aFrame.add(thanks);

PTS: 1 REF: 749

6. import javax.swing.\*;

import java.awt.\*;

public class JFrame4

{

public static void main(String[] args)

{

final int FRAME\_WIDTH = 250;

final int FRAME\_HEIGHT = 100;

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

JFrame aFrame = new JFrame("Fourth frame");

aFrame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT); aFrame.setVisible(true);

aFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel greeting = new JLabel("Good day");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

aFrame.add(greeting);

}

}

In the first blank line provided, write the statement to create a Font object named myLook with a typeface of Times New Roman, italic, and 30-point size. In the second blank line, create the statement to apply myLook to the greeting JLabel.

ANS:

Font myLook = new Font("Times New Roman", Font.ITALIC, 30);

greeting.setFont(headlineFont);

PTS: 1 REF: 751

7. import javax.swing.\*;

import java.awt.\*;

public class JFrame6

{

public static void main(String[] args)

{

final int FRAME\_WIDTH = 250;

final int FRAME\_HEIGHT = 100;

JFrame aFrame = new JFrame("Sixth frame");

aFrame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);

aFrame.setVisible(true);

aFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JLabel greeting = new JLabel("Hello");

JLabel greeting2 = new JLabel("Who are you?");

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

aFrame.add(greeting);

aFrame.add(greeting2);

}

}

Using the above code, write the FlowLayout statement in the blank line provided that will display the two greeting JLabels side by side.

ANS:

aFrame.setLayout(new FlowLayout());

PTS: 1 REF: 754

8. Write the statement to add a tool tip that displays “Submit your form” to a button named submit.

ANS:

submit.setToolTipText("Submit your form");

PTS: 1 REF: 762-763

9. Write the statement to create a JCheckBox object named feeWaived that is selected. Include the label “Fee Waived”.

ANS:

JCheckBox feeWaived = new JCheckBox("Fee Waived", true);

PTS: 1 REF: 780

10. Write the code to build a JComboBox named sideDish with no arguments. Then, create the addItem() methods to provide options for “Fries”, “Salad”, and “Fruit”.

ANS:

JComboBox<String> sideDish = new JComboBox<String>();

sideDish.addItem("Fries");

sideDish.addItem("Salad");

sideDish.addItem("Fruit");

PTS: 1 REF: 783