





Unit 4 – ICS3U0 – Java Applets

Sample Test May 2, 2023

Name: Gorski

Total	Knowledge 	Communication 	Thinking 	Application 
(100)	(27)	(25)	(28)	(20)

Knowledge

1. Identify the following code that would appear in an applet. /9

(a) Name 2 types of widgets.

JLabel JButton (also JTextField)

(b) Name 2 accessors.

getActionCommand getText

(c) Identify two distinct lines of code that can be used in init **but not** in actionPerformed.

add(widget); button.setActionCommand("done");

(d) Identify two lines of code that are needed to make a button clickable.

addActionListener setActionCommand

(e) Identify a library: (import) javax.swing

(f) Name 3 mutators that can be used on a JButton.

setFont setForeground setBackground
(setActionCommand, setEnabled, setIcon, setPreferredSize, setText)

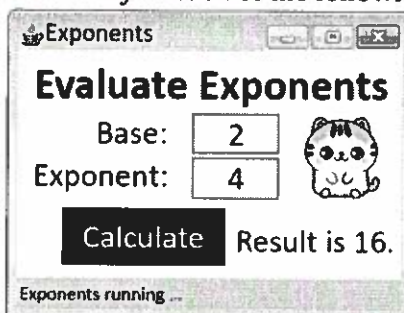
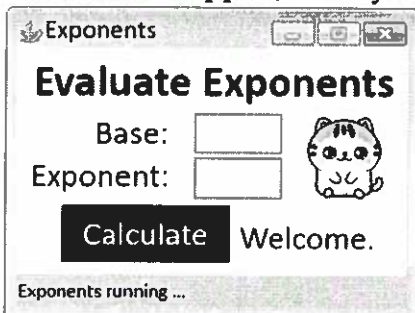
2. Fill in the following pieces of code: /3

(a) Write "Welcome" in the status bar. showStatus("Welcome");

(b) Set the applet background to be cyan. setBackground(Color.cyan);

(c) Resize the applet to 450, 800. resize(450, 800);

3. In the below applet, identify the how many of each of the following appear: /3



(a) Action Listeners

1

(b) Global widgets

3

(c) JLabels

5

(d) Total widgets

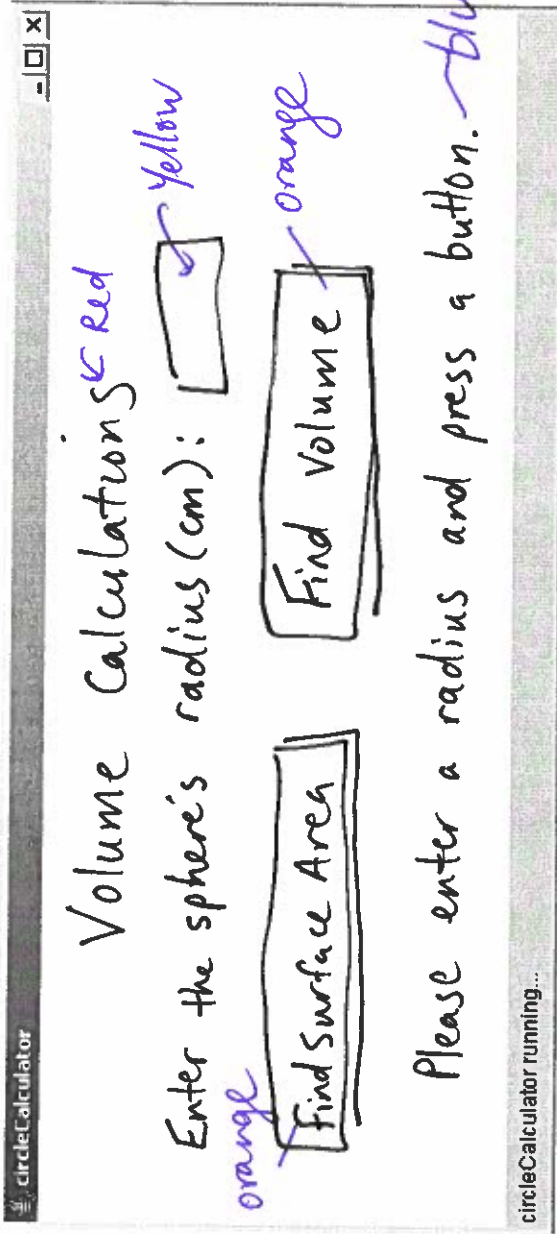
8

* don't forget the picture. It is a JLabel.

5. Fill in the applet's screen using the code shown below.
Be careful to **label the colours**.

4. Look at the below code and circle the correct answer:

- T (F) a. You should save this applet as Calculator.java.
- T (F) b. There is a createImageIcon method in this program.
- T (F) c. An int is a widget.
- T (F) d. An accessor used in the program is getText.
- T (F) e. There are 7 widgets on the screen. *6 widgets*
- T (F) f. setText is a mutator method.
- T (F) g. The output widget is globally declared.
- T (F) h. The widget whose text is changed in actionPerformed is showStatus. *not a widget*



```

import java.applet.*; import java.awt.*; import java.awt.event.*; import javax.swing.*;

public class circleCalculator extends Applet implements ActionListener {
    JTextField radius;
    JLabel output;

    public void init () {
        JButton b1 = new JButton ("Find Surface Area");
        b1.addActionListener (this);
        b1.setActionCommand ("calc1");
        b1.setBackground (Color.Orange);
        JButton b2 = new JButton ("Find Volume");
        b2.addActionListener (this);
        b2.setActionCommand ("calc2");
        b2.setBackground (Color.Orange);

        JLabel title = new JLabel ("Volume Calculations");
        title.setFont (new Font ("Ravie", Font.PLAIN, 20));
        title.setForeground (Color.Red);
        JLabel n = new JLabel ("Enter the sphere's radius (cm):");
        radius = new JTextField (6);
        radius.setBackground (Color.Yellow);
        output = new JLabel ("Please enter a radius and press a button.");
        output.setForeground (Color.Blue);
    }

    public void actionPerformed (ActionEvent e) {
        int r = Integer.parseInt (radius.getText ());
        if (r <= 0)
            output.setText ("Enter a positive radius.");
        else if (r > 10000)
            output.setText ("Enter a smaller number.");
        else if (e.getActionCommand ().equals ("calc2")) {
            double v = 4.0 / 3.0 * Math.PI * r * r * r;
            output.setText ("The volume is " + v + ".");
        }
        else if (e.getActionCommand ().equals ("calc1")) {
            double sa = 4.0 * Math.PI * r * r;
            output.setText ("The surface area is " + sa + ".");
        }
        showStatus ("Thank you.");
    }
}
    
```

add (title);
add (n);
add (radius);
add (b1);
add (b2);
add (output);

add order is important

accessor

colours. don't forget to label.

6. Code the init method of the Tulip applet.

/10

```
import java.awt.*; import javax.swing.*; import java.applet.Applet; import java.awt.event.*;
public class Tulip extends Applet implements ActionListener
{
```

JTextField num;

//The other global variable:

JLabel result;

```
public void init ()
{
    resize (300, 100);
```

//The first label: (Font is Arial, Font.BOLD and 30 pt)

JLabel title = new JLabel ("Tulip Bulb Cost");

title.setFont(new Font ("Arial", Font.BOLD, 30));

//The prompt

JLabel pmt = new JLabel ("#Tulip Bulbs:");

//The textfield:

num = new JTextField (3);

//The button: (Black background, white writing)

JButton b = new JButton ("Total Cost?");

b.setBackground(Color.Black);

b.setForeground(Color.White);

b.addActionListener(this);

b.setActionCommand("cost calc");

//The last label:

result = new JLabel ("Unknown");

//add the widgets:

add(title);

add(pmt);

add(num);

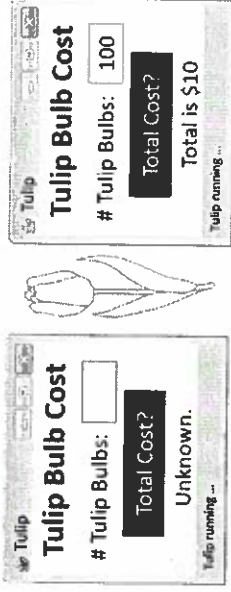
add(b);

add(result);

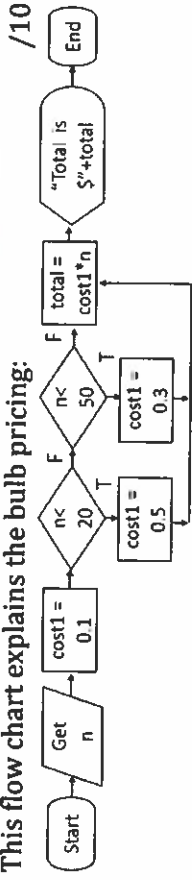
} //init

Application

7. Code the Tulip applet's actionPerformed method.



This flow chart explains the bulb pricing:



```
public void actionPerformed (ActionEvent e)
{
    int n = Integer.parseInt(NUM).getText();
```

double cost1 = 0.1;

if(n < 20)

cost1 = 0.5;

else if (n < 50)

cost1 = 0.3;

double total = cost1 * n;

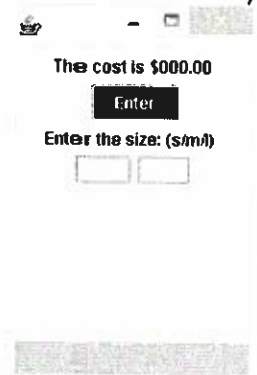
result.setText("Total is \$ " + total);

```
} //actionPerformed
} //Applet
```

Communication

8. Outline one way to achieve each aspect of user-interface design in the applet shown..

(a) Visually appealing	- Needs pictures - Not Resized well (Blank space)
(b) Clear Instructions	- Needs title - 2nd textfield needs prompt.
(c) Restricts Input	- use buttons for s/m/L instead of JTextField
(d) Widget Arrangement	- put widgets in order users will need them. (Button moves to end)
(e) Error Handling	- Reset button would be helpful



/5

9. Which colour is created by each line of code?

- (a) new Color(255, 255, 0)
- (b) new Color(0, 255, 0)
- (c) new Color(255, 0, 255)
- (c) new Color(255, 255, 255)
- (d) new Color(0, 0, 0)
- (e) new Color(0, 255, 255)

Yellow
Green
Magenta
White
Black
Cyan

/6

10. Fill in the following about applets.

set	(a) What is the word that signals a mutator?
new	(b) What is the word that signals a constructor?
get	(c) What is the word that signals an accessor?
declare	(d) What is the term for allocating memory for a variable?
construction	(e) What is the term for setting up RAM for a widget to use?
mutation	(f) What is the term for changing a widget's RAM?
Command Line Interface	(g) What does CLI abbreviate?
Graphic User Interface	(h) What does GUI abbreviate?
1	(i) How many times does init run?
as often as button clicked	(j) How many times does actionPerformed run?

/10

11. What can Denise Melanson's chemotherapy pump teach us about user centric design? (2 sentences) /2

User centric design uses if-statements to guard against mistakes. Preventing errors is important because everyone has "off-days" and incorrect data-entry can cause serious problems. For example, a tired, overworked nurse accidentally entered 4 days of medication as 4 hours on Melanson's chemo pump with tragic results.

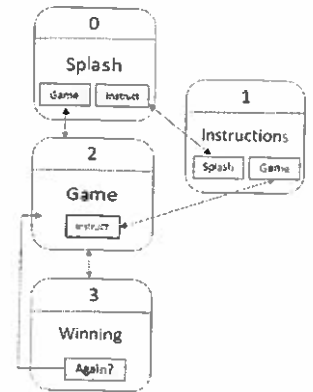
12. Why are ActionListener useful? (2 sentences) /2

ActionListeners are code that "watches" over a button to monitor if it is clicked. When it is clicked, the ActionListener calls ActionPerformed, passing it an ActionCommand so the applet can be updated. Thus, actionlisteners are critical in making an applet responsive to user clicks.

Thinking

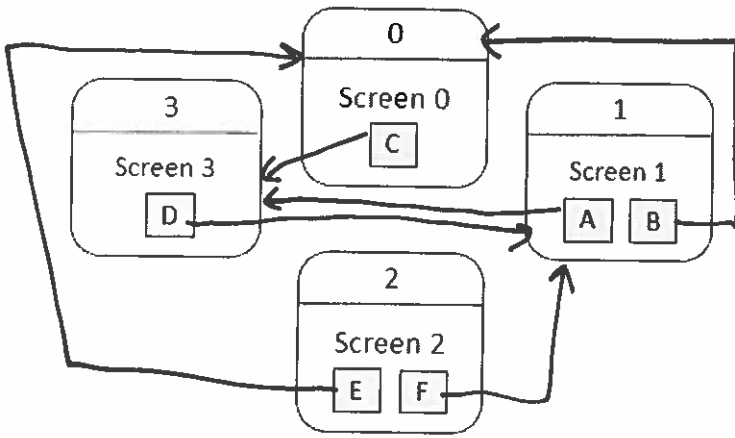
13. Answer the following true or false questions about this screen flow diagram. /8

- F a) The lines connecting the screens should be arrows.
- F b) There are 4 screens in the diagram.
- T c) Return types are labelled beside the lines in screen flow diagrams.
- T d) The diagram shows a lot of detail about formatting that will appear in the final program.
- T e) On a screen flow chart, output is indicated with a pencil shape.
- T f) All screens must have an arrow coming out of them.
- T g) All screens require a name and a screen number.
- T h) A button can have multiple arrows coming out of it.



14. Fill in the arrows on the screen flow diagram using the actionPerformed method. /3

Lines should not cross. Lines should not have curves, only line segments.



```
public void actionPerformed (ActionEvent e)
{ //moves between the screens
  if (e.getActionCommand ().equals ("A"))
    cdLayout.show (p_card, "3");
  else if (e.getActionCommand ().equals ("B"))
    cdLayout.show (p_card, "0");
  else if (e.getActionCommand ().equals ("C"))
    cdLayout.show (p_card, "3");
  else if (e.getActionCommand ().equals ("D"))
    cdLayout.show (p_card, "1");
  else if (e.getActionCommand ().equals ("E"))
    cdLayout.show (p_card, "0");
  else if (e.getActionCommand ().equals ("F"))
    cdLayout.show (p_card, "1");
}
```

15. Where do each of these lines of code go? Circle the appropriate place.

/6

- | | | | | |
|--|-----------|--------|------|-----------------|
| a) import java.applet.Applet; | libraries | global | init | actionPerformed |
| b) add(b1); | libraries | global | init | actionPerformed |
| c) JLabel result; | libraries | global | init | actionPerformed |
| d) result.setText ("Diameter = " + (r * 2)); | libraries | global | init | actionPerformed |
| e) radius = new JTextField (6); | libraries | global | init | actionPerformed |
| f) int r = Integer.parseInt (radius.getText ()); | libraries | global | init | actionPerformed |

16. Circle and correct 5 errors in this code.

/5

```

    JButton start = new JButton("Enter");
    start.addActionListener(this);
    start.setActionCommand(Enter whatever);
    start.setPreferredSize (new Dimension (250, 20));
    start.setBackground(Color.black);
    start.setForeground(Color.black);
    add(start start);
  
```

Annotations: ① new, ② whatever, ③ ;, ④ one needs to be different so you can read it

17. Write the actionPerformed method for this applet. It took Ms. Gorski 4 lines (you may use more).

/6

The textfields are global, they are named x and y.
The result at the bottom is also global, it is called result.

```
public void actionPerformed (ActionEvent e)
```

```

    int x2 = Integer.parseInt (x.getText ());
    int y2 = Integer.parseInt (y.getText ());
    if (y2 == (2 * x2 + 1))
        result.setText ("Yes. It is.");
    else
        result.setText ("No. It is not.");
  
```

For the 4 lines ... combine
1st 3 lines ... keep end

```
if (Integer.parseInt (y.getText ()) == (2 * Integer.parseInt (x.getText ()) + 1))
```

```
} //actionPerformed
```

