Unit 4 – ICS3U0 – Java Applets

Sample Test: Thursday May 1, 2025

Name: Gorski

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

Knowledge

1. Identify the following code that would appear in an applet. /8

(a) Name 2 types of widgets.

(b) Name 2 accessors.

(c) Identify 2 distinct lines of code than can be used in init but not in ActionPerformed.

(d) Identify 2 lines of code that are needed to make a button clickable.

(e) Identify a library.

(f) Identify 2 mutators.

JLabel	JTextField
getText	getActionCommand
add(title)	resize(200, 200)
setActionCommand	addActionListener
javax.Swing.*	
setText	setIcon

JButton

- new
- setAC
- addAc

- setFont
- setFore Ground
- setBack Ground etc.

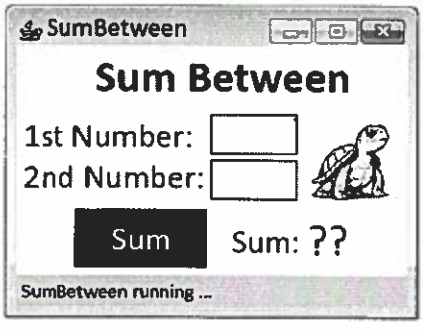
2. Fill in the following pieces of code: /3

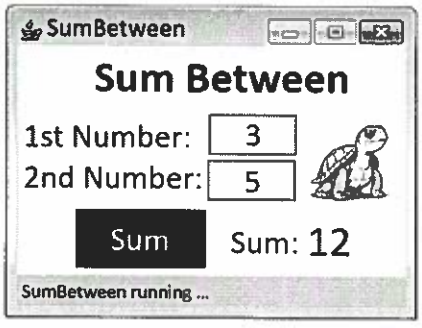
(a) Write "Welcome" in the status bar. showStatus("Welcome");

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

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

3. In the **applet below**, identify the how many of each of the following appear: /4



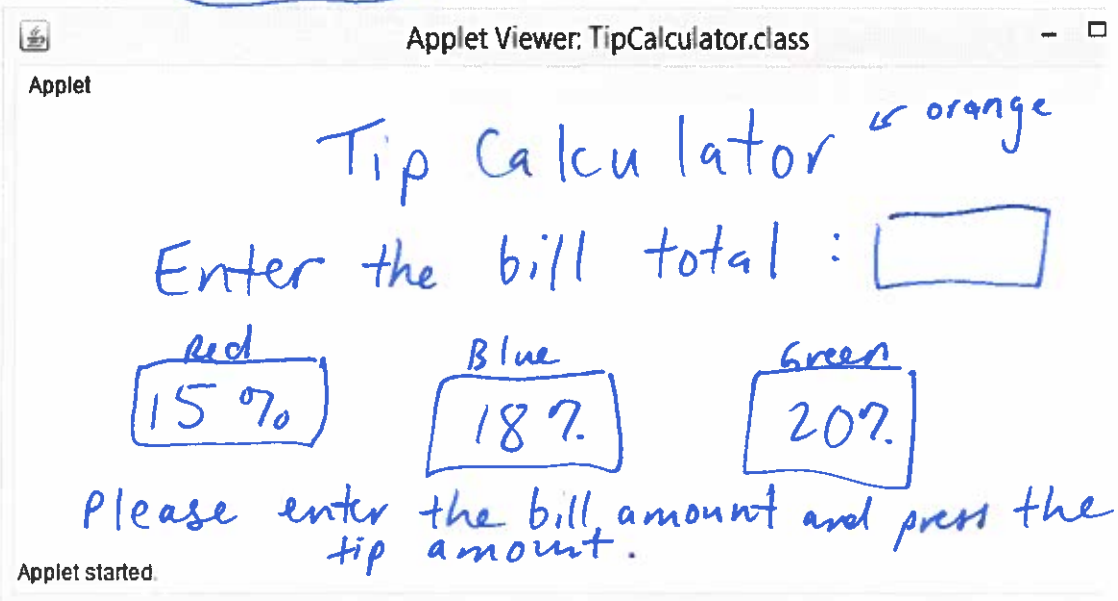


(a)	Action Listeners	1
(b)	Global widgets	3
(c)	JLabels	5 or: 6
(d)	Total widgets	8 9

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

T F a) There are 7 widgets added to the screen.
T F b) There are 3 `actionListeners` in this code, one for each button.
T F c) An `actionCommand` is a widget.
T F d) The `actionCommand` of the good button is 20. it is 18
T F e) The widget whose text is changed in `actionPerformed` is `bill`. tip is
T F f) There are two methods in the program: `init`, `actionPerformed`. no createImageIcon
T F g) All `JTextFields` are globally declared.
T F h) You should save this applet as `TipCalc.java`. TipCalculator.java

5. Fill in the applet's ~~screen using the~~ code shown below. Don't forget.
Be careful to **label the colours.** /12




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

public class TipCalculator extends Applet implements ActionListener{
    JTextField bill;
    JLabel tip;

    public void init (){
         resize (600, 200);
         setBackground(Color.white);
         JLabel title = new JLabel("Tip Calculator");
         title.setFont(new Font("Ravie", Font.PLAIN, 30));
         title.setForeground(Color.orange);
         JLabel ins = new JLabel("Enter the bill total: ");
         bill=new JTextField(5);
         JButton moderate = new JButton("15%");
         moderate.setActionCommand("15");
         moderate.addActionListener(this);
         moderate.setBackground(Color.red);
         JButton good = new JButton("18%");
         good.setActionCommand("18");
         good.addActionListener(this);
         good.setBackground(Color.blue);
         JButton great = new JButton("20%");
         great.setActionCommand("20");
         great.addActionListener(this);
         great.setBackground(Color.green);
         tip=new JLabel("Please enter the bill amount and press the tip amount.");
         add(title);
         add(ins);
         add(bill);
         add(moderate);
         add(good);
         add(great);
         add(tip);
    }
    public void actionPerformed (ActionEvent e) {
         double amt = Double.parseDouble(bill.getText());

         if(e.getActionCommand().equals("15"))
                  amt=amt*1.15;
         else if(e.getActionCommand().equals("18"))
                  amt=amt*1.18;
         else
                  amt=amt*1.20;

         tip.setText("The bill total (including the tip) is $"+amt);
         showStatus("Have a nice day!");
    }
}
```

6. Code the init method of the Turtle applet. /10

```
import java.awt.*; import javax.swing.*, import java.applet.Applet; import java.awt.event.*;
public class Turtle extends Applet implements ActionListener
{
    JTextField age;

//The other global variable:
    JLabel  (X)  ;
    public void init ()
    {  resize (300, 100);
//The first label: (Font is Arial, Font.BOLD and 30 pt)
    JLabel title = new JLabel ("Turtle Size");
    title.setFont(new Font("Arial",Font.BOLD,30));
//The prompt
    JLabel pmt = new JLabel ("Age(years):");
//The textfield:
    age = new JTextField (8);
//The button: (Black background, white writing)
    JButton b = new JButton ("Average Size?");
    b.setBackground(Color.black);
    b.setForeground(Color.white);
    b.addActionListener(this);
    b.setActionCommand("(Y)");
//The last label:
    (X) = new JLabel ("Unknown.");
//add the widgets:
    add(title);
    add(pmt);
    add(age);
    add(b);
    add((X));
} //init
```

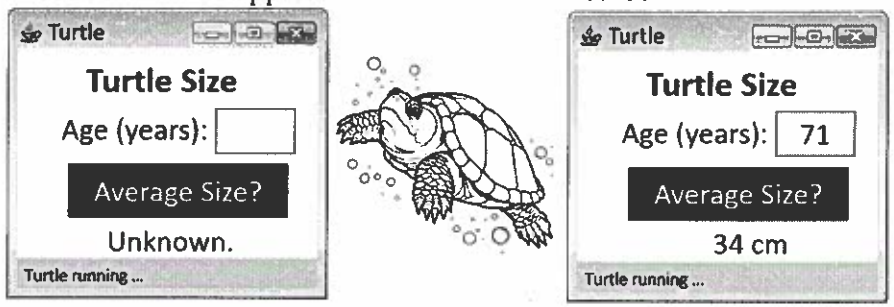
can be anything but must match all positions with (X)

can choose

(X) can't be a or age; they are already used.

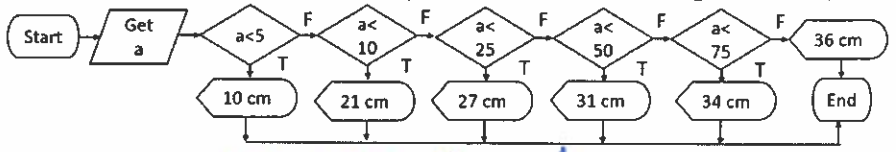
Application

7. Code the Turtle applet's actionPerformed method.



Professor Ronald Brooks of the University of Guelph has been researching the relationship between turtle size and age for over 50 years in Algonquin Park. He is probably the leading turtle researcher in the world.

This flow chart shows the average size for each turtle age: /10

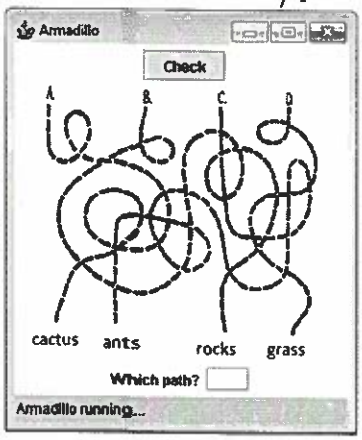



```
public void ActionPerformed (ActionEvent e)
{
   int a = Integer.parseInt(age.getText());
   if( a<5 )
        (X).setText("10cm");
   else if( a<10 )
        (X).setText("21cm");
   else if( a<25 )
        (X).setText("27cm");
   else if( a<50 )
        (X).setText("31cm");
   else if( a<75 )
        (X).setText("34cm");
   else
        (X).setText("36cm");
 } //actionPerformed
} //Applet
```

Communication

8. Outline one way to achieve each aspect of user-interface design in the applet shown. /4

(a) Clear Instructions	- Add an explanation of what to do. (instructions)
(b) Restricts Input	- Use JLabels instead of a JTextfield
(c) Widget Arrangement	- Arrange widgets in order they will be used (move Button last)
(d) Error Handling	- Add a reset/undo Button



9. Which colour is created by each line of code? Use the proper colour names. /6

(a) `new Color(0,0,255)`	Blue	(d) `new Color(0,255,255)`	Cyan
(b) `new Color(255,0,255)`	Magenta	(e) `new Color(255,255,0)`	yellow
(c) `new Color(255,255,255)`	White	(f) `new Color(0,0,0)`	Black

10. Fill in the following about applets. /11

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?
when button pressed	(j) When does ActionPerformed run?
widget	(k) A term for a piece of a GUI. It can be seen, typed in or clicked on.

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

Her pump allowed a tired, overworked nurse to accidentally enter 4 days of medication for 4 hours. User centric design uses if-statements to guard against mistakes. This prevents series errors of the type that occurred to Denise Melanson.

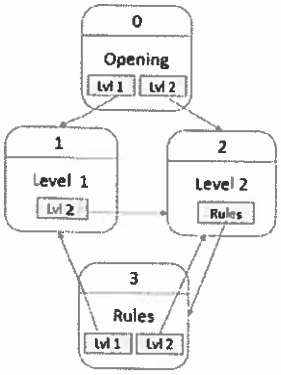
12. Why are actionListeners useful? (2 sentences) /2

ActionListeners are code that "watch" over a button to monitor if it is clicked. When the button is clicked, the ActionListener activates ActionPerformed so that the applet responds appropriately.

Thinking

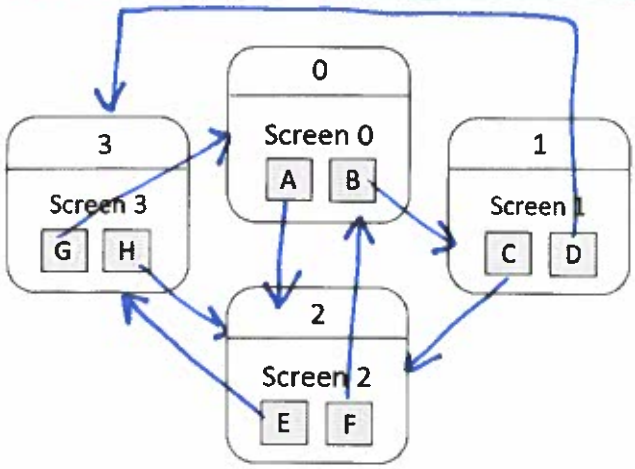
13. Answer the following true or false questions about this screen flow diagram. /8

T (F) a) The first screen that will appear is the Level 1 screen.
T (F) b) The Level 2 screen can access 2 screens directly.
(T) F c) The Level 1 screen can not access the Rules directly.
T (F) d) It is impossible to return to the Level 1 screen once you have left it.
(T) F e) Every screen in a screen flow diagram needs a unique number.
T (F) f) Buttons on screen flow diagrams can have multiple arrows coming out of them.
(T) F g) Screen flow diagrams are used to plan the applet's navigation.
(T) F h) Decision diamonds only appear on flow charts, not on screen flow diagrams.



14. Fill in the arrows on the screen flow diagram using the actionPerformed method. Lines should not cross. Lines should be straight, not curved. /3

Note. Also note




```
public void actionPerformed (ActionEvent e)
{ //moves between the screens
  if (e.getActionCommand ().equals ("A"))
      cdLayout.show (p_card, "2");
  else if (e.getActionCommand ().equals ("B"))
      cdLayout.show (p_card, "1");
  else if (e.getActionCommand ().equals ("C"))
      cdLayout.show (p_card, "2");
  else if (e.getActionCommand ().equals ("D"))
      cdLayout.show (p_card, "3");
  else if (e.getActionCommand ().equals ("E"))
      cdLayout.show (p_card, "3");
  else if (e.getActionCommand ().equals ("F"))
      cdLayout.show (p_card, "0");
  else if (e.getActionCommand ().equals ("G"))
      cdLayout.show (p_card, "0");
  else if (e.getActionCommand ().equals ("H"))
      cdLayout.show (p_card, "2");
}
```

15. Circle **and correct** 5 errors in this code. /5

```
JLabel title = new Label("Applet Sample Test");
title.setPreferredSize(new Dimension (250, 20))
title.setBackground(Colour.black);
title.getForeground(Color.red);
add(title);
```

JLabel ; Can't setBackground on a JLabel, only on JButton set

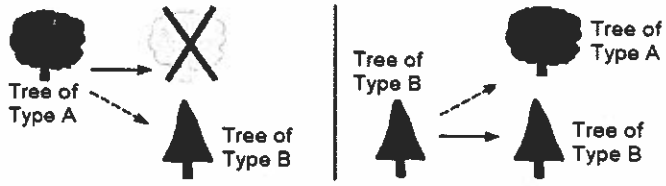
16. Where do each of these lines of code go? Circle the appropriate place. /6

a)	`JTextField bill;`	libraries	global	init	actionPerformed
b)	`bill=new JTextField(5);`	libraries	global	init	actionPerformed
c)	`add(title);`	libraries	global	init	actionPerformed
d)	`double a = Double.parseDouble(bill.getText());`	libraries	global	init	actionPerformed
e)	`tip.setText("The bill total is $"+amt);`	libraries	global	init	actionPerformed
f)	`import javax.swing.*;`	libraries	global	init	actionPerformed

17. In a forest, there are two types of trees. /6

- Type A trees live for only one year, but after this year, they transform into a tree of type B.
- Type B trees live forever and produce a new tree of type A at the end of every year.

These two scenarios can be illustrated, with each arrow representing the transformation at the end of one year.



For example, if we start with one type A tree, after one year there will be one type B tree in the forest.

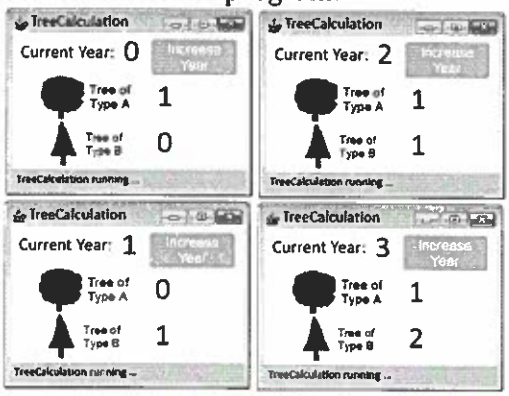
Similarly, if we start with one B tree, there will be one type A tree and one type B tree in the forest after one year.

	0	1	2	3	4	5	6	7	8	9	10
	1	0	1	1	2	3	5	8	13	21	34
	0	1	1	2	3	5	8	13	21	34	55

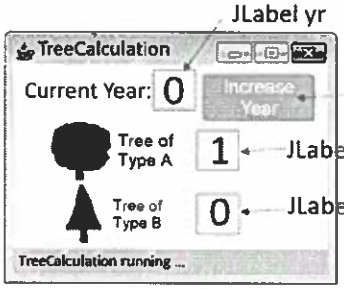
If we start with one type A tree, **after 10 years**, how many type A and B trees in the forest? **(Circle the answer)**

(A) 34 A trees, 20 B trees (B) 54 A trees, 144 B trees (C) 34 A trees, 55 B trees (D) 121 A trees, 55 B trees

Four clicks of the program:



Global Variables:




```
int A = 1; int B = 0;
int year = 0;
```

Write the actionPerformed method for this program.

```
public void actionPerformed (ActionEvent e)
{
    int temp = A+B ;
    A = B ;
    B = temp ;
    year ++ ;
    yr.setText( year );
    numA .setText( A );
    numB .setText( B );
}
```