Full Name: _____

Student ID#: _____

CSE 1325 OBJECT-ORIENTED PROGRAMMING

PRACTICE 2 -=#=- Exam #2 –=# 1 M #=– Java -=#=- PRACTICE 2

Instructions

1. Students are allowed pencils, erasers, and beverage only.
2. All books, bags, backpacks, phones, **smart watches**, and other electronics, etc. must be placed along the walls. **Silence all notifications.**
3. PRINT your name and student ID at the top of this page **and every coding sheet**, and verify that you have all pages.
4. **Read every question completely before you start to answer it.** If you have a question, please raise your hand. You may or may not get an answer, but it won't hurt to ask.
5. If you leave the room, you may not return.
6. You are required to SIGN and ABIDE BY the following Honor Pledge for each exam this semester.

Honor Pledge

On my honor, I pledge that I will not attempt to communicate with another student, view another student's work, or view any unauthorized notes or electronic devices during this exam. I understand that the professor and the CSE 1325 Course Curriculum Committee have zero tolerance for cheating of any kind, and that any violation of this pledge or the University honor code will result in an automatic grade of zero for the semester and referral to the Office of Student Conduct for scholastic dishonesty.

Student Signature: _____

WARNING: Questions are on the BACK of this page!

Vocabulary

Write the word or phrase from the Words list below to the left of the definition that it best matches. Each word or phrase is used at most once, but some will not be used. {15 at 2 points each}

Vocabulary

Word	Definition
1	A managed memory technique that tracks the number of references to allocated memory, so that the memory can be freed when the count reaches zero
2	Code for which specified assertions are guaranteed to be true
3	A statement that introduces a name with an associated type into a scope
4	Reuse and extension of fields and method implementations from another class
5	Algorithmically enforceable constraints on the correctness, meaningfulness, and security of input data
6	Defines the inheritance relationships between a set of classes
7	The process of identifying and specifying subclasses from a superclass.
8	A special class member that cleans up when an object is deleted (not supported by Java)
9	Ensuring that a program operates on clean, correct and useful data
10	A subclass replacing its superclass' implementation of a method
11	Specifying a general interface while hiding implementation details
12	The class inheriting members
13	A named scope
14	Bundling data and code into a restricted container
15	A declaration that also fully specifies the entity declared

Word List

Abstraction	Algorithm	Class Hierarchy	Class Library	Constructor
Data Validation	Declaration	Definition	Destructor	Encapsulation
Garbage Collector	Generalization	Inheritance	Invariant	Multiple Inheritance
Namespace	Override	Package	Reference Counter	Shadowing
Specialization	Subclass	Superclass	Validation Rules	

Note: ALL vocabulary words may be on the actual exam!

Multiple Choice

Read the full question and every possible answer. Choose the one best answer for each question and write the corresponding letter in the blank next to the number. {15 at 2 points each}

1. ____ **In a JFrame's BorderLayout, the JMenuBar is added**

   A. Below the PAGE_START area using SUB_PAGE_START

   B. To the PAGE_START area

   C. Above the PAGE_START area using the JFrame.addJMenuBar method

   D. JMenuBar can't be used with a BorderLayout

2. ____ **File Bell.java begins with** `package uta.edu.cse1325; public class Bell`**. In what directory is Bell.java?**

   A. `./uta/edu/cse1325/`

   B. `./uta.edu.cse1325/`

   C. `./cse1325/edu/uta/`

   D. `./package/cse1325.edu.uta/`

3. ____ **Which of the following statements about Swing is TRUE?**

   A. JRadioButton inherits from both the JRadio class and the JButton class

   B. Swing specifies a color using a 4-byte int (for example, 0xFF000000 is red)

   C. Swing uses HTML to specify rich text (for example, `"<b>bold text</b>"`)

   D. The main window for a Swing application inherits from JWindow

4. ____ **Which of the following demonstrates try-with-resources?**

   A. `try (new MP3Reader(file) != null) {`

   B. `try (new Resource(new MP3Reader(file))) {`

   C. `try (MP3Reader mp3 = new MP3Reader(file)) {`

   D. `try (new MP3Reader(file)) {`

5. ____ **To open** `filePath` **for writing, write**

   A. `File f = filePath.open('w');`

   B. `BufferedWriter bw = new BufferedWriter(new FileWriter(filePath));`

   C. `FileWriter fw = new File(filePath, 'w');`

   D. `File f = fopen(filePath, 'w');`

6. ____ **A Java anonymous class**

    A. omits the class name in a declaration, allowing the compiler to generate a random name

    B. is another name for a lambda, to which it is identical

    C. implements and instances an interface or abstract class in place

    D. defines a generic algorithm without specifying the types to which it applies

7. ____ **To read Strings from a text file opened as** `br` **until end of file is reached, write**

    A. `String line; for(int i=0; i<br.size(); ++i) line = br.readLine(i);`

    B. `String[] lines = br.readLines(0, br.size());`

    C. `String line; while((line = br.readLine()) != null)`

    D. Java can't detect end of file - use a special data value like "EOF!" instead

8. ____ **To create a canvas on which to draw lines and shapes,**

    A. Extend JPanel and create the drawing in the constructor

    B. Add an actionListener to a JPanel

    C. Extend JPanel and override paintComponent

    D. Add a drawingListener to a JPanel

9. ____ **Which is a valid lambda function?**

    A. `double lambda(x) {return Math.sin(x);}`

    B. `lambda{return Math.sin(x);}`

    C. `x -> Math.sin(x)`

    D. All of these are invalid

10. ____ **The Swing-provided Graphics object is usually duplicated before modification because**

    A. The supplied Graphics object may be deleted by the garbage collector while repainting the canvas

    B. It is a reference to the system object, so changes would affect other views

    C. Duplicating the Graphics object speeds up repainting the canvas

    D. The supplied Graphics object is read-only

11. ____ **To arrange widgets in a dialog in a row, wrapping to a new row when needed,**

    A. Use the `FlowLayout` manager

    B. Use the `BoxLayout` manager with a `wrapOnFull` submanager

    C. Use the `GridBagLayout` manager and set its `rowWrap` attribute to true

    D. You would need a custom layout manager for something this complex

12. ____ **To update a canvas when the data model (such as an ArrayList of Line objects) changes,**

    A. Call the repaint() method to request that the canvas be repainted

    B. The canvas repaints itself automatically when the data changes

    C. Call paintComponent() directly to repaint the canvas

    D. Construct a new canvas with the repainted image

13. ____ **To define the action to take when JButton X is pressed,**

    A. Call `X.addActionListener` passing an ActionListener implementation as a parameter

    B. Override `X.onButtonClick()`

    C. Poll `X.isClicked()` and respond if true

    D. Implement a `JButtonManager` to handle click responses

14. ____ **Which of these statements about Swing is TRUE?**

    A. Swing programs require a different compiler than command line programs

    B. Swing programs are portable, and run on Linux, Windows, or OS X

    C. Swing programs require installing the Swing library separately from the Java Development Kit

    D. All of these statements are true

15. ____ **On a canvas that is 1000 x 1000 pixels in size, pixel (25, 25) is**

    A. Near the upper right corner

    B. Near the upper left corner

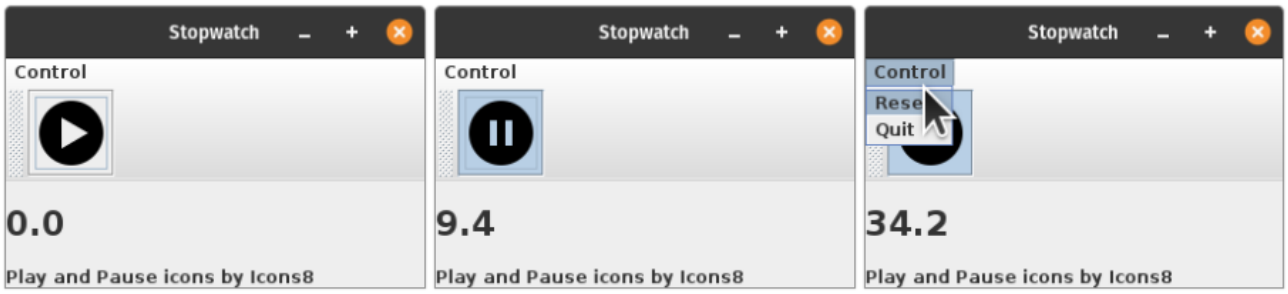
    C. Near the lower right corner

    D. Near the lower left corner

Free Response

Provide clear, concise answers to each question. Each question is *completely independent* of the other questions, and is intended to test your understanding of one aspect of Java and Swing programming. **Write only the code that is requested.** You need NOT write import statements - assume you have already imported whatever you need.

Free Response 1: Consider the following program, a simple stopwatch with elapsed time in seconds in the main window data area. The toggle button causes the time to advance only when selected. Control > Reset deselects the toggle button and sets the elapsed time back to 0. Control > Quit exits the program.



Stopped Running Menu

1.a. Finish the constructor. The comments will guide you as to what is needed.

```
// Imports omitted - assume you have whatever you need
class Stopwatch extends JFrame {
   private JToggleButton start; // Stopwatch runs when selected, stopped otherwise
   private double time = 0;     // Elapsed time in seconds
   private JLabel data;         // Data area in center of the main window

   public Stopwatch() {
       // {2 points} Configure main window - title "Stopwatch", 300x200 pixels,
       //   and exit on close









       // {3 points} Menu Bar with Control > Reset (call onResetClick())
       //   and Control > Quit (dispose of window)









       // Continued on next page
```

```
    // {3 points} Tool Bar with one toggle button named start (icon "start.png",
    //   selected icon "pause.png", and tooltip "Start or stop the stopwatch")
    // No listeners are needed for the toolbar or its toggle button




    // {1 point} Data area is field named data added to the center of the main window




    // The remaining constructor code is provided for you
    updateDisplay();
    setVisible(true);
    new Timer(100, event -> onTimerTick()).start(); // Call onTimerTick() every 100 ms
}
```

1.b {1 point} Code method `onTimerTick()`. If the start button is selected, increment `time` by 0.1, and call `updateDisplay()`.

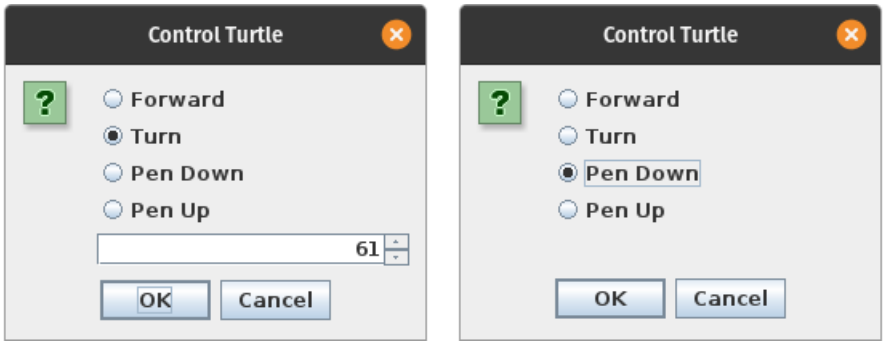
1.c {2 points} Code method `onResetClick()`. Deselect the start button, set `time` to 0, and call `updateDisplay()`.

1d. {2 points} Code method `updateDisplay()`. Set the text for field `data` to the String representation of field `time`, with 1 digit to the right of the decimal point (this is printf code %.1f). Make the text big (this is HTML tag <big>).

1e. {1 point} Code the main method. Instance a new Stopwatch.

Free Response 2. {12 points} The CSE1325 Paint program from Lecture 16 has invoked the turtle graphics shape, and requires the dialog shown below to control the turtle. Write ONLY the method to create, display, and respond to user interaction with this dialog. OR you may use two consecutive dialogs with any appropriate widget each instead of the single dialog shown below **for half credit**.

Four commands may be selected in the dialog: "Forward", "Turn", "Pen Up", and "Pen Down". For Forward and Turn, an integer distance or angle (respectively) is also required.



Select the command using 4 JRadioButton widgets in the same RadioGroup. Hint: Fewer lines of code are needed to create the radio buttons in a loop from the `String[] commands` array. You may assume a `private int command` field for use by your listeners.

Specify the integer distance and angle using a JSpinner with a SpinnerNumberModel of range 0 to 359 and a step of 1.

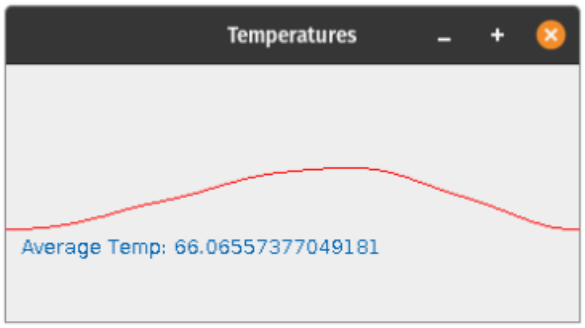
When the "Pen Up" or "Pen Down" command is selected, make the widget for selecting an integer not visible. When the "Forward" or "Turn" command is selected, make the widget for selecting an integer visible. This will require a listener.

When the OK button is clicked, print the command to the console. The two example dialogs above would print "Turn 61" and "Pen Down", respectively. When the Cancel button or any other button is clicked, close the dialog and print nothing.

Your dialog need NOT match the above dialog exactly as long as the functionality is correct and you use the specified widgets. You may use JOptionPane or extend JDialog as you please. Your layout and (if any) icon selection may vary from that shown.

```
public void onControlTurtleClick() {
    String[] commands = {"Forward", "Turn", "Pen Down", "Pen Up"};
```

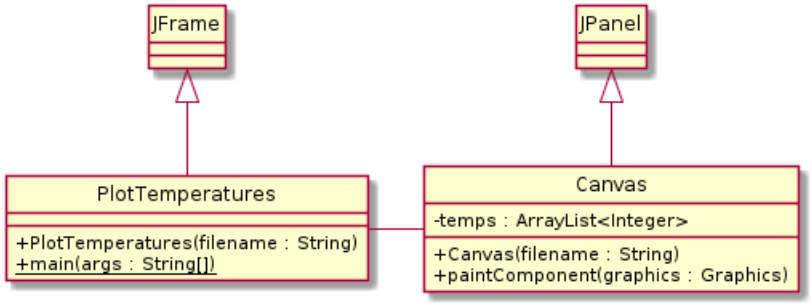
Free Response 3. Consider the following Java Swing main window from class PlotTemperatures.




```
45.7
45.75
45.75
45.75
45.8
45.85
45.85
45.9
```

The data to be plotted, the average temperature at UTA for each day of the year, comes from the text file specified as the first command line argument. The first few lines of this file are to the right of the window.

The class diagram for this program is below.



You will not write class PlotTemperatures, as you already wrote a main window in a previous question.

3.a {5 points} Write the declaration and constructor for class Canvas (you may omit imports). Open the file specified by the parameter for buffered reading using try-with-resources, displaying an error dialog with the message "Unable to read *filename*" (where *filename* is the parameter) on a read error. Otherwise, read each line, convert it to a double, round to the nearest integer, and append to field `temps`. (Note: `Math.round(double)` returns a *long* integer that is the rounded value of its parameter.)

3.b Override JPanel's `paintComponent` method, ensuring a compiler error is generated if no superclass declares this method. In this method:

- {1 point} Call the same method in the superclass, then duplicate and cast the Graphics object passed as its parameter into a Graphics2D object.

- {4 points} Set the color to red. Iterate over the temps ArrayList, drawing a line from point (i-1, 150-temps[i-1]) to point (i, 150-temps[i]) to plot the data. Note that temps is an ArrayList, *not* an array.

- {3 points} Calculate the average temperature from ArrayList temps (this may be as part of the previous loop or in a separate loop). Then change the color to UTA Blue (RGB color 0064B1 in hex) and print "Average Temp: " followed by the average temperature starting at point (10, 120).

Bonus 1: {Up to 2 points} *In one sentence*, define the Principle of Least Astonishment.

Bonus 2: {Up to 3 points} *In one sentence*, define the Second System Effect and give one example of its effect on a real product.

Bonus 3: {Up to 15 points: 3 points per unique dialog} Question 2 asked you to create a dialog with radio buttons (for the turtle command) and a spinner (for distance or angle). Now write up to 5 other dialogs offering combinations of other widgets: (1) The turtle command may be specified radio buttons (as above) or a combo box, and (2) distance or angle may be specified with a spinner (as above), a text box converted to int, or a slider. Your options are radio buttons / text box, radio buttons / slider, combo box / spinner, combo box / text box, and combo box / slider.