CS102 – Algorithms and Programming II
Programming Assignment 4
Fall 2023

ATTENTION:

- Compress all of the Java program source files (.java) files into a single zip file.
- The name of the zip file should follow the below convention:
  CS102_Sec1_Asgn4_YourSurname_YourName.zip
- Replace the variables “Sec1”, “YourSurname” and “YourName” with your actual section, surname, and name.
- You may ask questions on Moodle and during your section’s lab.
- Upload the above zip file to Moodle by the deadline (if not, significant points will be taken off). You will get a chance to update and improve your solution by consulting with the TAs and tutors during your section’s lab.

GRADING WARNING:

- Please read the grading criteria provided on Moodle. The work must be done individually. Code sharing is strictly forbidden. We are using sophisticated tools to check the code similarities. The Honor Code specifies what you can and cannot do. Breaking the rules will result in disciplinary action.

Four Seasons of a Tree

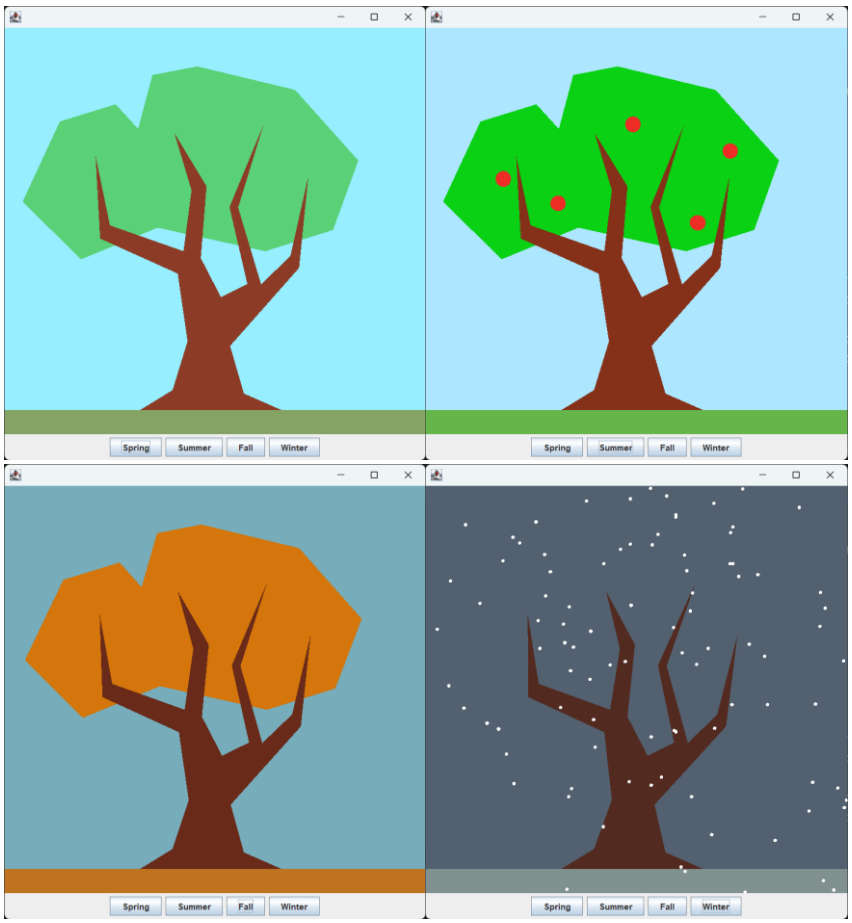
You will implement a program with a Graphical User Interface to animate the four seasons of a tree. Your frame should include four buttons to switch the seasons, where each season has certain graphical elements:

- In spring, there is a weak wind, and the colors are mostly bright; the tree has green leaves.
- In summer, there is no wind, and colors are brighter; the tree has green leaves and red fruits.
- In fall, there is a strong wind, and colors are mostly brown; the tree has orange-brown leaves.
- In winter, there is a stronger wind, and colors are dull; the tree has no leaves, and the weather is snowing.

Implement a **TreePanel** class that extends JPanel. You will overwrite **void paintComponent (Graphics g)** method of this class to paint the content of the panel to draw the scene with the tree. Do not forget to first call **super.paintComponent(g)** so that the panel's background is painted correctly. Keep an int variable in TreePanel that indicates the current season. Include any necessary methods and variables to paint the scenery based on the current season. You will also need a method to change the current season.

You may draw your tree using the fillRect or fillPolygon methods. You can draw the fruits and snowflakes using the fillOval method. Note that the positions of snowflakes should change with time. To this end, you can include a Timer in your TreePanel (import javax.swing.Timer). The timer needs an ActionListener (import java.awt.event.ActionListener), which would be given as a parameter to its constructor. Your TreePanel can implement the ActionListener

interface, which requires you to override its actionPerformed(ActionEvent e) method. Once the timer is run, it will call the actionPerformed method in a repeated manner with a delay; you should use this method for moving the snowflake positions to create a snowing effect. Keep arrays for the x and y positions of snowflakes. You can keep the x and y speeds of the snowflakes in a separate array. In the beginning, initialize a random speed for each snowflake; note that to have them falling, the y speed should be positive. Reset the position of the snowflakes that reach the bottom, probably with a new speed. Each time actionPerformed is called, update the snowflake positions and repaint the panel to have an animated snow. The following screenshots are examples for each season.



You will implement the effect of the wind in a similar manner. Use the same Timer and actionPerformed method to slightly change the positions of the tree's branches each time the actionPerformed method is called. You can keep a shift amount in the TreePanel class and change it gradually to shift the branch positions. In case you use the fillPolygon method to draw your tree, you can shift the x positions of the polygon based on wind speed and how small the y coordinates are (for each point of the polygon, the smaller the y value, the higher it is in the tree so it would be influenced by the wind more).

Implement a TreeFrame class to include your TreePanel in it. This class should create and position your season buttons at the bottom of the screen. BorderLayout can be useful for this task (import java.awt.BorderLayout). You will add your TreePanel to BorderLayout.CENTER, and a different JPanel that contains your buttons to BorderLayout.SOUTH. After adding all the panels, call pack() method of the JFrame so that panel sizes are adjusted to fit; otherwise, you

need to manually resize your panels to make them visible. The TreePanel does not contain any other components in it, we paint the tree by overriding the paintComponent method, so we need to set its size correctly so that it is visible on the screen. Do not forget to also set your JFrame's size and to make it visible.

To make your buttons functional, you will add actionListeners to them. Suppose b is a JButton, the following is a shorthand for adding an actionListener to it and overriding its actionPerformed method. The button will trigger the code under its actionPerformed method.

```
b.addActionListener(new ActionListener() {
   @Override
   public void actionPerformed(ActionEvent e) {
      System.out.println("You clicked!");
    }
});
```

For this assignment, go step by step visualizing the results each time. Start by drawing a simple rectangle on the screen first, then make it a tree. Add some buttons to change the colors of the tree. Draw the snowflakes using the positions from the array, then try changing those positions by time.

Preliminary Submission: You will submit an early version of your solution before the final submission. This version should at least include the following:

- The content for summer should be complete (drawing the tree with leaves and fruits).
- Four buttons should be functional (you should at least print a message on the console when we click on different buttons).

You will have time to complete your solution after you submit your preliminary solution. You can consult the TAs and tutors during the lab. Do not forget to make your final submission at the end.

Even if you finish the assignment in the preliminary submission, you should submit for the final submission on Moodle. **Not completing the preliminary submission on time results in a 50% reduction of this assignment's final grade.**