



## Faculty of Science

**Course:** CSCI 2020u – Software System Development & Integration

**Component:** Assignment

**Weight:** 10%

**Deadline:** March 6, 2019 (due by 11:59pm)

### Collaboration Policy

You are permitted to work on this assignment in a team, and submit the results as a team. For this sort of assignment, with an open-ended component, the collaboration between multiple team members can be beneficial. Between groups, however, please limit the discussion to the level of general strategy (not code). Groups of size 2 are recommended. Larger groups will be considered with the proviso that the marker will mark your assignment with higher expectations. In any case, be sure that all members of the team fully understand all code, otherwise they will miss intended learning objectives, which may be a considerable disadvantage at exam time.

### How to Submit

You will maintain a **git repository** for this assignment, which is a public repository. To submit the assignment, create a single file 'README.txt' that contains instructions on how to download, compile, and run your codes for each question. A .zip, .7z, or .rar file will not be acceptable. **Also submit this word file (once you complete) into related drop box on Blackboard before deadline.**

**Note:** *Comments are mandatory. Failure to properly document your program will result in a deduction on the marks you receive for this (and any other) assignment.*

**<https://github.com/xGarry/2020-Assignment.git>**

### Remember:

You need to complete this file and submit it in related **drop box on Blackboard**, in addition to uploading your codes in your **git repository**, before deadline.

# Question 1: Displaying Three Cards

## Problem Description:

Display a frame that contains three labels. Each label displays a card, as shown in the figure below. The card image files are named 1.png, 2.png, ..., 54.png and stored in the image/card directory. All three cards are distinct and selected randomly.

The image icons can be found in the attached card folder.



## Your Task:

1. Create three `ImageView` and set their icons using the images.
2. Display three images from 54 image cards randomly.

## Your Code:

Copy-paste your code here:

```
package sample;

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
import javafx.scene.image.ImageView;
import java.util.Random;

public class Q1Cards extends Application {

    public HBox pane = new HBox();
    String source = "file:///C:/Users/Garry/Documents/Assignment1/Cards/"; // destination to the images
    @Override // Override the start method in the Application class
    public void start(Stage primaryStage) {

        Random rand = new Random();

        for(int i=0;i<=2;i++){
            pane.setAlignment(Pos.CENTER); // positioning to the center

            int n = rand.nextInt(59);
            if(n<55){
                String end = n + ".png";
                imageFix(end);
            }
            if(n==55){
                String end = "b1fh.png";
                imageFix(end);
            }
            if(n==56){
                String end = "b2fh.png";
                imageFix(end);
            }
            if(n==57){
                String end = "b2fv.png";
            }
        }
    }
}
```

```

        imageFix(end);
    }
    if(n==58){
        String end = "backcard.png";
        imageFix(end);
    }
}

Scene scene = new Scene(pane,300,100); // Create a scene and place it in the stage
primaryStage.setTitle("Cards"); // Set the stage title
primaryStage.setScene(scene); // Place the scene in the stage
primaryStage.show(); // Display the stage
}

public void imageFix (String ending){
    String finalo = source + ending; // adding the extension with the destination
    ImageView imageView = new ImageView(finalo); // making a new image view
    pane.getChildren().add(imageView); // adding the image view to the pane
    pane.setSpacing(10); // spacing between each imageview
}

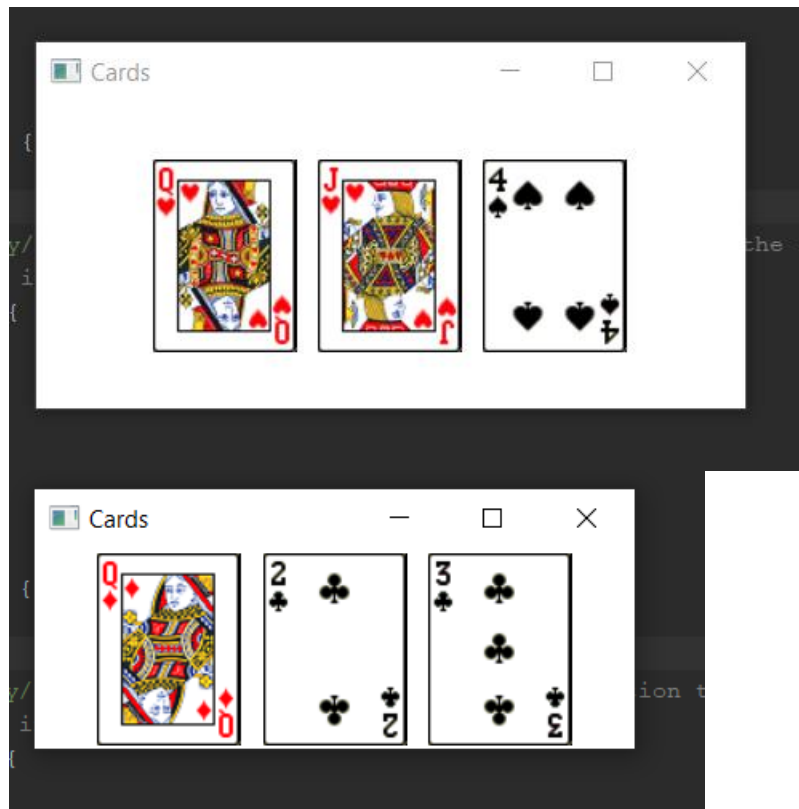
}

public static void main(String[] args) {
    launch(args);
}
}

```

### Screen shots:

Include two screen shots here:

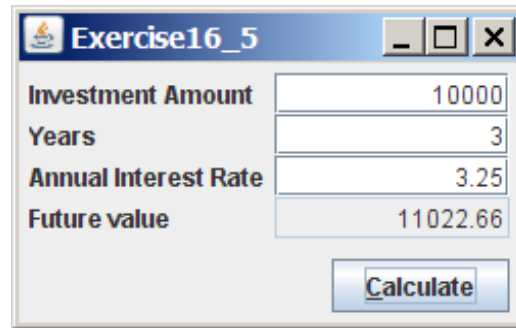


## Question 2: Investment-Value calculator

### Problem Description:

Write a program that calculates the future value of an investment at a given interest rate for a specified number of years. The formula for the calculation is as follows:

$$\text{futureValue} = \text{investmentAmount} * (1 + \text{monthlyInterestRate})^{\text{years} * 12}$$



Investment Amount	10000
Years	3
Annual Interest Rate	3.25
Future value	11022.66

Calculate

### Your Task:

Use text fields for interest rate, investment amount, and years. Display the future amount in a text field when the user clicks the Calculate button, as shown in the figure.

### Your Code:

Copy-paste your code here.

```
package sample;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import javafx.geometry.Pos;
import javafx.geometry.HPos;
import javafx.scene.control.Button;
import java.lang.Math;

public class Q2Calculator extends Application {
    private TextField InvestAmount = new TextField(); // four fields
    private TextField Year = new TextField();
    private TextField AnnualRate = new TextField();
    private TextField result = new TextField();

    public void start(Stage primaryStage) {
        // Create a pane and set its properties
        GridPane pane = new GridPane();

        InvestAmount.setAlignment(Pos.CENTER_RIGHT); // setting the alignment of textfield to the right
        Year.setAlignment(Pos.CENTER_RIGHT);
        AnnualRate.setAlignment(Pos.CENTER_RIGHT);
        result.setAlignment(Pos.CENTER_RIGHT);

        Label labell = new Label("Investment Amount");
```

```

Label label2 = new Label("Years");
Label label3 = new Label("Annual Interest Rate");
Label label4 = new Label("Future Value");

label1.setStyle("-fx-font-weight: bold;"); // making the label bold
label2.setStyle("-fx-font-weight: bold;");
label3.setStyle("-fx-font-weight: bold;");
label4.setStyle("-fx-font-weight: bold;");
result.setEditable(false);

pane.add(label1, 1, 0); // adding labels
pane.add(label2, 1, 1);
pane.add(label3, 1, 2);
pane.add(label4, 1, 3);

Button calculate = new Button("Calculate"); // new button
pane.add(calculate, 2, 4);
pane.setHalignment(calculate, HPos.RIGHT);

calculate.setOnAction(e -> futureValue()); // when the calculate button is pressed, it will output
the future value on to the screen

pane.add(InvestAmount, 2, 0); // adding textfields
pane.add(Year, 2, 1);
pane.add(AnnualRate, 2, 2);
pane.add(result, 2, 3);

pane.setPadding(new Insets(1, 1, 1, 1));
pane.setHgap(5); //vertical spacing
pane.setVgap(5); //horizontal spacing

Scene scene = new Scene(pane, 350, 165); // Create a scene and place it in the stage
primaryStage.setScene(scene); // Place the scene in the stage
primaryStage.show(); // Display the stage
}

private void futureValue() {
    double investmentAmount = Double.parseDouble(InvestAmount.getText()); // getting the input
from textfield and converting it to double
    int years = Integer.parseInt(Year.getText()); // getting the input from textfield and
converting it to integer
    double monthlyInterestRate = Double.parseDouble(AnnualRate.getText()) / 1200;
    result.setText(String.format("$%.2f", (investmentAmount * Math.pow(1 + monthlyInterestRate,
years * 12)))); // calculating the future value with equation
}

public static void main(String[] args) {
    launch(args); // launching the program
}
}

```

## Screen shots:

Include two screen shots here:

The image displays two sequential screenshots of a 'Future Value Calculator' application window. The window has a standard title bar with minimize, maximize, and close buttons. The interface includes four input fields with corresponding labels: 'Investment Amount', 'Years', 'Annual Interest Rate', and 'Future Value'. A 'Calculate' button is positioned at the bottom right of the input area.

**Top Screenshot (Initial State):**

- Investment Amount:
- Years:
- Annual Interest Rate:
- Future Value:
- Calculate:

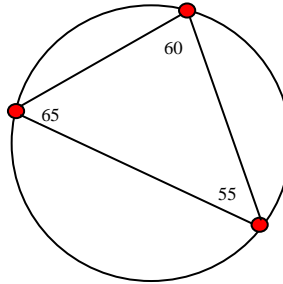
**Bottom Screenshot (After Calculation):**

- Investment Amount:
- Years:
- Annual Interest Rate:
- Future Value:
- Calculate:

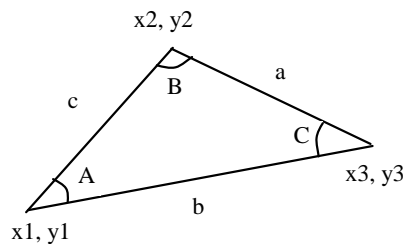
## Question 3: Dragging Points on a Circle

### Problem Description:

Draw a circle with three random points on the circle. Connect the points to form a triangle. Display the angles in the triangle. Use the mouse to drag a point along the perimeter of the circle. As you drag it, the triangle and angles are redisplayed dynamically.



Here is the formula to compute angles:



$$A = \arccos((a * a - b * b - c * c) / (-2 * b * c))$$
$$B = \arccos((b * b - a * a - c * c) / (-2 * a * c))$$
$$C = \arccos((c * c - b * b - a * a) / (-2 * a * b))$$

### Your Code:

Copy-paste your code here:

```
package sample;

import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Line;
import javafx.scene.text.Text;
import javafx.stage.Stage;
import javafx.geometry.Point2D;
import java.util.Random;

public class Q3Circle extends Application {
    //Circle that will determine path of the points
    private double r = 100.0; //radius
    private Circle c = new Circle(250, 250, r);

    Random rand = new Random();
    //Random starting positions for the 3 points
```

```

private int theta1 = rand.nextInt(360);
private int theta2 = rand.nextInt(360);
private int theta3 = rand.nextInt(360);
private double x1 = 250 + r*Math.cos(theta1);
private double y1 = 250 + r*Math.sin(theta1);
private double x2 = 250 + r*Math.cos(theta2);
private double y2 = 250 + r*Math.sin(theta2);
private double x3 = 250 + r*Math.cos(theta3);
private double y3 = 250 + r*Math.sin(theta3);

//Making the points
private double pointRadius = 10.0;
private Circle[] circle = {new Circle(x1, y1, pointRadius), new Circle(x2, y2, pointRadius), new
Circle(x3, y3, pointRadius)};

//Lines connecting the three points
private Line line1 = new Line();
private Line line2 = new Line();
private Line line3 = new Line();
private Text[] text = {new Text(), new Text(), new Text()};

@Override
public void start(Stage primaryStage) {
    Pane pane = new Pane();
    Scene scene = new Scene(pane, 500, 500);
    primaryStage.setResizable(false);

    //set properties of main circle
    c.setFill(null);
    c.setStroke(Color.BLACK);

    //set properties of the 3 points
    circle[0].setFill(Color.BLUE);
    circle[1].setFill(Color.RED);
    circle[2].setFill(Color.GREEN);

    //Keep points on circle while they are dragged, and update their positions and angles
    circle[0].setOnMouseDragged((MouseEvent me) -> {
        Point2D redCenter = new Point2D(c.getCenterX(), c.getCenterY());
        Point2D mouse = new Point2D(me.getX(), me.getY());
        Point2D centerToMouse = mouse.subtract(redCenter);
        Point2D centerToNewPoint = centerToMouse.normalize().multiply(c.getRadius());
        Point2D newPoint = centerToNewPoint.add(redCenter);
        circle[0].setCenterX(newPoint.getX());
        circle[0].setCenterY(newPoint.getY());

        if (circle[0].contains(me.getX(), me.getY())) {
            // Recompute and display angles
            circle[0].setCenterX(me.getX());
            circle[0].setCenterY(me.getY());
            updateLines();
        }
    });

    circle[1].setOnMouseDragged((MouseEvent me) -> {
        Point2D redCenter = new Point2D(c.getCenterX(), c.getCenterY());
        Point2D mouse = new Point2D(me.getX(), me.getY());
        Point2D centerToMouse = mouse.subtract(redCenter);
        Point2D centerToNewPoint = centerToMouse.normalize().multiply(c.getRadius());
        Point2D newPoint = centerToNewPoint.add(redCenter);
        circle[1].setCenterX(newPoint.getX());
        circle[1].setCenterY(newPoint.getY());

        if (circle[1].contains(me.getX(), me.getY())) {
            // Recompute and display angles
            circle[1].setCenterX(me.getX());
            circle[1].setCenterY(me.getY());
            updateLines();
        }
    });

    circle[2].setOnMouseDragged((MouseEvent me) -> {
        Point2D redCenter = new Point2D(c.getCenterX(), c.getCenterY());
        Point2D mouse = new Point2D(me.getX(), me.getY());
        Point2D centerToMouse = mouse.subtract(redCenter);
        Point2D centerToNewPoint = centerToMouse.normalize().multiply(c.getRadius());
        Point2D newPoint = centerToNewPoint.add(redCenter);
        circle[2].setCenterX(newPoint.getX());
        circle[2].setCenterY(newPoint.getY());

        if (circle[2].contains(me.getX(), me.getY())) {
            // Recompute and display angles

```



```

        circle[2].setCenterX(me.getX());
        circle[2].setCenterY(me.getY());
        updateLines();
    }
});

pane.getChildren().addAll(c, circle[0], circle[1], circle[2], line1, line2, line3, text[0],
text[1], text[2]);
primaryStage.setTitle("Question 3");
primaryStage.setScene(scene);
primaryStage.show();
}

//Updates the angles and line positions
private void updateLines() {
    //Find starting and ending positions of all lines
    line1.setStartX(circle[0].getCenterX());
    line1.setStartY(circle[0].getCenterY());
    line1.setEndX(circle[1].getCenterX());
    line1.setEndY(circle[1].getCenterY());
    line2.setStartX(circle[0].getCenterX());
    line2.setStartY(circle[0].getCenterY());
    line2.setEndX(circle[2].getCenterX());
    line2.setEndY(circle[2].getCenterY());
    line3.setStartX(circle[1].getCenterX());
    line3.setStartY(circle[1].getCenterY());
    line3.setEndX(circle[2].getCenterX());
    line3.setEndY(circle[2].getCenterY());

    //Update angles
    double x = new Point2D(circle[2].getCenterX(), circle[2].getCenterY()).
        distance(circle[1].getCenterX(), circle[1].getCenterY());
    double y = new Point2D(circle[2].getCenterX(), circle[2].getCenterY()).
        distance(circle[0].getCenterX(), circle[0].getCenterY());
    double z = new Point2D(circle[1].getCenterX(), circle[1].getCenterY()).
        distance(circle[0].getCenterX(), circle[0].getCenterY());
    double[] angle = new double[3];

    angle[0] = Math.acos((x * x - y * y - z * z) / (-2 * y * z));
    angle[1] = Math.acos((y * y - x * x - z * z) / (-2 * x * z));
    angle[2] = Math.acos((z * z - y * y - x * x) / (-2 * x * y));

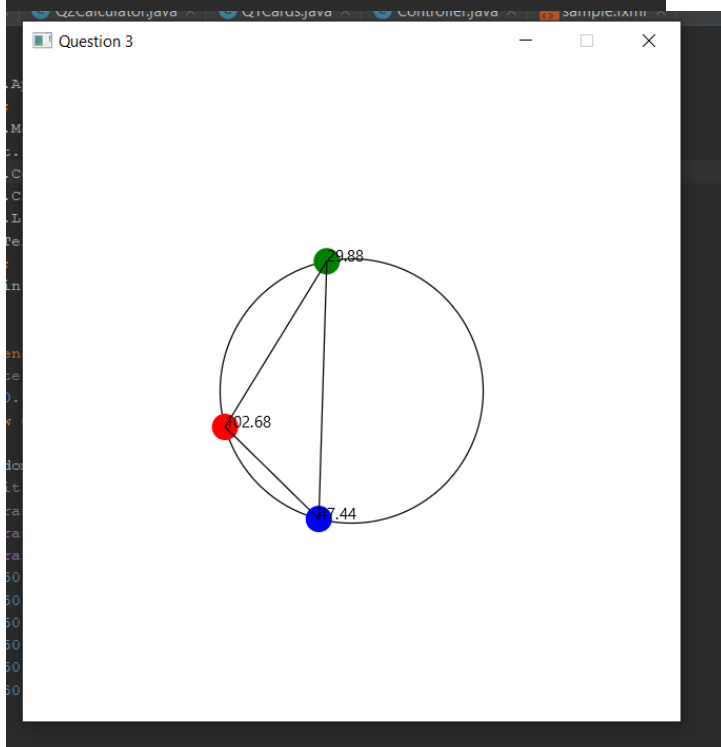
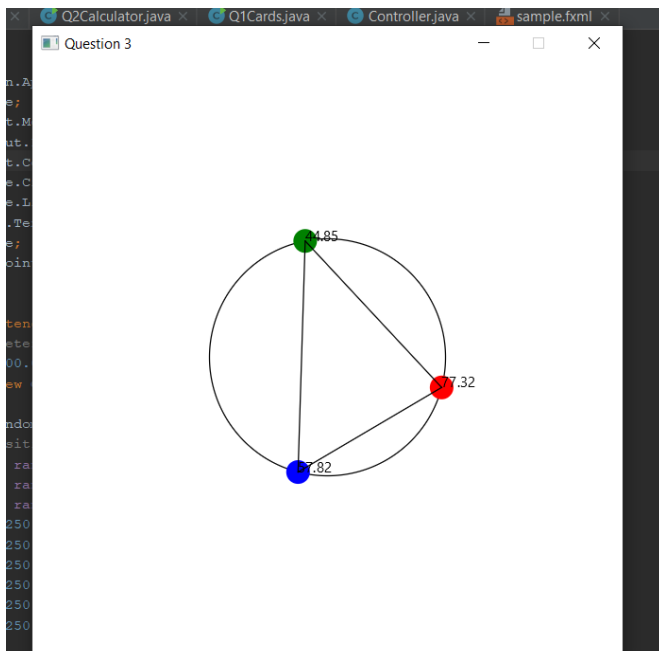
    //Update text positions
    for (int i = 0; i < 3; i++) {
        text[i].setX(circle[i].getCenterX());
        text[i].setY(circle[i].getCenterY());
        text[i].setText(String.format("%.2f", Math.toDegrees(angle[i])));
    }
}

public static void main(String[] args) {
    launch(args);
}
}

```

### Screen shots:

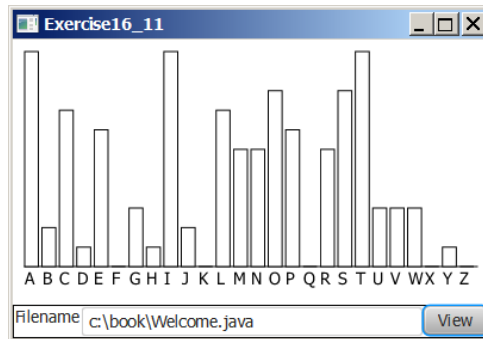
Include two screen shots here:



## Question 4: Histogram

### Problem Description:

Develop a program that displays a histogram to show the occurrences of each letter in a text area. The histogram should show the occurrences of each letter in a text file, as shown in the following figure. Assume that the letters are not case sensitive.



### Your Task:

- Place a pane that will display the histogram in the center of the frame.
- Place a label and a text field in a panel, and put the panel in the south side of the frame. The text file will be entered from this text field.
- Pressing the Enter key on the text field causes the program to count the occurrences of each letter and display the count in a histogram.

### Your Code:

Copy-paste your code here:

```
package sample;

import java.io.File;
import java.io.FileInputStream;
import java.io.IOException;
import java.util.Scanner;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.application.Application;
import javafx.geometry.Insets;
import javafx.scene.Scene;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.layout.VBox;
import javafx.scene.layout.HBox;
import javafx.stage.Stage;
import javafx.geometry.Pos;
import javafx.geometry.HPos;
import javafx.scene.control.Button;
import java.lang.Math;
import javafx.scene.shape.Line;
import javafx.scene.chart.BarChart;
import javafx.scene.chart.CategoryAxis;
import javafx.scene.chart.NumberAxis;
import javafx.scene.chart.XYChart;
import javafx.scene.input.KeyCode;
```

```

import javafx.scene.input.KeyEvent;
import java.io.FileNotFoundException;
import javafx.scene.Node;
import javafx.beans.value.ObservableValue;
import javafx.scene.chart.XYChart.Data;
import javafx.scene.chart.XYChart.Series;

public class Q4Histogram extends Application
{
    VBox vbox = new VBox();
    HBox hbox = new HBox();
    CategoryAxis xAxis = new CategoryAxis(); // making xAxis
    NumberAxis yAxis = new NumberAxis(); // making yAxis
    BarChart<String, Number> barChart = new BarChart<String, Number>(xAxis, yAxis); // Create
a BarChart
    XYChart.Series<String, Number> dataSeries1 = new XYChart.Series<String, Number>(); // making a
series in a bar graph
    TextField fileinput = new TextField(); // textfield for the filename;

    char current;
    int number=0;
    int ascii=65; // used for ascii character

    public void start(Stage primaryStage){

        Label filename = new Label("Filename ");
        fileinput.setPrefWidth(680); // setting a prefix size of textfield
        Button view = new Button("View");
        hbox.getChildren().addAll(filename,fileinput,view); // adding the label, textfield and button
to the pane

        if(fileinput.getText().trim().isEmpty()){ // if the fileinput textfield is empty
            while (ascii <=90){
                char c = (char)ascii; // converting number to character
                dataSeries1.getData().add(new XYChart.Data<String, Number>(String.valueOf(c), 0)); //
making data to plot on the graph
                ascii++;
            }
        }

        view.setOnAction(new EventHandler<ActionEvent>() {
            @Override // Override the handle method
            public void handle(ActionEvent e) {
                action();
            }
        });

        fileinput.setOnKeyPressed(new EventHandler<KeyEvent>() { // if the enter key is pressed in
textfield, update the graph
            @Override
            public void handle(KeyEvent keyEvent) {
                if (keyEvent.getCode() == KeyCode.ENTER) {
                    action();
                }
            }
        });

        changes(); // visual changes to the graph

        barChart.getData().add(dataSeries1); // adding the dataset to the barchart
        vbox.getChildren().add(barChart); // adding the barchart to vbox pane
        hbox.setAlignment(Pos.CENTER);
        vbox.getChildren().add(hbox); // adding hbox to vbox pane

        Scene scene = new Scene(vbox, 800, 800);
        primaryStage.setScene(scene);
        primaryStage.setHeight(400);
        primaryStage.setWidth(800);

        primaryStage.show();
    }

    public void changes(){
        barChart.setVerticalGridLinesVisible(false); // remove the vertical grid lines
        barChart.setHorizontalGridLinesVisible(false); // remove the horizontal grid lines
        barChart.getYAxis().setTickLabelsVisible(false); // remove the y tick labels
        barChart.getYAxis().setOpacity(0); // remove the y line
    }
}

```

```

        barChart.setLegendVisible(false); // remove the legend
    }
    public static void main(String args[]){
        launch(args); // launch the program
    }
    public void action(){
        int ascii=65; // reinitilize the ascii value
        int asciiLower=97;
        try{
            while (ascii <=90 && asciiLower <=122 ){
                char l = (char)asciiLower;
                char c = (char)ascii;
                FileInputStream fis = new FileInputStream(fileinput.getText()); // read in the file as
                fis
                while (fis.available() > 0) {
                    current = (char) fis.read(); // reading in character by character
                    if (current == c || current == l){ // if the character from ascii and the file are
                        same(lowercase and uppercase )
                        number = number+1; // increment the number
                    }

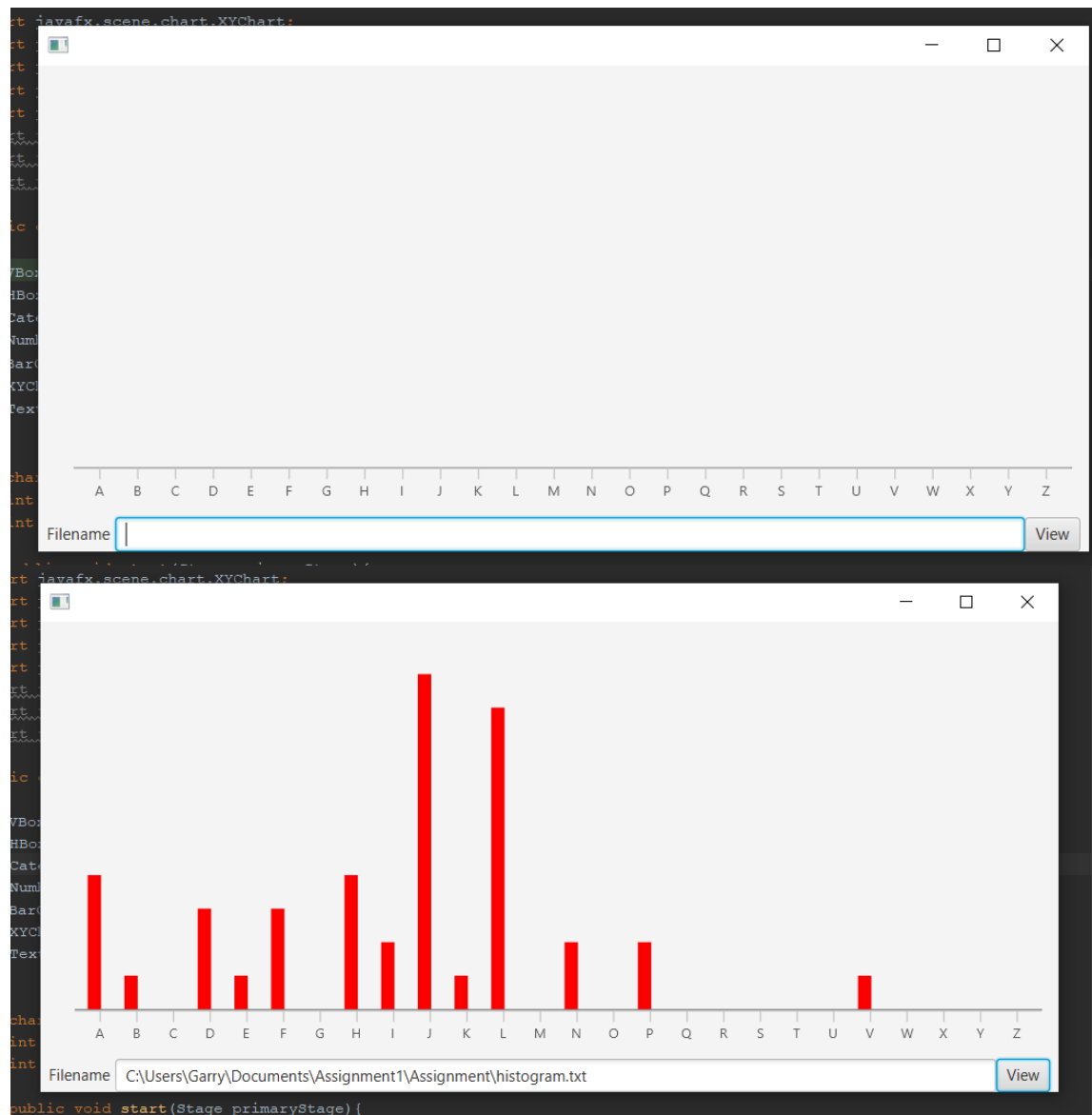
                    dataSeries1.getData().add(new XYChart.Data<String, Number>(String.valueOf(c), number));
                }
                // add the data
                asciiLower++;
                ascii ++;
                number =0;
            }
            for(Node n:barChart.lookupAll(".default-color0.chart-bar")){ // used to change the color of
                the bars
                    n.setStyle("-fx-bar-fill: red");
                }

            }catch(FileNotFoundException fnfe){ // if file doesn't exist, then give an error on the console
                System.out.println(fnfe.getMessage());
            }catch(IOException fefe){
                fefe.printStackTrace();
            }
        }
    }
}

```

### Screen shots:

Include two screen shots here:



**Remember:**

You need to complete this file and submit it in related **drop box on Blackboard**, in addition to uploading your codes in your **git repository**, before deadline.