INFORMATICS INSTITUTE OF TECHNOLOGY

UNIVERSITY OF WESTMINSTER

Informatics Institute of Technology

Department of Computing

(B.Eng.) in Software Engineering

Module: 4COSC0010C Programming Principles II

Module Leader: Mr. Guhanathan Poravi

Summative Assessment

CWK_01

Date of Submission: 13.07.2020

Student ID: 2019554

Student UOW: W1790105

Student Name: Hirun Kodituwakku

Contents

Main Class

In the main class I called the main menu page.

```
package com.hirunz2000;

import javafx.application.Application;
import javafx.stage.Stage;



public class Main extends Application{
    @Override
    public void start(Stage primaryStage) throws Exception {
        //starting the main menu
        primaryStage.setTitle("All in One Finance Calculator");
        primaryStage.setScene(MainMenu.Mainmenu());
        primaryStage.show();

        //setting a fix size for stage
        primaryStage.setMaxHeight(440);
        primaryStage.setMaxWidth(420);
        primaryStage.setMinHeight(440);
        primaryStage.setMinWidth(420);
    }

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

Main Menu Class

Main menu has 4 buttons that leads to four calculators.

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import java.io.FileInputStream;
import java.io.FileNotFoundException;


public class MainMenu {
    public static Scene Mainmenu() {

        //heading
        Label lblheading=new Label("WELCOME TO MY CALC");
        lblheading.setId("heading1");
        lblheading.setFocusTraversable(false);
        lblheading.setPrefSize(250,35);
        lblheading.setLayoutX(95);
        lblheading.setLayoutY(25);

        //button which opens the mortgage calculator
        Button btnMortgage = new Button("Mortgage Calculator");
        btnMortgage.setId("btn1");
        btnMortgage.setFocusTraversable(false);

       // remove the selection when mouse clicked
        btnMortgage.setPrefSize(80,80);
        btnMortgage.setLayoutX(304);
        btnMortgage.setLayoutY(220);
        btnMortgage.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //close the current stage
                Stage stage = (Stage) btnMortgage.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                // set the stage to a fixed size
                primaryStage.setMaxHeight(400);
                primaryStage.setMaxWidth(520);
                primaryStage.setMinHeight(400);
                primaryStage.setMinWidth(520);

                primaryStage.setTitle("All in One Finance Calculator");

primaryStage.setScene(MortgageCalculatorWindow.MortgageCalculator());
```

```
                primaryStage.show();
            }
        });

        Button btnLoan = new Button("Loan Calculator");
        btnLoan.setPrefSize(80,80);
        btnLoan.setFocusTraversable(false);
        // remove the selection when mouse clicked
        btnLoan.setId("btn1");
        btnLoan.setLayoutX(208);
        btnLoan.setLayoutY(220);
        btnLoan.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //close the current stage
                Stage stage = (Stage) btnLoan.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                // set the stage to a fixed size
                primaryStage.setMaxHeight(400);
                primaryStage.setMaxWidth(520);
                primaryStage.setMinHeight(400);
                primaryStage.setMinWidth(520);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(LoanCalculatorWindow.LoanCalculator());
                primaryStage.show();
            }
        });

        Button btnFD = new Button("Fixed Deposit Calculator");
        btnFD.setPrefSize(80,80);
        btnFD.setId("btn1");
        btnFD.setFocusTraversable(false);
        // remove the selection when mouse clicked
        btnFD.setLayoutX(112);
        btnFD.setLayoutY(220);
        btnFD.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //close the current stage
                Stage stage = (Stage) btnLoan.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                // set the stage to a fixed size
                primaryStage.setMaxHeight(400);
                primaryStage.setMaxWidth(520);
                primaryStage.setMinHeight(400);
                primaryStage.setMinWidth(520);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(FixedDeposit.FD());
                primaryStage.show();

            }
        });
```

```
        Button btnFinance = new Button("Savings Calculator");
        btnFinance.setId("btn1");
        btnFinance.setFocusTraversable(false);
        // remove the selection when mouse clicked
        btnFinance.setPrefSize(80,80);
        btnFinance.setLayoutX(16);
        btnFinance.setLayoutY(220);
        btnFinance.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //close the current stage
                Stage stage = (Stage) btnLoan.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                // set the stage to a fixed size
                primaryStage.setMaxHeight(400);
                primaryStage.setMaxWidth(520);
                primaryStage.setMinHeight(400);
                primaryStage.setMinWidth(520);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(SavingsCalculator.savings());
                primaryStage.show();
            }
        });

        Button btnExit = new Button("Exit");
        btnExit.setId("btnExit");
        btnExit.setFocusTraversable(false);
        // remove the selection when mouse clicked
        btnExit.setPrefSize(75,30);
        btnExit.setLayoutX(300);
        btnExit.setLayoutY(350);
        btnExit.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
         //close the current stage and this closes the whole application
                Stage stage = (Stage) btnExit.getScene().getWindow();
                stage.close();
            }
        });
//-----------------------------------------------------------------------------//
//-----------------Adding the image to main menu --------------------------//

        Image image = null;
        try {
            image = new Image(new FileInputStream("src/images/img1.png"));
        } catch (FileNotFoundException e) {
            System.out.println("Warning! Image not found.");
        }
        ImageView imageView = new ImageView(image);

        imageView.setX(65);
        imageView.setY(71);
```

```
        imageView.setFitHeight(140);
        imageView.setFitWidth(270);
        imageView.setId("image");


//-------------------------------------------------------------------------//
// --------------Adding to anchorpane and stage display--------------------//

        AnchorPane root =new AnchorPane();
        root.setId("root");

root.getChildren().addAll(btnExit,btnFD,btnFinance,btnLoan,btnMortgage,lblhea
ding,imageView);
        Scene scene=new Scene(root,400,400);
        scene.getStylesheets().add("css/layout.css");
        return scene;
    }
}
//*************************************************************************
*************************************************************************//
```

Keyboard Class

This class contains the on screen keyboard, which is used by all other classes.

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.Button;
import javafx.scene.control.TextField;
import javafx.scene.layout.GridPane;

public class Keyboard{
    // keyboard
    public static void keyboardDisplay(GridPane keyboardGrid,TextField txt){
        // setting properties for the received gridpane
        keyboardGrid.setPrefSize(141,228);
        keyboardGrid.setHgap(5);
        keyboardGrid.setVgap(5);
        keyboardGrid.setLayoutX(335);
        keyboardGrid.setLayoutY(80);
        keyboardGrid.setId("keyboard");// id for external css

//-------------------- keyboard buttons-----------------------------------//

        Button btnZero=new Button("0");
        btnZero.setPrefSize(45,45);
        btnZero.setFocusTraversable(false);
        btnZero.setId("keyboardButton");
        btnZero.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("0"); //add 0 to the received textfield
                }
            }
        });

        Button btnPeriod =new Button(".");
        btnPeriod.setPrefSize(45,45);
        btnPeriod.setFocusTraversable(false);
        btnPeriod.setId("period");
        btnPeriod.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    if (!txt.getText().contains(".")) {
                        txt.appendText(".");
                   //add period to the received textfield

                    } else {
                   //if there is a period sign already, shows an error message
                        Validations.periodErrorAlert();
                    }
                }
            }
```

```
        });

        Button btnOne =new Button("1");
        btnOne.setPrefSize(45,45);
        btnOne.setFocusTraversable(false);
        btnOne.setId("keyboardButton");
        btnOne.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("1");//add 1 to the received textfield
                }
            }
        });

        Button btnTwo =new Button("2");
        btnTwo.setPrefSize(45,45);
        btnTwo.setFocusTraversable(false);
        btnTwo.setId("keyboardButton");
        btnTwo.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("2");//add 2 to the received textfield
                }
            }
        });
        Button btnThree =new Button("3");
        btnThree.setPrefSize(45,45);
        btnThree.setFocusTraversable(false);
        btnThree.setId("keyboardButton");
        btnThree.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("3");//add 3 to the received textfield
                }
            }
        });

        Button btnFour =new Button("4");
        btnFour.setPrefSize(45,45);
        btnFour.setFocusTraversable(false);
        btnFour.setId("keyboardButton");
        btnFour.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("4");//add 4 to the received textfield
                }
            }
        });

        Button btnFive =new Button("5");
        btnFive.setPrefSize(45,45);
        btnFive.setFocusTraversable(false);
        btnFive.setId("keyboardButton");
```

```
        btnFive.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("5");//add 5 to the received textfield
                }
            }
        });

        Button btnSix =new Button("6");
        btnSix.setPrefSize(45,45);
        btnSix.setFocusTraversable(false);
        btnSix.setId("keyboardButton");
        btnSix.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("6");//add 6 to the received textfield
                }
            }
        });

        Button btnSeven =new Button("7");
        btnSeven.setPrefSize(45,45);
        btnSeven.setFocusTraversable(false);
        btnSeven.setId("keyboardButton");
        btnSeven.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("7");//add 7 to the received textfield
                }
            }
        });

        Button btnEight =new Button("8");
        btnEight.setPrefSize(45,45);
        btnEight.setFocusTraversable(false);
        btnEight.setId("keyboardButton");
        btnEight.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("8");//add 8 to the received textfield
                }
            }
        });

        Button btnNine =new Button("9");
        btnNine.setPrefSize(45,45);
        btnNine.setFocusTraversable(false);
        btnNine.setId("keyboardButton");
```

```
        btnNine.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.appendText("9");//add 9 to the received textfield
                }
            }
        });

        Button btnBackSP =new Button("Backspace");
        btnBackSP.setPrefSize(90,45);
        btnBackSP.setFocusTraversable(false);
        btnBackSP.setId("backSP");
        btnBackSP.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    String str = txt.getText();
                   if (str.length()>0) {
                // deletes only if the string is not null
                        str = str.substring(0, str.length() - 1);
                //make a substring of the textfield string and return
                        txt.setText(str);
                    }
                }
            }
        });

        Button btnClr =new Button("Del");
        btnClr.setPrefSize(45,45);
        btnClr.setFocusTraversable(false);
        btnClr.setId("backSP");// applied  the same id in css
        btnClr.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                if (txt!=null) {
                    txt.setText("");// clears a one textfield completely
                }
            }
        });
//-----------------------add buttons to keyboard grid----------------------//
        keyboardGrid.add(btnBackSP,0,0,2,1);
        keyboardGrid.add(btnSeven,0,1);
        keyboardGrid.add(btnClr,2,0);
        keyboardGrid.add(btnEight,1,1);
        keyboardGrid.add(btnNine,2,1);
        keyboardGrid.add(btnFour,0,2);
        keyboardGrid.add(btnFive,1,2);
        keyboardGrid.add(btnSix,2,2);
        keyboardGrid.add(btnOne,0,3);
        keyboardGrid.add(btnTwo,1,3);
        keyboardGrid.add(btnThree,2,3);
        keyboardGrid.add(btnZero,0,4);
        keyboardGrid.add(btnPeriod,2,4);
    }
}
//*************************************************************************
*************************************************************************//
```

Formulae Class

This class contains all the formulas that I used in this calculator application. I have divided to sections so that it is easy to read. Since there are lots of double values I used default constructor because otherwise I will get constructors with same signature value.

```
package com.hirunz2000;

public class Formulae {

    /*
    A = the future value of the investment/loan, including interest.
    P = the principal investment amount (the initial deposit or loan amount - present value).
    r = annual interest rate.
    n = the number of times that interest is compounded per unit time(this is always monthly for the purpose of this CW).
    t = the time the money is invested or borrowed in years.
    PMT = payments amount per month
    */

    private double A;
    private double P;
    private double r;
    public static final int n=12;
    private double t;
    private double PMT;
    private double FV;

//----------------------------Fixed deposit----------------------------------//

    public double presentValue(double t,double A, double r){
        this.A=A;
        this.r=(r/100)/n;
        this.t=t*n;
        this.P= this.A/Math.pow(1+this.r,this.t);
        return this.P;
    }

    public double futureValue(double t,double P, double r){
        this.P=P;
        this.r=(r/100)/n;
        this.t=t*n;
        this.A= this.P*(Math.pow(1+this.r,this.t));
        return this.A;
    }

    public double interestRate(double t,double P, double A){
        this.P=P;
        this.A=A;
        this.t=t*n;
        this.r= n*(Math.pow((this.A/this.P),(1.0/this.t))-1)*100;

        return this.r;
    }
```

```
    public double timePeriod(double A,double P, double r){
        this.P=P;
        this.r=(r/100)/n;
        this.A=A;
        double t= Math.round(Math.log(this.A/this.P)/(n*Math.log(1+this.r)));
        return t;
    }

//-------------------------------------------------------------------------//
//-------------------------Savings calculator------------------------------//

    public double FV(double PMT,double P,double r, double t){
        this.PMT=PMT;
        this.P=P;
        this.r=(r/100)/n;
        this.t=t*n;
        this.FV=(this.PMT*(Math.pow(1+this.r,this.t)-1)/this.r) +
this.P*Math.pow(1+this.r,this.t);

        return this.FV;
    }

    public double P(double PMT, double A, double r, double t){
        this.PMT=PMT;
        this.A=A;
        this.r=(r/100)/n;
        this.t=t*n;
        this.P=(this.A*this.r-this.PMT*(Math.pow(1+this.r,this.t)-
1))/(this.r*Math.pow(1+this.r,this.t));

        return this.P;
    }

    public double PMT(double P,double A,double r, double t){
        this.P=P;
        this.A=A;
        this.r=(r/100)/n;
        this.t=t*n;
        this.PMT=this.r*(this.A-
this.P*Math.pow(1+this.r,this.t))/(Math.pow(1+this.r,this.t)-1);

        return this.PMT;
    }

    public double time(double P,double A,double r, double PMT){
        this.P=P;
        this.A=A;
        this.r=(r/100)/n;
        this.PMT=PMT;

this.t=(Math.log(((this.A*this.r+this.PMT)/(this.P*this.r+this.PMT))))/Math.l
og(1+this.r);

        return this.t/n;
    }
```

```
//-------------------------------------------------------------------------//
//-----------------------Mortgage and loan---------------------------------//

    public double PMT(double P, double r, double t){
        this.P=P;
        this.r=(r/100)/n;
        this.t=t*n;
        double ans=(this.P* this.r *Math.pow(1+ this.r, this.t))/(Math.pow(1+
this.r, this.t)-1);

        return ans ;
    }

    public double LoanAmount(double PMT,double t, double r){
        this.PMT=PMT;
        this.r=(r/100)/n;
        this.t=t*n;
        double ans=(this.PMT*(Math.pow(1+ this.r, this.t)-1)/( this.r
*Math.pow(1+ this.r, this.t)));

        return ans ;
    }

    public double LoanTerm(double P,double PMT, double r){
        this.PMT=PMT;
        this.r=(r/100)/n;
        this.P=P;
        this.t=Math.log(this.PMT/(this.PMT-
this.r*this.P))/Math.log(1+this.r);

        return this.t/n;
    }

    }
//*************************************************************************
*************************************//
```

Validations Class

Validations class contains almost all the validations I've done in this course work. All the alert messages are included in this class.

```
package com.hirunz2000;

import javafx.scene.control.Alert;

public class Validations {

// This method checks whether the given String array can be converted into
Double.
// If possible, it will return true and otherwise it will put an error msg.
// This method also ignores the empty strings.

    public static boolean doubleInputCheck(String[] arr){
        try{
            for(int i=0;i<arr.length;i++) {
                if (arr[i].isEmpty()) {
                    continue;
                }
                double double_input = Double.parseDouble(arr[i]);
            }
                return true;

        }catch (Exception e){
            Alert alert = new Alert(Alert.AlertType.INFORMATION);
            alert.setTitle("Invalid Input");
            alert.setHeaderText(null);
            alert.setContentText("Please input only integer or double
values!");
            alert.showAndWait();
        }
        return false;
    }

    // this is an error message to inform the user to input only one field
empty.
    //for mortgage calculation, there can be two empty fields but one of them
must be down payment.

    public static void InputErrorAlert(String str){
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Invalid Input");
        alert.setHeaderText(null);
        alert.setContentText("Please leave only one field empty "+str+"to
calculate.\nRefer HELP for more details.");
        alert.showAndWait();
    }
```

```
    // this method will accept a String array and returns the number of empty
strings.
    // this method is used to prevent the user from entering all or leaving
more than 2 text fields.

    public static int EmptyStringCount(String[] arr,int stop){
        int EmptyStringCount =0;
        for(int i=0;i<stop;i++){

            if(arr[i].isEmpty()){
                EmptyStringCount++;
            }
        }
        return EmptyStringCount;
    }

    // this method will return the empty string index of an array.
    // string with the index exception will be excluded.
    // this method used after the previous method.
    // this method determines what to calculate.
    public static int EmptyStringCount(String[] arr,int stop,int exception){
        int EmptyStringCount =0;
        for(int i=0;i<stop;i++){
            if(i==exception){
                continue;
            }
            if(arr[i].isEmpty()){
                EmptyStringCount++;
            }
        }
        return EmptyStringCount;
    }

    // this is an overloaded method of the EmptyStringCount method.
    // this is used when there is no exception.
    // usage is same as EmptyStringCount method
    public static int EmptyStringIndex(String[] arr,int stop){
        int EmptyStringIndex = 0;
        for(int i=0;i<stop;i++){
            if(arr[i].isEmpty()){
                EmptyStringIndex = i;
                break;
            }
        }
       return EmptyStringIndex;
    }

    // alert message to alert user that the interest cannot be calculated in
relevant calculators.
    public static void EmptyInterestAlert(){
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Invalid Input");
        alert.setHeaderText(null);
        alert.setContentText("Sorry!. Cannot calculate interest in this
calculator.");
        alert.showAndWait();
    }
```

```
    // this method is called only in keyboard class
    // this method is used to alert the user when they try to input more than
one periods per text field.'
    public static void periodErrorAlert(){
        Alert alert = new Alert(Alert.AlertType.INFORMATION);
        alert.setTitle("Invalid Input");
        alert.setHeaderText(null);
        alert.setContentText("Cannot enter period twice for a number!");
        alert.showAndWait();
    }



}
//*************************************************************************
************************************************************************//
```

Calculator Designing in Brief

For designing purposes, I used Grid Panes inside Anchor Panes because Grid Panes don't require LayoutX and LayoutY. Therefore, I could easily position the labels and text fields.

Then I added the Grid Panes to an Anchor Pane named root which is then added to the scene.

I used methods with return type as Scene and I returned the previous scene.

Then in Main Menu, under the relevant button on click action, under the event handler I called the above method and assigned the returned Scene to a stage.

Each calculator is written in the following order.

1. Input Labels
2. Input Text Fields
3. Input Type labels (e.x: $)
4. Scan for last entry and display
5. Calculate Button and calculation part (with file handling)
6. Clear button
7. Back button
8. Help Button
9. History Button

All the styling part is done with an external stylesheet which is linked to the calculator.

When a textfield is clicked, I call the keyboard class and give a gridpane and the selected textfield as arguments. When a gridpane is received to keyboard class, it will be visible and it will do the necessary function (add or remove) from the received textfield.

In the calculate button, I first validate the inputs which user has inputted in the text fields and show an alert when necessary. Then I find out which field is empty and then using if-else conditions, I called the relevant method from Formulae class and set the output.

Once the output is set, app will store the data in a txt file with appending to the existing file (by using auto flush). This saves the whole history of the application with the time.

Then the file will again write to another txt file and overwrite it each time. That saves the last entry.

This process is same for all four calculators.

The button positions, colours are the same for all four calculators to maintain the colour consistency throughout the app.

Fixed Deposit Calculator (No Monthly Contributions)

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.InputMismatchException;
import java.util.Scanner;

public class FixedDeposit {
    public static Scene FD(){

     //display keyboard before clicking a textfield and  not allows to enter.
        GridPane keyboardGrid=new GridPane();
        Keyboard.keyboardDisplay(keyboardGrid,null);

//------------------------------Input Labels--------------------------------//

        Label lblHeading =new Label("Fixed Deposit Calculator");
        lblHeading.setPrefSize(498,55);
        lblHeading.setLayoutX(3);
        lblHeading.setLayoutY(3);
        lblHeading.setId("label-heading");

        Label lblPrincipleAmount =new Label("Principle Amount");
        lblPrincipleAmount.setPrefSize(125,21);
        lblPrincipleAmount.setId("label1");

        Label lblFutureValue =new Label("Future Value");
        lblFutureValue.setPrefSize(125,21);
        lblFutureValue.setId("label1");

        Label lblInterestRate =new Label("Interest Rate");
        lblInterestRate.setPrefSize(125,21);
        lblInterestRate.setId("label1");

        Label lblTimePeriod =new Label("Time Period");
        lblTimePeriod.setPrefSize(125,21);
        lblTimePeriod.setId("label1");

//--------------------------------------------------------------------------//
//-------------------------Input TextFields---------------------------------//

        TextField txtPrincipleAmount = new TextField();
        txtPrincipleAmount.setPrefSize(170,25);
        txtPrincipleAmount.setId("txt");
```

```
        txtPrincipleAmount.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtPrincipleAmount);
            }
        });

        TextField txtFutureValue =new TextField();
        txtFutureValue.setPrefSize(170,25);
        txtFutureValue.setId("txt");
        txtFutureValue.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtFutureValue);
            }
        });

        TextField txtInterestRate =new TextField();
        txtInterestRate.setPrefSize(170,25);
        txtInterestRate.setId("txt");
        txtInterestRate.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtInterestRate);
            }
        });

        TextField txtTimePeriod = new TextField();
        txtTimePeriod.setPrefSize(170,25);
        txtTimePeriod.setId("txt");
        txtTimePeriod.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtTimePeriod);
            }
        });


//-------------------------------------------------------------------------//
//-----------------------Add Inputs to a gridpane--------------------------//

        GridPane inputs=new GridPane();
        inputs.setPrefSize(288,226);
        inputs.setVgap(5);
        inputs.setLayoutX(35);
        inputs.setLayoutY(100);

        inputs.add(lblPrincipleAmount,0,0);
        inputs.add(lblFutureValue,0,1);
        inputs.add(lblInterestRate,0,2);
        inputs.add(lblTimePeriod,0,3);

        inputs.add(txtPrincipleAmount,1,0);
        inputs.add(txtFutureValue,1,1);
        inputs.add(txtInterestRate,1,2);
        inputs.add(txtTimePeriod,1,3);
```

```
//-------------------------------------------------------------------------//
//--------------------Retrieving Last Entry--------------------------------//

        Double[] lastEntries = new Double[4];
        try {
            try {

                File file = new File("src/history/lastEntry/last-entry-
fd.txt");

                Scanner input = new Scanner(file);
                for (int i = 0; i < lastEntries.length; i++) {
                    lastEntries[i]=input.nextDouble();
                }
                input.close();
            } catch (FileNotFoundException e) {
                System.out.println("File not found");
            }
        }catch (InputMismatchException e){
        // to handle the error when the file is empty
            System.out.println("File is empty.");;
        }
        txtPrincipleAmount.setText(String.valueOf(lastEntries[0]));
        txtFutureValue.setText(String.valueOf(lastEntries[1]));
        txtInterestRate.setText(String.valueOf(lastEntries[2]));
        txtTimePeriod.setText(String.valueOf(lastEntries[3]));


//-------------------------------------------------------------------------//
//----------------------Input Type Labels----------------------------------//

        Label dollar1=new Label(" $");
        dollar1.setPrefSize(15,25);
        dollar1.setId("label1");

        Label dollar2=new Label(" $");
        dollar2.setPrefSize(15,25);
        dollar2.setId("label1");

        Label rate=new Label("%");
        rate.setPrefSize(15,25);
        rate.setId("label1");

        Label period=new Label("yrs");
        period.setPrefSize(20,25);
        period.setId("label1");

//-------------------------------------------------------------------------//
//---------------------------Input Type Grid-------------------------------//

        GridPane inputType=new GridPane();
        inputType.setPrefSize(256,226);
        inputType.setVgap(5);
        inputType.setLayoutX(295);
        inputType.setLayoutY(100);
```

```
        inputType.add(dollar1,0,0);
        inputType.add(dollar2,0,1);
        inputType.add(rate,0,2);
        inputType.add(period,0,3);

//-------------------------------------------------------------------------//
//-------------------------Calculate button--------------------------------//

        Button btnCalculate = new Button("Calculate");
        btnCalculate.setId("btnCalculate");
        btnCalculate.setPrefSize(84,35);
        btnCalculate.setLayoutX(210);
        btnCalculate.setLayoutY(260);
        btnCalculate.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                String str = "%.2f";
                int arrSize=4;
                // assigning textfield text to  array to validate
                String[] arr=new String[arrSize];
                arr[0]= txtPrincipleAmount.getText();
                arr[1]=txtFutureValue.getText();
                arr[2]=txtInterestRate.getText();
                arr[3]=txtTimePeriod.getText();

                //finds out how many empty strings are there in the array
                int EmptyStringCount
=Validations.EmptyStringCount(arr,arrSize);
                if(EmptyStringCount !=1){
    //if empty number count is greater than 1 or less than 1 cannot calculate
                    Validations.InputErrorAlert("");
                }
                else {
            //if there is only one empty index, this block will execute.
                    //then takes finds out the empty string index.
                    int EmptyStringIndex = Validations.EmptyStringIndex(arr,
arrSize);
// according to the empty string index, the relevant code block will execute.
                    if (Validations.doubleInputCheck(arr)//creating formulae
object andthen pass the relevant arguments to the method and get the result.
                        // finally sets the value to the empty textfield.
                        if (EmptyStringIndex == 0) {
                            Formulae P=new Formulae();
                            double
result=P.presentValue(Double.valueOf(arr[3]),Double.valueOf(arr[1]),Double.va
lueOf(arr[2]));

txtPrincipleAmount.setText(String.format(str,result));
                        }
                        else if (EmptyStringIndex ==1) {
                            Formulae A = new Formulae();
                            double result =
A.futureValue(Double.valueOf(arr[3]), Double.valueOf(arr[0]),
Double.valueOf(arr[2]));
                            txtFutureValue.setText(String.format(str,
result));
                        }
                        else if (EmptyStringIndex ==2){
```

```
                            Formulae r=new Formulae();
                            double
result=r.interestRate(Double.valueOf(arr[3]),Double.valueOf(arr[0]),Double.va
lueOf(arr[1]));

txtInterestRate.setText(String.format(str,result));
                        }
                        else {
                            Formulae t=new Formulae();
                            double
result=t.timePeriod(Double.valueOf(arr[1]),Double.valueOf(arr[0]),Double.valu
eOf(arr[2]));
                            txtTimePeriod.setText(String.format(str,result));
                        }

//-----------------------------file handling-------------------------------//
        //once the above code block is executed, writing to the file.
        //to make the history more sense, i added date and time to the entry.

                        //date and time
                        DateTimeFormatter dateTimeFormatter =
DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
                        LocalDateTime currentTime = LocalDateTime.now();

                        //write to file begins.
                        FileWriter fileWriter =null;
                        PrintWriter output =null;
                        File file=new File("src/history/history-fd.txt");

                        try{

                            fileWriter =new FileWriter(file,true);
                            output =new PrintWriter(fileWriter,true);


output.write("*************************************************\n");
                            output.write("***********
Entry:"+dateTimeFormatter.format(currentTime)+"  ***********\n");

output.write("*************************************************\n");

                            output.write(String.format("Monthly Pay  :
$%12s\n",Double.valueOf(txtPrincipleAmount.getText())));
                            output.write(String.format("Home Price   :
$%12s\n",Double.valueOf(txtFutureValue.getText())));
                            output.write(String.format("Interest Rate:   %10s
%%\n",Double.valueOf(txtInterestRate.getText())));
                            output.write(String.format("Loan Term    :   %8s
yrs\n",Double.valueOf(txtTimePeriod.getText())));


output.write("*************************************************\n\n\n");

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                        finally {
                            try {
                                fileWriter.close();// closing the filewriter.
```

```
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                            output.close();// closing the output.
                        }

                        // again I write write to a file. This time i don't
use auto flush because  i want to overwrite.
                        // by overwriting, i save the last entry.

                        try {
                            File file1=new File("src/history/lastEntry/last-
entry-fd.txt");
                            FileWriter fileWriter1=new FileWriter(file1);


fileWriter1.write(Double.valueOf(txtPrincipleAmount.getText()) + " ");

fileWriter1.write(Double.valueOf(txtFutureValue.getText()) + " ");

fileWriter1.write(Double.valueOf(txtInterestRate.getText()) + " ");

fileWriter1.write(Double.valueOf(txtTimePeriod.getText()) + " ");
                            fileWriter1.close();

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }

                }

            }
        });

//-----------------------------------------------------------------------------//
//----------------------------Clear button-------------------------------------//

        Button btnClear = new Button("Clear");
        btnClear.setId("btnClear");
        btnClear.setPrefSize(70,35);
        btnClear.setLayoutX(40);
        btnClear.setLayoutY(260);
        btnClear.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                txtPrincipleAmount.setText("");
                txtFutureValue.setText("");
                txtInterestRate.setText("");
                txtTimePeriod.setText("");
            }
        });

//-----------------------------------------------------------------------------//
//-----------------------Back button-------------------------------------------//

        Button btnBack = new Button("Go Back");
        btnBack.setId("btn2");
        btnBack.setPrefSize(60,20);
```

```
        btnBack.setLayoutX(10);
        btnBack.setLayoutY(15);
        btnBack.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //closing the current stage
                Stage stage = (Stage) btnBack.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(440);
                primaryStage.setMaxWidth(420);
                primaryStage.setMinHeight(440);
                primaryStage.setMinWidth(420);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(MainMenu.Mainmenu());
                primaryStage.show();

            }
        });

//-------------------------------------------------------------------------//
//-------------------------Help button-------------------------------------//

        Button btnHelp = new Button("Help");
        btnHelp.setId("btn2");
        btnHelp.setPrefSize(60,20);
        btnHelp.setLayoutX(360);
        btnHelp.setLayoutY(15);
        btnHelp.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(550);
                primaryStage.setMaxWidth(500);
                primaryStage.setMinHeight(550);
                primaryStage.setMinWidth(500);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(HelpView.FD());
                primaryStage.show();

            }
        });

//-------------------------------------------------------------------------//
//-------------------------History button-----------------------------------//

        Button btnHistory = new Button("History");
        btnHistory.setId("btn2");
        btnHistory.setPrefSize(60,20);
        btnHistory.setLayoutX(430);
        btnHistory.setLayoutY(15);
```

```
        btnHistory.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                HistoryDisplay.history("src/history/history-fd.txt");

            }
        });

//-------------------------------------------------------------------------//
// -----------------Adding to anchorPane and stage display-----------------//

        AnchorPane root =new AnchorPane();
        root.getChildren().addAll(lblHeading,inputs,
inputType,keyboardGrid,btnCalculate,btnClear,btnBack, btnHelp,btnHistory);
        root.setId("root");

        Scene scene=new Scene(root,500,400);
        scene.getStylesheets().add("css/layout.css");// link to external css
        return scene;
    }
}
```

Savings Calculator (With Monthly Contribution)

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.InputMismatchException;
import java.util.Scanner;

public class SavingsCalculator {
    public static Scene savings(){
        //display keyboard before clicking a textfield and  not allows to
enter.
        GridPane keyboardGrid=new GridPane();
        Keyboard.keyboardDisplay(keyboardGrid,null);

//----------------------------Input Labels---------------------------------//

        Label lblHeading =new Label("Savings Calculator");
        lblHeading.setPrefSize(498,55);
        lblHeading.setLayoutX(3);
        lblHeading.setLayoutY(3);
        lblHeading.setId("label-heading");

        Label lblPrincipleAmount =new Label("Principle Amount");
        lblPrincipleAmount.setPrefSize(125,21);
        lblPrincipleAmount.setId("label1");

        Label lblFutureValue =new Label("Future Value");
        lblFutureValue.setPrefSize(125,21);
        lblFutureValue.setId("label1");

        Label lblTimePeriod =new Label("Time Period");
        lblTimePeriod.setPrefSize(125,21);
        lblTimePeriod.setId("label1");

        Label lblPMT =new Label("Monthly Payment");
        lblPMT.setPrefSize(125,21);
        lblPMT.setId("label1");


        Label lblInterestRate =new Label("Interest Rate");
        lblInterestRate.setPrefSize(125,21);
        lblInterestRate.setId("label1");
```

```
//-----------------------------------------------------------------------------//
//----------------------------Input TextFields---------------------------------//

        TextField txtPrincipleAmount = new TextField();
        txtPrincipleAmount.setPrefSize(176,25);
        txtPrincipleAmount.setId("txt");
        txtPrincipleAmount.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtPrincipleAmount);
            }
        });

        TextField txtFutureValue =new TextField();
        txtFutureValue.setPrefSize(176,25);
        txtFutureValue.setId("txt");
        txtFutureValue.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtFutureValue);
            }
        });

        TextField txtPMT =new TextField();
        txtPMT.setPrefSize(176,25);
        txtPMT.setId("txt");
        txtPMT.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtPMT);
            }
        });

        TextField txtTimePeriod = new TextField();
        txtTimePeriod.setPrefSize(176,25);
        txtTimePeriod.setId("txt");
        txtTimePeriod.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtTimePeriod);
            }
        });

        TextField txtInterestRate =new TextField();
        txtInterestRate.setPrefSize(176,25);
        txtInterestRate.setId("txt");
        txtInterestRate.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid, txtInterestRate);
            }
        });

//-----------------------------------------------------------------------------//
//---------------------Add Inputs to a gridpane--------------------------------//

        GridPane inputs=new GridPane();
        inputs.setPrefSize(288,226);
        inputs.setVgap(5);
```

```
        inputs.setLayoutX(35);
        inputs.setLayoutY(100);

        inputs.add(lblPrincipleAmount,0,0);
        inputs.add(lblFutureValue,0,1);
        inputs.add(lblPMT,0,2);
        inputs.add(lblTimePeriod,0,3);
        inputs.add(lblInterestRate,0,4);


        inputs.add(txtPrincipleAmount,1,0);
        inputs.add(txtFutureValue,1,1);
        inputs.add(txtPMT,1,2);
        inputs.add(txtTimePeriod,1,3);
        inputs.add(txtInterestRate,1,4);

//-----------------------------------------------------------------------------//
//--------------------------Retrieving Last Entry--------------------------//

        Double[] lastEntries = new Double[5];
        try {
            try {

                File file = new File("src/history/lastEntry/last-entry-
savings.txt");

                Scanner input = new Scanner(file);
                for (int i = 0; i < lastEntries.length; i++) {
                    lastEntries[i]=input.nextDouble();
                }
                input.close();
            } catch (FileNotFoundException e) {
                System.out.println("File not found");
            }
        }catch (InputMismatchException e){
// to handle the error when the file is empty
            System.out.println("File is empty.");;
        }
        txtPrincipleAmount.setText(String.valueOf(lastEntries[0]));
        txtFutureValue.setText(String.valueOf(lastEntries[1]));
        txtPMT.setText(String.valueOf(lastEntries[2]));
        txtInterestRate.setText(String.valueOf(lastEntries[3]));
        txtTimePeriod.setText(String.valueOf(lastEntries[4]));

//-----------------------------------------------------------------------------//
//-------------------------Input Type Labels-------------------------------//

       Label dollar1=new Label(" $");
        dollar1.setPrefSize(15,25);
        dollar1.setId("label1");

        Label dollar2=new Label(" $");
        dollar2.setPrefSize(15,25);
        dollar2.setId("label1");

        Label dollar3=new Label(" $");
        dollar3.setPrefSize(15,25);
        dollar3.setId("label1");
```

```
        Label period=new Label("yrs");
        period.setPrefSize(20,25);
        period.setId("label1");

        Label rate=new Label("%");
        rate.setPrefSize(15,25);
        rate.setId("label1");

// ------------------------------------------------------------------------//
//----------------------Input Type Grid-----------------------------------//

        GridPane inputType=new GridPane();
        inputType.setPrefSize(256,226);
        inputType.setVgap(5);
        inputType.setLayoutX(295);
        inputType.setLayoutY(100);

        inputType.add(dollar1,0,0);
        inputType.add(dollar2,0,1);
        inputType.add(dollar3,0,2);
        inputType.add(period,0,3);
        inputType.add(rate,0,4);

//------------------------------------------------------------------------//
//----------------------Calculate button----------------------------------//

        Button btnCalculate = new Button("Calculate");
        btnCalculate.setId("btnCalculate");
        btnCalculate.setPrefSize(84,35);
        btnCalculate.setLayoutX(210);
        btnCalculate.setLayoutY(260);
        btnCalculate.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                String str = "%.2f";
                int arrSize=5;
                // assigning textfield text to  array to validate
                String[] arr=new String[arrSize];
                arr[0]=txtPrincipleAmount.getText();
                arr[1]=txtFutureValue.getText();
                arr[2]=txtPMT.getText();
                arr[3]=txtTimePeriod.getText();
                arr[4]=txtInterestRate.getText();

                //finds out how many empty strings are there in the array
                int EmptyStringCount
=Validations.EmptyStringCount(arr,arrSize);
                if(EmptyStringCount !=1 ){
                    Validations.InputErrorAlert("");
                }
                else if (arr[4].isEmpty()){
                    //if empty number count is greater than 1 or less than 1
cannot calculate
                    Validations.EmptyInterestAlert();
                }
```

```
                else {
                 //if there is only one empty index, this block will execute.
                    //then takes finds out the empty string index.
                    int EmptyStringIndex = Validations.EmptyStringIndex(arr,
arrSize);
                    if (Validations.doubleInputCheck(arr)) {
                        //creating formulae object andthen pass the relevant
arguments to the method and get the result.
                        // finally sets the value to the empty textfield.

                        if (EmptyStringIndex == 0) {
                            Formulae P=new Formulae();
                            double result=P.P(Double.valueOf(arr[2]),
Double.valueOf(arr[1]), Double.valueOf(arr[4]), Double.valueOf(arr[3]));

txtPrincipleAmount.setText(String.format(str,result));
                        }
                        else if (EmptyStringIndex ==1){
                            Formulae FV=new Formulae();
                            double result=FV.FV(Double.valueOf(arr[2]),
Double.valueOf(arr[0]), Double.valueOf(arr[4]), Double.valueOf(arr[3]));

txtFutureValue.setText(String.format(str,result));
                        }
                        else if (EmptyStringIndex ==2){
                            Formulae pmt=new Formulae();
                            double result=pmt.PMT(Double.valueOf(arr[0]),
Double.valueOf(arr[1]), Double.valueOf(arr[4]),Double.valueOf(arr[3]));
                            txtPMT.setText(String.format(str,result));
                        }
                        else {
                            Formulae t=new Formulae();
                            double result=t.time(Double.valueOf(arr[0]),
Double.valueOf(arr[1]), Double.valueOf(arr[4]), Double.valueOf(arr[2]));
                            txtTimePeriod.setText(String.format(str,result));
                        }

//------------------------------file handling--------------------------------//
               //once the above code block is executed, writing to the file.
       //to make the history more sense, i added date and time to the entry.

                        //date and time
                        DateTimeFormatter dateTimeFormatter =
DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
                        LocalDateTime currentTime = LocalDateTime.now();

                        //write to file begins.
                        FileWriter fileWriter =null;
                        PrintWriter output =null;
                        File file=new File("src/history/history-
savings.txt");
                        try{

                            fileWriter =new FileWriter(file,true);
                            output =new PrintWriter(fileWriter,true);


output.write("************************************************\n");
```

```
                            output.write("***********
Entry:"+dateTimeFormatter.format(currentTime)+"  ***********\n");

output.write("************************************************\n");

                            output.write(String.format("Monthly Pay  :
$%12s\n",Double.valueOf(txtPrincipleAmount.getText())));
                            output.write(String.format("Home Price   :
$%12s\n",Double.valueOf(txtFutureValue.getText())));
                            output.write(String.format("Home Price   :
$%12s\n",Double.valueOf(txtPMT.getText())));
                            output.write(String.format("Interest Rate:   %10s
%%\n",Double.valueOf(txtInterestRate.getText())));
                            output.write(String.format("Loan Term    :   %8s
yrs\n",Double.valueOf(txtTimePeriod.getText())));


output.write("************************************************\n\n\n");
                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                        finally {
                            try {
                                fileWriter.close();
                               // closing the file writer.
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                            output.close();// closing the output.
                        }
                        // again I write write to a file. This time i don't
use auto flush because  i want to overwrite.
                        // by overwriting, i save the last entry.
                        try {
                            File file1=new File("src/history/lastEntry/last-
entry-savings.txt");
                            FileWriter fileWriter1=new FileWriter(file1);


fileWriter1.write(Double.valueOf(txtPrincipleAmount.getText()) + " ");

fileWriter1.write(Double.valueOf(txtFutureValue.getText()) + " ");

fileWriter1.write(Double.valueOf(txtPMT.getText()) + " ");

fileWriter1.write(Double.valueOf(txtInterestRate.getText()) + " ");

fileWriter1.write(Double.valueOf(txtTimePeriod.getText()) + " ");
                            fileWriter1.close();

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }

                }

            }
        });
```

```
//-------------------------------------------------------------------------//
//----------------------Clear button---------------------------------------//

        Button btnClear = new Button("Clear");
        btnClear.setId("btnClear");
        btnClear.setPrefSize(70,35);
        btnClear.setLayoutX(40);
        btnClear.setLayoutY(260);
        btnClear.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                txtPMT.setText("");
                txtPrincipleAmount.setText("");
                txtFutureValue.setText("");
                txtInterestRate.setText("");
                txtTimePeriod.setText("");

            }
        });

//-------------------------------------------------------------------------//
//-------------------------Back button-------------------------------------//

        Button btnBack = new Button("Go Back");
        btnBack.setId("btn2");
        btnBack.setPrefSize(60,20);
        btnBack.setLayoutX(10);
        btnBack.setLayoutY(15);
        btnBack.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //closing the current stage
                Stage stage = (Stage) btnBack.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                primaryStage.setMaxHeight(440);
                primaryStage.setMaxWidth(420);
                primaryStage.setMinHeight(440);
                primaryStage.setMinWidth(420);
                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(MainMenu.Mainmenu());
                primaryStage.show();

            }
        });

//-------------------------------------------------------------------------//
//-------------------------Help button-------------------------------------//

        Button btnHelp = new Button("Help");
        btnHelp.setId("btn2");
        btnHelp.setPrefSize(60,20);
        btnHelp.setLayoutX(360);
        btnHelp.setLayoutY(15);
```

```
        btnHelp.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(550);
                primaryStage.setMaxWidth(500);
                primaryStage.setMinHeight(550);
                primaryStage.setMinWidth(500);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(HelpView.Savings());
                primaryStage.show();

            }
        });

//-------------------------------------------------------------------------//
//----------------------History button-----------------------------------//

        Button btnHistory = new Button("History");
        btnHistory.setId("btn2");
        btnHistory.setPrefSize(60,20);
        btnHistory.setLayoutX(430);
        btnHistory.setLayoutY(15);
        btnHistory.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                HistoryDisplay.history("src/history/history-savings.txt");

            }
        });

//-------------------------------------------------------------------------//
// ---------------Adding to anchorPane and stage display-------------------//

        AnchorPane root =new AnchorPane();
        root.getChildren().addAll(lblHeading,inputs,
inputType,keyboardGrid,btnCalculate,btnClear,btnBack,btnHelp,btnHistory);
        root.setId("root");

        Scene scene=new Scene(root,500,400);
        scene.getStylesheets().add("css/layout.css");// link to external css
        return scene;
    }
}
//*************************************************************************
*************************************************************************//
```

Mortgage Calculator

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.TextField;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;
import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.InputMismatchException;
import java.util.Scanner;


public class MortgageCalculatorWindow  {

    public static Scene MortgageCalculator() {
    //display keyboard before clicking a textfield and  not allows to eenter.
        GridPane keyboardGrid=new GridPane();
        Keyboard.keyboardDisplay(keyboardGrid,null);

//-----------------------Input Labels---------------------------------------//


        Label lblHeading =new Label("Mortgage Calculator");
        lblHeading.setPrefSize(498,55);
        lblHeading.setLayoutX(3);
        lblHeading.setLayoutY(3);
        lblHeading.setId("label-heading");



        Label lblMonthlyPay =new Label("Monthly Pay");
        lblMonthlyPay.setPrefSize(105,21);
        lblMonthlyPay.setId("label1");

        Label lblHousePrice =new Label("Home Price");
        lblHousePrice.setPrefSize(105,21);
        lblHousePrice.setId("label1");

        Label lblDownPayment =new Label("Down Payment");
        lblDownPayment.setPrefSize(105,21);
        lblDownPayment.setId("label1");

        Label lblLoanTerm =new Label("Loan Term");
        lblLoanTerm.setPrefSize(105,21);
        lblLoanTerm.setId("label1");

        Label lblInterestRate =new Label("Interest Rate");
        lblInterestRate.setPrefSize(105,21);
```

```
        lblInterestRate.setId("label1");


//-----------------------------------------------------------------------------//
//-----------------------------Input TextFields--------------------------------//

        TextField txtMonthlyPay = new TextField();
        txtMonthlyPay.setPrefSize(176,25);
        txtMonthlyPay.setId("txt");
        txtMonthlyPay.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtMonthlyPay);
            }
        });

        TextField txtHousePrice =new TextField();
        txtHousePrice.setPrefSize(176,25);
        txtHousePrice.setId("txt");
        txtHousePrice.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtHousePrice);
            }
        });

        TextField txtDownPayment =new TextField();
        txtDownPayment.setPrefSize(176,25);
        txtDownPayment.setId("txt");
        txtDownPayment.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtDownPayment);
            }
        });

        TextField txtLoanTerm = new TextField();
        txtLoanTerm.setPrefSize(176,25);
        txtLoanTerm.setId("txt");
        txtLoanTerm.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtLoanTerm);
            }
        });

        TextField txtInterestRate = new TextField();
        txtInterestRate.setPrefSize(176,25);
        txtInterestRate.setId("txt");
        txtInterestRate.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtInterestRate);
            }
        });
```

```
//-------------------------------------------------------------------------//
//-------------------------Add Inputs to a gridpane-----------------------//

        GridPane inputs=new GridPane();
        inputs.setPrefSize(288,226);
        inputs.setVgap(5);
        inputs.setLayoutX(35);
        inputs.setLayoutY(90);

        inputs.add(lblMonthlyPay,0,0);
        inputs.add(lblHousePrice,0,1);
        inputs.add(lblDownPayment,0,2);
        inputs.add(lblLoanTerm,0,3);
        inputs.add(lblInterestRate,0,4);

        inputs.add(txtMonthlyPay,1,0);
        inputs.add(txtHousePrice,1,1);
        inputs.add(txtDownPayment,1,2);
        inputs.add(txtLoanTerm,1,3);
        inputs.add(txtInterestRate,1,4);

//-------------------------------------------------------------------------//
//---------------------Retrieving Last Entry-------------------------------//

        Double[] lastEntries = new Double[5];
        try {
            try {

                File file = new File("src/history/lastEntry/last-entry-
mortgage.txt");

                Scanner input = new Scanner(file);
                for (int i = 0; i < lastEntries.length; i++) {
                    lastEntries[i]=input.nextDouble();
                }
                input.close();
            } catch (FileNotFoundException e) {
                System.out.println("File not found");
            }
        }catch (InputMismatchException e){
// to handle the error when the file is empty
            System.out.println("File is empty.");;
        }
        txtMonthlyPay.setText(String.valueOf(lastEntries[0]));
        txtHousePrice.setText(String.valueOf(lastEntries[1]));
        txtInterestRate.setText(String.valueOf(lastEntries[2]));
        txtLoanTerm.setText(String.valueOf(lastEntries[3]));
        txtDownPayment.setText(String.valueOf(lastEntries[4]));

//-------------------------------------------------------------------------//
//-------------------------Input Type Labels-----------------------------//


        Label dollar1=new Label("$");
        dollar1.setPrefSize(15,25);
        dollar1.setId("label1");
```

```
        Label dollar2=new Label("$");
        dollar2.setPrefSize(15,25);
        dollar2.setId("label1");

        Label dollar3=new Label("$");
        dollar3.setPrefSize(15,25);
        dollar3.setId("label1");

        Label period=new Label("yrs");
        period.setPrefSize(20,25);
        period.setId("label1");

        Label rate=new Label("%");
        rate.setPrefSize(15,25);
        rate.setId("label1");


//-------------------------------------------------------------------------//
//-----------------------Input Type Grid-----------------------------------//

        GridPane inputType=new GridPane();
        inputType.setPrefSize(256,226);
        inputType.setVgap(5);
        inputType.setLayoutX(295);
        inputType.setLayoutY(90);

        inputType.add(dollar1,0,0);
        inputType.add(dollar2,0,1);
        inputType.add(dollar3,0,2);
        inputType.add(period,0,3);
        inputType.add(rate,0,4);


//-------------------------------------------------------------------------//
//-----------------------Calculate button----------------------------------//

        Button btnCalculate = new Button("Calculate");
        btnCalculate.setId("btnCalculate");
        btnCalculate.setPrefSize(84,35);
        btnCalculate.setLayoutX(210);
        btnCalculate.setLayoutY(260);
        btnCalculate.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                String str = "%.2f";
                int arrSize=5;
                // assigning textfield text to  array to validate
                String[] arr=new String[arrSize];
                arr[0]=txtMonthlyPay.getText();
                arr[1]=txtHousePrice.getText();
                arr[2]=txtInterestRate.getText();
                arr[3]=txtLoanTerm.getText();
                arr[4]=txtDownPayment.getText();

                //finds out how many empty strings are there in the array
                int EmptyStringCount
=Validations.EmptyStringCount(arr,arrSize,4);
```

```
                if(EmptyStringCount >1 | EmptyStringCount ==0 ){
                    Validations.InputErrorAlert("(except down payment) ");
                }
                else if(arr[2].isEmpty()){
    //if empty number count is greater than 1 or less than 1 cannot calculate
                    Validations.EmptyInterestAlert();
                }
                else {
            //if there is only one empty index, this block will execute.
                    //then takes finds out the empty string index.
                    int EmptyStringIndex = Validations.EmptyStringIndex(arr,
arrSize);
// according to the empty string index, the relevant code block will execute.
                    if (Validations.doubleInputCheck(arr)) {
                        if (arr[4].isEmpty()){
                    //assigned value 0 to down payment if  there is no value.
                            arr[4]="0";
                            txtDownPayment.setText("0.00");
                        }
                        if (EmptyStringIndex == 0) {
                            double P = Double.valueOf(arr[1] )-
Double.valueOf(arr[4]);
                            double r = Double.valueOf(arr[2]);
                            double t = Double.valueOf(arr[3]);
                            //creating an object of formulae class
                            Formulae Loan = new Formulae();
        // then pass the relevant arguments to the method and get the result.
                            double result = Loan.PMT(P, r, t);
                            // finally sets the value to the empty textfield.
                            txtMonthlyPay.setText(String.format(str,
result));
                        }
                        else if (EmptyStringIndex ==1){
                            double PMT = Double.valueOf(arr[0]);
                            double r = Double.valueOf(arr[2]);
                            double t = Double.valueOf(arr[3]);
                            //creating an object of formulae class
                            Formulae Loan = new Formulae();
      // then pass the relevant arguments to the method and get the result
                            double result = Loan.LoanAmount(PMT, t, r);
                            // finally sets the value to the empty textfield.
                            txtHousePrice.setText(String.format(str, result
+Double.valueOf(arr[4])));
                        }
                        else {
                            double P = Double.valueOf(arr[1] )-
Double.valueOf(arr[4]);
                            double r = Double.valueOf(arr[2]);
                            double PMT = Double.valueOf(arr[0]);
                            //creating an object of formulae class
                            Formulae Loan = new Formulae();
        // then pass the relevant arguments to the method and get the result
                            double result = Loan.LoanTerm(P, PMT, r);
                            // finally sets the value to the empty textfield.
                            txtLoanTerm.setText(String.format(str,result));

                        }
```

```
//-----------------------------file handling--------------------------------//
             //once the above code block is executed, writing to the file.
        //to make the history more sense, i added date and time to the entry.

                        //date and time
                        DateTimeFormatter dateTimeFormatter =
DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
                        LocalDateTime currentTime = LocalDateTime.now();

                        //write to file begins.
                        FileWriter fileWriter =null;
                        PrintWriter output =null;
                        File file=new File("src/history/history-
mortgage.txt");

                        try{

                            fileWriter =new FileWriter(file,true);
                            output =new PrintWriter(fileWriter,true);
                        //auto flush.


output.write("**************************************************\n");
                            output.write("***********
Entry:"+dateTimeFormatter.format(currentTime)+"  ***********\n");

output.write("**************************************************\n");

                            output.write(String.format("Monthly Pay  :
$%12s\n",Double.valueOf(txtMonthlyPay.getText())));
                            output.write(String.format("Home Price   :
$%12s\n",Double.valueOf(txtHousePrice.getText())));
                            output.write(String.format("Interest Rate:   %10s
%%\n",Double.valueOf(txtInterestRate.getText())));
                            output.write(String.format("Loan Term    :   %8s
yrs\n",Double.valueOf(txtLoanTerm.getText())));
                            output.write(String.format("Down Payment :
$%12s\n",Double.valueOf(txtDownPayment.getText())));


output.write("**************************************************\n\n\n");

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                        finally {
                            try {
                                fileWriter.close();
                       // closing the file writer.
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                            output.close();// closing the output.
                        }

                        // again I write write to a file. This time i don't
use auto flush because  i want to overwrite.
                        // by overwriting, i save the last entry.
```

```
                        try {
                            File file1=new File("src/history/lastEntry/last-
entry-mortgage.txt");
                            FileWriter fileWriter1=new FileWriter(file1);


fileWriter1.write(Double.valueOf(txtMonthlyPay.getText()) + " ");

fileWriter1.write(Double.valueOf(txtHousePrice.getText()) + " ");

fileWriter1.write(Double.valueOf(txtInterestRate.getText()) + " ");

fileWriter1.write(Double.valueOf(txtLoanTerm.getText()) + " ");

fileWriter1.write(Double.valueOf(txtDownPayment.getText()) + "\n");
                            fileWriter1.close();

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }
                }
            }
        });

//-------------------------------------------------------------------------//
//-------------------------Clear button------------------------------------//

        Button btnClear = new Button("Clear");
        btnClear.setId("btnClear");
        btnClear.setPrefSize(70,35);
        btnClear.setLayoutX(40);
        btnClear.setLayoutY(260);
        btnClear.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                txtMonthlyPay.setText("");
                txtHousePrice.setText("");
                txtDownPayment.setText("");
                txtInterestRate.setText("");
                txtLoanTerm.setText("");

            }
        });

//-------------------------------------------------------------------------//
//-------------------------Back button-------------------------------------//

        Button btnBack = new Button("Go Back");
        btnBack.setId("btn2");
        btnBack.setPrefSize(60,20);
        btnBack.setLayoutX(10);
        btnBack.setLayoutY(15);
        btnBack.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //closing the current stage
                Stage stage = (Stage) btnBack.getScene().getWindow();
```

```
                stage.close();

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(440);
                primaryStage.setMaxWidth(420);
                primaryStage.setMinHeight(440);
                primaryStage.setMinWidth(420);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(MainMenu.Mainmenu());
                primaryStage.show();
            }
        });

//-------------------------------------------------------------------------//
//-------------------------Help button-------------------------------------//

        Button btnHelp = new Button("Help");
        btnHelp.setId("btn2");
        btnHelp.setPrefSize(60,20);
        btnHelp.setLayoutX(360);
        btnHelp.setLayoutY(15);
        btnHelp.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(550);
                primaryStage.setMaxWidth(500);
                primaryStage.setMinHeight(550);
                primaryStage.setMinWidth(500);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(HelpView.Mortgage());
                primaryStage.show();
            }
        });

//-------------------------------------------------------------------------//
//-------------------------History button-----------------------------------//

        Button btnHistory = new Button("History");
        btnHistory.setId("btn2");
        btnHistory.setPrefSize(60,20);
        btnHistory.setLayoutX(430);
        btnHistory.setLayoutY(15);
        btnHistory.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                HistoryDisplay.history("src/history/history-mortgage.txt");

            }
        });
```

```
//-------------------------------------------------------------------------//
// -----------------Adding to anchorPane and stage display-----------------//


        AnchorPane root =new AnchorPane();
        root.getChildren().addAll(lblHeading,inputs,
inputType,keyboardGrid,btnCalculate,btnClear,btnBack, btnHistory,btnHelp);
        root.setId("root");

        Scene scene=new Scene(root,500,400);
        scene.getStylesheets().add("css/layout.css");// link to external css
        return scene;

    }

}
//*************************************************************************
*************************************************************************//
```

Loan Calculator

```
package com.hirunz2000;


import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

import java.io.*;
import java.time.LocalDateTime;
import java.time.format.DateTimeFormatter;
import java.util.InputMismatchException;
import java.util.Scanner;


public class LoanCalculatorWindow {
    public static Scene LoanCalculator() {
        //display keyboard before clicking a textfield and  not allows to
eenter.
        GridPane keyboardGrid=new GridPane();
        Keyboard.keyboardDisplay(keyboardGrid,null);

        Label lblHeading =new Label("Auto Loan Calculator");
        lblHeading.setPrefSize(498,55);
        lblHeading.setLayoutX(3);
        lblHeading.setLayoutY(3);
        lblHeading.setId("label-heading");

//---------------------------Input Labels ---------------------------------//

        Label lblMonthlyPay =new Label("Monthly Pay");
        lblMonthlyPay.setPrefSize(105,21);
        lblMonthlyPay.setId("label1");

        Label lblAutoLoan =new Label("Auto Loan");
        lblAutoLoan.setPrefSize(105,21);
        lblAutoLoan.setId("label1");

        Label lblLoanTerm =new Label("Loan Term");
        lblLoanTerm.setPrefSize(105,21);
        lblLoanTerm.setId("label1");

        Label lblInterestRate =new Label("Interest Rate");
        lblInterestRate.setPrefSize(105,21);
        lblInterestRate.setId("label1");

//-------------------------------------------------------------------------//
//-------------------------Input TextFields--------------------------------//

        TextField txtMonthlyPay = new TextField();
        txtMonthlyPay.setPrefSize(176,25);
```

```
        txtMonthlyPay.setId("txt");
        txtMonthlyPay.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtMonthlyPay);
            }
        });

        TextField txtAutoLoan = new TextField();
        txtAutoLoan.setPrefSize(176,25);
        txtAutoLoan.setId("txt");
        txtAutoLoan.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtAutoLoan);
            }
        });

        TextField txtLoanTerm = new TextField();
        txtLoanTerm.setPrefSize(176,25);
        txtLoanTerm.setId("txt");
        txtLoanTerm.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtLoanTerm);
            }
        });

        TextField txtInterestRate = new TextField();
        txtInterestRate.setPrefSize(176,25);
        txtInterestRate.setId("txt");
        txtInterestRate.setOnMouseClicked(new EventHandler<MouseEvent>() {
            @Override
            public void handle(MouseEvent event) {
                Keyboard.keyboardDisplay(keyboardGrid,txtInterestRate);
            }
        });

//-------------------------------------------------------------------------------//
//-----------------------Add Inputs to a gridpane--------------------------//

        GridPane inputGrid=new GridPane();
        inputGrid.setPrefSize(288,226);
        inputGrid.setVgap(5);
        inputGrid.setLayoutX(35);
        inputGrid.setLayoutY(100);

        inputGrid.add(lblMonthlyPay,0,0);
        inputGrid.add(lblAutoLoan,0,1);
        inputGrid.add(lblLoanTerm,0,2);
        inputGrid.add(lblInterestRate,0,3);


        inputGrid.add(txtMonthlyPay,1,0);
        inputGrid.add(txtAutoLoan,1,1);
        inputGrid.add(txtLoanTerm,1,2);
        inputGrid.add(txtInterestRate,1,3);
```

```
//-------------------------------------------------------------------------//
//--------------------------Retrieving Last Entry--------------------------//

        Double[] lastEntries = new Double[4];
        try {
            try {

                File file = new File("src/history/lastEntry/last-entry-
loan.txt");

                Scanner input = new Scanner(file);
                for (int i = 0; i < lastEntries.length; i++) {
                    lastEntries[i]=input.nextDouble();
                }
                input.close();
            } catch (FileNotFoundException e) {
                System.out.println("File not found");
            }
        }catch (InputMismatchException e){
        // to handle the error when the file is empty
            System.out.println("File is empty.");;
        }
        txtMonthlyPay.setText(String.valueOf(lastEntries[0]));
        txtAutoLoan.setText(String.valueOf(lastEntries[1]));
        txtInterestRate.setText(String.valueOf(lastEntries[2]));
        txtLoanTerm.setText(String.valueOf(lastEntries[3]));


//-------------------------------------------------------------------------//
//--------------------------Input Type Labels------------------------------//

        Label dollar1=new Label("$");
        dollar1.setPrefSize(15,25);
        dollar1.setId("label1");

        Label dollar2=new Label("$");
        dollar2.setPrefSize(15,25);
        dollar2.setId("label1");

        Label period=new Label("yrs");
        period.setPrefSize(20,25);
        period.setId("label1");

        Label rate=new Label("%");
        rate.setPrefSize(15,25);
        rate.setId("label1");

// -------------------------------------------------------------------------//
//-------------------------Input Type Grid---------------------------------//

        GridPane inputType=new GridPane();
        inputType.setPrefSize(256,226);
        inputType.setVgap(5);
        inputType.setLayoutX(295);
        inputType.setLayoutY(100);

        inputType.add(dollar1,0,0);
        inputType.add(dollar2,0,1);
```

```
        inputType.add(period,0,2);
        inputType.add(rate,0,3);

//-------------------------------------------------------------------------//
// ----------------------Calculate button---------------------------------//

        Button btnCalculate = new Button("Calculate");
        btnCalculate.setId("btnCalculate");
        btnCalculate.setPrefSize(84,35);
        btnCalculate.setLayoutX(210);
        btnCalculate.setLayoutY(260);
        btnCalculate.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                String str = "%.2f";
               int arrSize=4;
                // assigning textfield text to  array to validate
                String[] arr=new String[arrSize];
                arr[0]=txtMonthlyPay.getText();
                arr[1]=txtAutoLoan.getText();
                arr[2]=txtInterestRate.getText();
                arr[3]=txtLoanTerm.getText();

                //finds out how many empty strings are there in the array
                int EmptyStringCount
=Validations.EmptyStringCount(arr,arrSize);
                if(EmptyStringCount >1 | EmptyStringCount ==0){
    //if empty number count is greater than 1 or less than 1 cannot calculate
                    Validations.InputErrorAlert("");

                }
                else if(arr[2].isEmpty()){
                    Validations.EmptyInterestAlert();
                }
                else {
              //if there is only one empty index, this block will execute.
                    //then takes finds out the empty string index.
                    int EmptyStringIndex = Validations.EmptyStringIndex(arr,
arrSize);
                    if (Validations.doubleInputCheck(arr)) {
                        // according to the empty string index, the relevant
code block will execute.
                       //creating formulae object andthen pass the relevant
arguments to the method and get the result.
                        // finally sets the value to the empty textfield.

                        if (EmptyStringIndex == 0) {
                            double P = Double.valueOf(arr[1] );
                            double r = Double.valueOf(arr[2]);
                            double t = Double.valueOf(arr[3]);
                            Formulae Loan = new Formulae();
                            double result = Loan.PMT(P, r, t);
                            txtMonthlyPay.setText(String.format(str,
result));

                        }
                        else if (EmptyStringIndex ==1){
                            double PMT = Double.valueOf(arr[0]);
                            double r = Double.valueOf(arr[2]);
```

```
                            double t = Double.valueOf(arr[3]);
                            Formulae Loan = new Formulae();
                            double result = Loan.LoanAmount(PMT, t, r);
                            double totalInterest=(PMT*(t * Formulae.n) -
result);
                            txtAutoLoan.setText(String.format(str, result ));
                        }
                        else {
                            double P = Double.valueOf(arr[1] );
                            double r = Double.valueOf(arr[2]);
                            double PMT = Double.valueOf(arr[0]);
                            Formulae Loan = new Formulae();
                            double result = Loan.LoanTerm(P, PMT, r);
                            txtLoanTerm.setText(String.format(str,result));
                        }

//-----------------------------file handling-------------------------------//
        //once the above code block is executed, writing to the file.
        //to make the history more sense, i added date and time to the entry.

                        //date and time
                        DateTimeFormatter dateTimeFormatter =
DateTimeFormatter.ofPattern("yyyy/MM/dd HH:mm:ss");
                        LocalDateTime currentTime = LocalDateTime.now();

                        //write to file begins.
                        FileWriter fileWriter =null;
                        PrintWriter output =null;
                        File file=new File("src/history/history-loan.txt");

                        try{

                            fileWriter =new FileWriter(file,true);
                            output =new PrintWriter(fileWriter,true);


output.write("*************************************************\n");
                            output.write("***********
Entry:"+dateTimeFormatter.format(currentTime)+"  ***********\n");

output.write("*************************************************\n");

                            output.write(String.format("Monthly Pay  :
$%12s\n",Double.valueOf(txtMonthlyPay.getText())));
                            output.write(String.format("Home Price   :
$%12s\n",Double.valueOf(txtAutoLoan.getText())));
                            output.write(String.format("Interest Rate:   %10s
%%\n",Double.valueOf(txtInterestRate.getText())));
                            output.write(String.format("Loan Term    :   %8s
yrs\n",Double.valueOf(txtLoanTerm.getText())));


output.write("*************************************************\n\n\n");

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
```

```
                        finally {
                            try {
                                fileWriter.close();
                // closing the file writer.
                            } catch (IOException e) {
                                e.printStackTrace();
                            }
                            output.close();// closing the output.
                        }

                        // again I write write to a file. This time i don't
use auto flush because  i want to overwrite.
                        // by overwriting, i save the last entry.

                        try {
                            File file1=new File("src/history/lastEntry/last-
entry-loan.txt");
                            FileWriter fileWriter1=new FileWriter(file1);


fileWriter1.write(Double.valueOf(txtMonthlyPay.getText()) + " ");

fileWriter1.write(Double.valueOf(txtAutoLoan.getText()) + " ");

fileWriter1.write(Double.valueOf(txtInterestRate.getText()) + " ");

fileWriter1.write(Double.valueOf(txtLoanTerm.getText()) + " ");

                            fileWriter1.close();

                        } catch (IOException e) {
                            e.printStackTrace();
                        }
                    }

                }


            }
        });

//-------------------------------------------------------------------------//
//----------------------Clear button---------------------------------------//

        Button btnClear = new Button("Clear");
        btnClear.setId("btnClear");
        btnClear.setPrefSize(70,35);
        btnClear.setLayoutX(40);
        btnClear.setLayoutY(260);
        btnClear.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                txtMonthlyPay.setText("");
                txtAutoLoan.setText("");
                txtInterestRate.setText("");
                txtLoanTerm.setText("");
            }
        });
```

```
//-----------------------------------------------------------------------------//
//---------------------------Back button---------------------------------------//

        Button btnBack = new Button("Go Back");
        btnBack.setId("btn2");
        btnBack.setPrefSize(60,20);
        btnBack.setLayoutX(10);
        btnBack.setLayoutY(15);
        btnBack.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                //closing the current stage
                Stage stage = (Stage) btnBack.getScene().getWindow();
                stage.close();

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(440);
                primaryStage.setMaxWidth(420);
                primaryStage.setMinHeight(440);
                primaryStage.setMinWidth(420);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(MainMenu.Mainmenu());
                primaryStage.show();

            }
        });

//-----------------------------------------------------------------------------//
//-----------------------Help button-------------------------------------------//

        Button btnHelp = new Button("Help");
        btnHelp.setId("btn2");
        btnHelp.setPrefSize(60,20);
        btnHelp.setLayoutX(360);
        btnHelp.setLayoutY(15);
        btnHelp.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {

                Stage primaryStage=new Stage();
                //setting a fix size for stage
                primaryStage.setMaxHeight(550);
                primaryStage.setMaxWidth(500);
                primaryStage.setMinHeight(550);
                primaryStage.setMinWidth(500);

                primaryStage.setTitle("All in One Finance Calculator");
                primaryStage.setScene(HelpView.Loan());
                primaryStage.show();

            }
        });
```

```
//-------------------------------------------------------------------------//
//-------------------------History button-----------------------------------//

        Button btnHistory = new Button("History");
        btnHistory.setId("btn2");
        btnHistory.setPrefSize(60,20);
        btnHistory.setLayoutX(430);
        btnHistory.setLayoutY(15);
        btnHistory.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                HistoryDisplay.history("src/history/history-loan.txt");

            }
        });

//-------------------------------------------------------------------------//
// ----------------Adding to anchorPane and stage display------------------//

        AnchorPane root =new AnchorPane();

root.getChildren().addAll(lblHeading,inputGrid,inputType,btnCalculate,keyboar
dGrid,btnClear, btnBack,btnHelp,btnHistory);
        root.setId("root");

        Scene scene=new Scene(root,500,400);
        scene.getStylesheets().add("css/layout.css");// link to external css
        return scene;


    }
}
//*************************************************************************
*************************************************************************//
```

History Display Class

This class contains a method that will scan the history file get all lines to an arraylist.
Then I declared a label and appended arraylist values to it in a for loop.
That way, I can get the whole history into one single label.
Then I put that label inside a scroll pane.

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.control.ScrollPane;
import javafx.scene.layout.AnchorPane;
import javafx.stage.Stage;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.InputMismatchException;
import java.util.Scanner;

public class HistoryDisplay {
    public static void history(String url){
        // arraylist to get all the lines in history
        ArrayList<String> arr=new ArrayList<>();

        //label to store the all arraylist data
        Label lbl=new Label();
        lbl.setPrefWidth(435);
        lbl.setId("historyDisplay1");

//-------------------- Reading The history file----------------------------//
        try {
            try {

                File file = new File(url);
                Scanner input = new Scanner(file);
                while(input.hasNext()){
                    arr.add(input.nextLine());
                }

                for (int i=0;i<arr.size();i++){
                    lbl.setText(lbl.getText()+"\n"+arr.get(i));
                }

                input.close();
            } catch (FileNotFoundException e) {
                System.out.println("File not found");
            }
        }catch (InputMismatchException e){
            System.out.println("File is empty.");;
        }
```

```
//-------------------------------------------------------------------------//
//---------------------- Label and back button----------------------------//

        Label lblHeading =new Label("History");
        lblHeading.setPrefSize(479,55);
        lblHeading.setLayoutX(3);
        lblHeading.setLayoutY(3);
        lblHeading.setId("label-heading");


        Button btnBack = new Button("Go Back");
        btnBack.setId("btn2");
        btnBack.setPrefSize(60,20);
        btnBack.setLayoutX(10);
        btnBack.setLayoutY(15);
        btnBack.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage stage = (Stage) btnBack.getScene().getWindow();
                stage.close();
            }
        });

//-------------------------------------------------------------------------//
//----------Adding the label to a scrollpane and then displaying-----------//

        ScrollPane scrollPane=new ScrollPane();
        scrollPane.setId("scroll");
        scrollPane.setContent(lbl);
        scrollPane.setPrefSize(450,350);
        scrollPane.setLayoutX(15);
        scrollPane.setLayoutY(80);

        AnchorPane root=new AnchorPane();
        root.setPrefSize(500,500);
        root.getChildren().addAll(lblHeading,scrollPane,btnBack);
        root.setId("root");

        Stage primaryStage=new Stage();
        //setting a fix size for stage
        primaryStage.setMaxHeight(500);
        primaryStage.setMaxWidth(500);
        primaryStage.setMinHeight(500);
        primaryStage.setMinWidth(500);


        Scene scene=new Scene(root,500,500);
        scene.getStylesheets().add("css/layout.css");//link to external css
file
        primaryStage.setTitle("All in One Finance Calculator");
        primaryStage.setScene(scene);
        primaryStage.showAndWait();
    }
}
//*************************************************************************
*************************************************************************//
```

Help View Class

Help view has four methods with labels customized for each calculator.

```
package com.hirunz2000;

import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.control.Label;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.GridPane;
import javafx.stage.Stage;

public class HelpView {
    private static char ch1 ='\u29BF';// unicode character for bullet

//****************This class contains all four help windows ************** //


//-------------------------------------------------------------------------//
//********************** Fixed Deposit Calculator ***********************//
//-------------------------------------------------------------------------//
    public static Scene FD(){

        Label lblheading =new Label("Help View - Fixed Deposit Calculator");
        lblheading.setPrefSize(479,55);
        lblheading.setLayoutX(3);
        lblheading.setLayoutY(3);
        lblheading.setId("label-heading");

        Label lblQuestion1 = new Label("What can this calculator calculate
?");
        lblQuestion1.setPrefSize(250,21);
        lblQuestion1.setLayoutX(24);
        lblQuestion1.setLayoutY(76);
        lblQuestion1.setId("question");

        Label lblPrincipleAmount = new Label(ch1 +" Principle Amount");
        lblPrincipleAmount.setPrefSize(150,20);
        lblPrincipleAmount.setId("answer");

        Label lblFutureValue = new Label(ch1 +" Future Value");
        lblFutureValue.setPrefSize(150,20);
        lblFutureValue.setId("answer");

        Label lblInterestRate = new Label(ch1 +" Interest Rate");
        lblInterestRate.setPrefSize(150,20);
        lblInterestRate.setId("answer");

        Label lblTimePeriod = new Label(ch1 +" Time Period");
        lblTimePeriod.setPrefSize(150,20);
        lblTimePeriod.setId("answer");

        //adding above labels to a gridpane
```

```
        GridPane gridQuestion1 =new GridPane();
        gridQuestion1.setPrefSize(130,90);
        gridQuestion1.setLayoutX(50);
        gridQuestion1.setLayoutY(105);

        gridQuestion1.add(lblPrincipleAmount,0,0);
        gridQuestion1.add(lblFutureValue,0,1);
        gridQuestion1.add(lblInterestRate,0,2);
        gridQuestion1.add(lblTimePeriod,0,3);


        Label lblQuestion2 = new Label("How to calculate ?");
        lblQuestion2.setPrefSize(235,21);
        lblQuestion2.setLayoutX(24);
        lblQuestion2.setLayoutY(215);
        lblQuestion2.setId("question");

        Label lblInstruction1 = new Label(ch1 +" Insert all the other fields
except the one which you wish to calculate.");
        lblInstruction1.setPrefSize(410,20);
        lblInstruction1.setId("answer");

        Label lblInstruction2 = new Label(ch1 +" Use the on screen keyboard
to avoid entering alphabetical values.");
        lblInstruction2.setPrefSize(410,20);
        lblInstruction2.setId("answer");

        Label lblInstruction3 = new Label(ch1 +" Click calculate button to
calculate.");
        lblInstruction3.setPrefSize(410,20);
        lblInstruction3.setId("answer");

        //adding above labels to a gridpane
        GridPane gridQuestion2 =new GridPane();
        gridQuestion2.setPrefSize(410,90);
        gridQuestion2.setLayoutX(50);
        gridQuestion2.setLayoutY(242);

        gridQuestion2.add(lblInstruction1,0,0);
        gridQuestion2.add(lblInstruction2,0,1);
        gridQuestion2.add(lblInstruction3,0,2);


        Label lblQuestion3 = new Label("What to enter ?");
        lblQuestion3.setPrefSize(235,21);
        lblQuestion3.setLayoutX(24);
        lblQuestion3.setLayoutY(341);
        lblQuestion3.setId("question");

        Label lblPrincipleAmount2 = new Label(ch1 +" Principle Amount");
        lblPrincipleAmount2.setPrefSize(150,20);
        lblPrincipleAmount2.setId("answer");

        Label lblFutureValue2 = new Label(ch1 +" Future Value");
        lblFutureValue2.setPrefSize(150,20);
        lblFutureValue2.setId("answer");

        Label lblInterestRate2 = new Label(ch1 +" Interest Rate");
        lblInterestRate2.setPrefSize(150,20);
```

```
        lblInterestRate2.setId("answer");

        Label lblTimePeriod2 = new Label(ch1 +" Time Period");
        lblTimePeriod2.setPrefSize(150,20);
        lblTimePeriod2.setId("answer");

        Label lblPrincipleAmount1 = new Label("Principle Investment");
        lblPrincipleAmount1.setPrefSize(200,20);
        lblPrincipleAmount1.setId("answer");

        Label lblFutureValue1 = new Label("Future value of the Principle
amount");
        lblFutureValue1.setPrefSize(200,20);
        lblFutureValue1.setId("answer");

        Label lblInterestRate1 = new Label("Annual Interest Rate");
        lblInterestRate1.setPrefSize(200,20);
        lblInterestRate1.setId("answer");

        Label lblTimePeriod1 = new Label("Time of investment in years");
        lblTimePeriod1.setPrefSize(200,20);
        lblTimePeriod1.setId("answer");

        //adding above labels to a gridpane
        GridPane gridQuestion3 =new GridPane();
        gridQuestion3.setPrefSize(350,95);
        gridQuestion3.setLayoutX(50);
        gridQuestion3.setLayoutY(370);

        gridQuestion3.add(lblPrincipleAmount2,0,0);
        gridQuestion3.add(lblFutureValue2,0,1);
        gridQuestion3.add(lblInterestRate2,0,2);
        gridQuestion3.add(lblTimePeriod2,0,3);

        gridQuestion3.add(lblPrincipleAmount1,1,0);
        gridQuestion3.add(lblFutureValue1,1,1);
        gridQuestion3.add(lblInterestRate1,1,2);
        gridQuestion3.add(lblTimePeriod1,1,3);

        //Close button to close this window
        Button btnClose = new Button("Close");
        btnClose.setFocusTraversable(false);
        btnClose.setId("btnClose");
        btnClose.setPrefSize(70,30);
        btnClose.setLayoutX(360);
        btnClose.setLayoutY(450);
        btnClose.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage stage = (Stage) btnClose.getScene().getWindow();
                stage.close();
            }
        });

        //add the above grids and labels to an anchorpane and setting the
scene
        AnchorPane root=new AnchorPane();

root.getChildren().addAll(lblheading,lblQuestion1,lblQuestion2,lblQuestion3,g
```

```
ridQuestion1,gridQuestion2,gridQuestion3,btnClose);
        root.setId("root");
        Scene scene=new Scene(root,500,550);
        scene.getStylesheets().add("css/layout.css");
        return scene;
    }


//-------------------------------------------------------------------------//
//********************** Savings Calculator ****************************//
//-------------------------------------------------------------------------//

    public static Scene Savings(){

        Label lblheading =new Label("Help View - Savings Calculator");
        lblheading.setPrefSize(479,55);
        lblheading.setLayoutX(3);
        lblheading.setLayoutY(3);
        lblheading.setId("label-heading");

        Label lblQuestion1 = new Label("What can this calculator calculate
?");
        lblQuestion1.setPrefSize(250,21);
        lblQuestion1.setLayoutX(24);
        lblQuestion1.setLayoutY(76);
        lblQuestion1.setId("question");

        Label lblPrincipleAmount = new Label(ch1 +" Principle Amount");
        lblPrincipleAmount.setPrefSize(150,20);
        lblPrincipleAmount.setId("answer");

        Label lblFutureValue = new Label(ch1 +" Future Value");
        lblFutureValue.setPrefSize(150,20);
        lblFutureValue.setId("answer");

        Label lblPMT = new Label(ch1 +" Monthly Payment");
        lblPMT.setPrefSize(150,20);
        lblPMT.setId("answer");

        Label lblTimePeriod = new Label(ch1 +" Time Period");
        lblTimePeriod.setPrefSize(150,20);
        lblTimePeriod.setId("answer");

        GridPane gridQuestion1 =new GridPane();
        gridQuestion1.setPrefSize(130,90);
        gridQuestion1.setLayoutX(50);
        gridQuestion1.setLayoutY(105);

        gridQuestion1.add(lblPrincipleAmount,0,0);
        gridQuestion1.add(lblFutureValue,0,1);
        gridQuestion1.add(lblPMT,0,2);
        gridQuestion1.add(lblTimePeriod,0,3);


        Label lblQuestion2 = new Label("How to calculate ?");
        lblQuestion2.setPrefSize(235,21);
        lblQuestion2.setLayoutX(24);
        lblQuestion2.setLayoutY(215);
        lblQuestion2.setId("question");
```

```
        Label lblInstruction1 = new Label(ch1 +" Insert all the other fields
except the one which you wish to calculate.");
        lblInstruction1.setPrefSize(410,20);
        lblInstruction1.setId("answer");

        Label lblInstruction2 = new Label(ch1 +" Use the on screen keyboard
to avoid entering alphabetical values.");
        lblInstruction2.setPrefSize(410,20);
        lblInstruction2.setId("answer");

        Label lblInstruction3 = new Label(ch1 +" Click calculate button to
calculate.");
        lblInstruction3.setPrefSize(410,20);
        lblInstruction3.setId("answer");

        Label lblInstruction4 = new Label(ch1 +" Cannot calculate interest in
this calculator.");
        lblInstruction4.setPrefSize(410,20);
        lblInstruction4.setId("answer");

        GridPane gridQuestion2 =new GridPane();
        gridQuestion2.setPrefSize(410,100);
        gridQuestion2.setLayoutX(50);
        gridQuestion2.setLayoutY(242);

        gridQuestion2.add(lblInstruction1,0,0);
        gridQuestion2.add(lblInstruction2,0,1);
        gridQuestion2.add(lblInstruction3,0,2);
        gridQuestion2.add(lblInstruction4,0,3);


        Label lblQuestion3 = new Label("What to enter ?");
        lblQuestion3.setPrefSize(235,21);
        lblQuestion3.setLayoutX(24);
        lblQuestion3.setLayoutY(341);
        lblQuestion3.setId("question");

        Label lblPrincipleAmount2 = new Label(ch1 +" Principle Amount");
        lblPrincipleAmount2.setPrefSize(150,20);
        lblPrincipleAmount2.setId("answer");

        Label lblFutureValue2 = new Label(ch1 +" Future Value");
        lblFutureValue2.setPrefSize(150,20);
        lblFutureValue2.setId("answer");

        Label lblPMT2 = new Label(ch1 +" Monthly Payment");
        lblPMT2.setPrefSize(150,20);
        lblPMT2.setId("answer");

        Label lblTimePeriod2 = new Label(ch1 +" Time Period");
        lblTimePeriod2.setPrefSize(150,20);
        lblTimePeriod2.setId("answer");

        Label lblInterestRate2 = new Label(ch1 +" Interest Rate");
        lblInterestRate2.setPrefSize(150,20);
        lblInterestRate2.setId("answer");
```

```
        Label lblPrincipleAmount1 = new Label("Principle Investment");
        lblPrincipleAmount1.setPrefSize(200,20);
        lblPrincipleAmount1.setId("answer");

        Label lblFutureValue1 = new Label("Future value of the Principle
amount");
        lblFutureValue1.setPrefSize(200,20);
        lblFutureValue1.setId("answer");

        Label lblPMT1 = new Label("Monthly payment amount");
        lblPMT1.setPrefSize(200,20);
        lblPMT1.setId("answer");

        Label lblTimePeriod1 = new Label("Time of investment in years");
        lblTimePeriod1.setPrefSize(200,20);
        lblTimePeriod1.setId("answer");

        Label lblInterestRate1 = new Label("Annual Interest Rate");
        lblInterestRate1.setPrefSize(200,20);
        lblInterestRate1.setId("answer");

        GridPane gridQuestion3 =new GridPane();
        gridQuestion3.setPrefSize(350,95);
        gridQuestion3.setLayoutX(50);
        gridQuestion3.setLayoutY(370);

        gridQuestion3.add(lblPrincipleAmount2,0,0);
        gridQuestion3.add(lblFutureValue2,0,1);
        gridQuestion3.add(lblPMT2,0,2);
        gridQuestion3.add(lblTimePeriod2,0,3);
        gridQuestion3.add(lblInterestRate2,0,4);

        gridQuestion3.add(lblPrincipleAmount1,1,0);
        gridQuestion3.add(lblFutureValue1,1,1);
        gridQuestion3.add(lblPMT1,1,2);
        gridQuestion3.add(lblTimePeriod1,1,3);
        gridQuestion3.add(lblInterestRate1,1,4);

        Button btnClose = new Button("Close");
        btnClose.setFocusTraversable(false);
        btnClose.setId("btnClose");
        btnClose.setPrefSize(70,30);
        btnClose.setLayoutX(360);
        btnClose.setLayoutY(450);
        btnClose.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage stage = (Stage) btnClose.getScene().getWindow();
                stage.close();
            }
        });

        AnchorPane root=new AnchorPane();

root.getChildren().addAll(lblheading,lblQuestion1,lblQuestion2,lblQuestion3,g
ridQuestion1,gridQuestion2,gridQuestion3,btnClose);
        root.setId("root");
        Scene scene=new Scene(root,500,550);
        scene.getStylesheets().add("css/layout.css");
```

```
        return scene;
    }


//-------------------------------------------------------------------------//
//********************** Mortgage Calculator ****************************//
//-------------------------------------------------------------------------//

    public static Scene Mortgage(){

        Label lblheading =new Label("Help View - Mortgage Calculator");
        lblheading.setPrefSize(479,55);
        lblheading.setLayoutX(3);
        lblheading.setLayoutY(3);
        lblheading.setId("label-heading");

        Label lblQuestion1 = new Label("What can this calculator calculate
?");
        lblQuestion1.setPrefSize(250,21);
        lblQuestion1.setLayoutX(24);
        lblQuestion1.setLayoutY(76);
        lblQuestion1.setId("question");

        Label lblPrincipleAmount = new Label(ch1 +" Loan Amount");
        lblPrincipleAmount.setPrefSize(150,20);
        lblPrincipleAmount.setId("answer");


        Label lblPMT = new Label(ch1 +" Monthly Payment");
        lblPMT.setPrefSize(150,20);
        lblPMT.setId("answer");

        Label lblTimePeriod = new Label(ch1 +" Time Period");
        lblTimePeriod.setPrefSize(150,20);
        lblTimePeriod.setId("answer");

        GridPane gridQuestion1 =new GridPane();
        gridQuestion1.setPrefSize(130,90);
        gridQuestion1.setLayoutX(50);
        gridQuestion1.setLayoutY(105);

        gridQuestion1.add(lblPrincipleAmount,0,0);
        gridQuestion1.add(lblPMT,0,2);
        gridQuestion1.add(lblTimePeriod,0,3);


        Label lblQuestion2 = new Label("How to calculate ?");
        lblQuestion2.setPrefSize(235,21);
        lblQuestion2.setLayoutX(24);
        lblQuestion2.setLayoutY(180);
        lblQuestion2.setId("question");

        Label lblInstruction1 = new Label(ch1 +" Insert all the other fields
except the one which you wish to calculate.");
        lblInstruction1.setPrefSize(410,20);
        lblInstruction1.setId("answer");

        Label lblInstruction2 = new Label(ch1 +" Use the on screen keyboard
to avoid entering alphabetical values.");
```

```
        lblInstruction2.setPrefSize(410,20);
        lblInstruction2.setId("answer");

        Label lblInstruction3 = new Label(ch1 +" Click calculate button to
calculate.");
        lblInstruction3.setPrefSize(410,20);
        lblInstruction3.setId("answer");

        Label lblInstruction4 = new Label(ch1 +" Cannot calculate interest or
down payment in this calculator.");
        lblInstruction4.setPrefSize(410,20);
        lblInstruction4.setId("answer");

        Label lblInstruction5 = new Label(ch1 +" If down payment is zero you
can leave it empty or put zero.");
        lblInstruction5.setPrefSize(410,20);
        lblInstruction5.setId("answer");

        GridPane gridQuestion2 =new GridPane();
        gridQuestion2.setPrefSize(410,125);
        gridQuestion2.setLayoutX(50);
        gridQuestion2.setLayoutY(212);

        gridQuestion2.add(lblInstruction1,0,0);
        gridQuestion2.add(lblInstruction2,0,1);
        gridQuestion2.add(lblInstruction3,0,2);
        gridQuestion2.add(lblInstruction4,0,3);
        gridQuestion2.add(lblInstruction5,0,4);


        Label lblQuestion3 = new Label("What to enter ?");
        lblQuestion3.setPrefSize(235,21);
        lblQuestion3.setLayoutX(24);
        lblQuestion3.setLayoutY(341);
        lblQuestion3.setId("question");

        Label lblHousePrice2 = new Label(ch1 +" House Price");
        lblHousePrice2.setPrefSize(150,20);
        lblHousePrice2.setId("answer");

        Label lblDownPayment2 = new Label(ch1 +" Down Payment");
        lblDownPayment2.setPrefSize(150,20);
        lblDownPayment2.setId("answer");

        Label lblPMT2 = new Label(ch1 +" Monthly Payment");
        lblPMT2.setPrefSize(150,20);
        lblPMT2.setId("answer");

        Label lblTimePeriod2 = new Label(ch1 +" Time Period");
        lblTimePeriod2.setPrefSize(150,20);
        lblTimePeriod2.setId("answer");

        Label lblInterestRate2 = new Label(ch1 +" Interest Rate");
        lblInterestRate2.setPrefSize(150,20);
        lblInterestRate2.setId("answer");


        Label lblHousePrice1 = new Label("Full price including down
payment");
```

```
        lblHousePrice1.setPrefSize(250,20);
        lblHousePrice1.setId("answer");

        Label lblDownPayment1 = new Label("Down payment");
        lblDownPayment1.setPrefSize(200,20);
        lblDownPayment1.setId("answer");

        Label lblPMT1 = new Label("Monthly payment amount");
        lblPMT1.setPrefSize(200,20);
        lblPMT1.setId("answer");

        Label lblTimePeriod1 = new Label("Time of investment in years");
        lblTimePeriod1.setPrefSize(200,20);
        lblTimePeriod1.setId("answer");

        Label lblInterestRate1 = new Label("Annual Interest Rate");
        lblInterestRate1.setPrefSize(200,20);
        lblInterestRate1.setId("answer");

        GridPane gridQuestion3 =new GridPane();
        gridQuestion3.setPrefSize(350,95);
        gridQuestion3.setLayoutX(50);
        gridQuestion3.setLayoutY(370);

        gridQuestion3.add(lblHousePrice2,0,0);
        gridQuestion3.add(lblDownPayment2,0,1);
        gridQuestion3.add(lblPMT2,0,2);
        gridQuestion3.add(lblTimePeriod2,0,3);
        gridQuestion3.add(lblInterestRate2,0,4);

        gridQuestion3.add(lblHousePrice1,1,0);
        gridQuestion3.add(lblDownPayment1,1,1);
        gridQuestion3.add(lblPMT1,1,2);
        gridQuestion3.add(lblTimePeriod1,1,3);
        gridQuestion3.add(lblInterestRate1,1,4);

        Button btnClose = new Button("Close");
        btnClose.setFocusTraversable(false);
        btnClose.setId("btnClose");
        btnClose.setPrefSize(70,30);
        btnClose.setLayoutX(360);
        btnClose.setLayoutY(450);
        btnClose.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage stage = (Stage) btnClose.getScene().getWindow();
                stage.close();
            }
        });

        AnchorPane root=new AnchorPane();

root.getChildren().addAll(lblheading,lblQuestion1,lblQuestion2,lblQuestion3,g
ridQuestion1,gridQuestion2,gridQuestion3,btnClose);
        root.setId("root");
        Scene scene=new Scene(root,500,550);
        scene.getStylesheets().add("css/layout.css");
        return scene;
    }
```

```
//-------------------------------------------------------------------------//
//************************** Loan Calculator ****************************//
//-------------------------------------------------------------------------//

    public static Scene Loan(){

        Label lblheading =new Label("Help View - Loan Calculator");
        lblheading.setPrefSize(479,55);
        lblheading.setLayoutX(3);
        lblheading.setLayoutY(3);
        lblheading.setId("label-heading");

        Label lblQuestion1 = new Label("What can this calculator calculate
?");
        lblQuestion1.setPrefSize(250,21);
        lblQuestion1.setLayoutX(24);
        lblQuestion1.setLayoutY(76);
        lblQuestion1.setId("question");

        Label lblPrincipleAmount = new Label(ch1 +" Loan Amount");
        lblPrincipleAmount.setPrefSize(150,20);
        lblPrincipleAmount.setId("answer");


        Label lblPMT = new Label(ch1 +" Monthly Payment");
        lblPMT.setPrefSize(150,20);
        lblPMT.setId("answer");

        Label lblTimePeriod = new Label(ch1 +" Time Period");
        lblTimePeriod.setPrefSize(150,20);
        lblTimePeriod.setId("answer");

        GridPane gridQuestion1 =new GridPane();
        gridQuestion1.setPrefSize(140,90);
        gridQuestion1.setLayoutX(50);
        gridQuestion1.setLayoutY(105);

        gridQuestion1.add(lblPrincipleAmount,0,0);
        gridQuestion1.add(lblPMT,0,2);
        gridQuestion1.add(lblTimePeriod,0,3);


        Label lblQuestion2 = new Label("How to calculate ?");
        lblQuestion2.setPrefSize(250,21);
        lblQuestion2.setLayoutX(24);
        lblQuestion2.setLayoutY(180);
        lblQuestion2.setId("question");

        Label lblInstruction1 = new Label(ch1 +" Insert all the other fields
except the one which you wish to calculate.");
        lblInstruction1.setPrefSize(410,20);
        lblInstruction1.setId("answer");

        Label lblInstruction2 = new Label(ch1 +" Use the on screen keyboard
to avoid entering alphabetical values.");
        lblInstruction2.setPrefSize(410,20);
        lblInstruction2.setId("answer");
```

```
        Label lblInstruction3 = new Label(ch1 +" Click calculate button to
calculate.");
        lblInstruction3.setPrefSize(410,20);
        lblInstruction3.setId("answer");

        Label lblInstruction4 = new Label(ch1 +" Cannot calculate interest in
this calculator.");
        lblInstruction4.setPrefSize(410,20);
        lblInstruction4.setId("answer");


        GridPane gridQuestion2 =new GridPane();
        gridQuestion2.setPrefSize(410,110);
        gridQuestion2.setLayoutX(50);
        gridQuestion2.setLayoutY(212);

        gridQuestion2.add(lblInstruction1,0,0);
        gridQuestion2.add(lblInstruction2,0,1);
        gridQuestion2.add(lblInstruction3,0,2);
        gridQuestion2.add(lblInstruction4,0,3);


        Label lblQuestion3 = new Label(" What to enter ?");
        lblQuestion3.setPrefSize(235,21);
        lblQuestion3.setLayoutX(24);
        lblQuestion3.setLayoutY(321);
        lblQuestion3.setId("question");

        Label lblLoanAmount2 = new Label(ch1 +" Loan Amount");
        lblLoanAmount2.setPrefSize(150,20);
        lblLoanAmount2.setId("answer");

        Label lblPMT2 = new Label(ch1 +" Monthly Payment");
        lblPMT2.setPrefSize(150,20);
        lblPMT2.setId("answer");

        Label lblTimePeriod2 = new Label(ch1 +" Time Period");
        lblTimePeriod2.setPrefSize(150,20);
        lblTimePeriod2.setId("answer");

        Label lblInterestRate2 = new Label(ch1 +" Interest Rate");
        lblInterestRate2.setPrefSize(150,20);
        lblInterestRate2.setId("answer");


        Label lblLoanAmunt1 = new Label("Loan Amount");
        lblLoanAmunt1.setPrefSize(250,20);
        lblLoanAmunt1.setId("answer");

        Label lblPMT1 = new Label("Monthly payment amount");
        lblPMT1.setPrefSize(200,20);
        lblPMT1.setId("answer");

        Label lblTimePeriod1 = new Label("Time of investment in years");
        lblTimePeriod1.setPrefSize(200,20);
        lblTimePeriod1.setId("answer");

        Label lblInterestRate1 = new Label("Annual Interest Rate");
```

```
        lblInterestRate1.setPrefSize(200,20);
        lblInterestRate1.setId("answer");

        GridPane gridQuestion3 =new GridPane();
        gridQuestion3.setPrefSize(350,95);
        gridQuestion3.setLayoutX(50);
        gridQuestion3.setLayoutY(350);

        gridQuestion3.add(lblLoanAmount2,0,0);
        gridQuestion3.add(lblPMT2,0,1);
        gridQuestion3.add(lblTimePeriod2,0,2);
        gridQuestion3.add(lblInterestRate2,0,3);

        gridQuestion3.add(lblLoanAmunt1,1,0);
        gridQuestion3.add(lblPMT1,1,1);
        gridQuestion3.add(lblTimePeriod1,1,2);
        gridQuestion3.add(lblInterestRate1,1,3);

        Button btnClose = new Button("Close");
        btnClose.setFocusTraversable(false);
        btnClose.setId("btnClose");
        btnClose.setPrefSize(70,30);
        btnClose.setLayoutX(360);
        btnClose.setLayoutY(450);
        btnClose.setOnAction(new EventHandler<ActionEvent>() {
            @Override
            public void handle(ActionEvent event) {
                Stage stage = (Stage) btnClose.getScene().getWindow();
                stage.close();
            }
        });

        AnchorPane root=new AnchorPane();

root.getChildren().addAll(lblheading,lblQuestion1,lblQuestion2,lblQuestion3,g
ridQuestion1,gridQuestion2,gridQuestion3,btnClose);
        root.setId("root");
        Scene scene=new Scene(root,500,550);
        scene.getStylesheets().add("css/layout.css");
        return scene;
    }

}
//*****************************************************************************
*****************************************************************************//
```

CSS Styles

I used an external stylesheet to add CSS to my calculator.

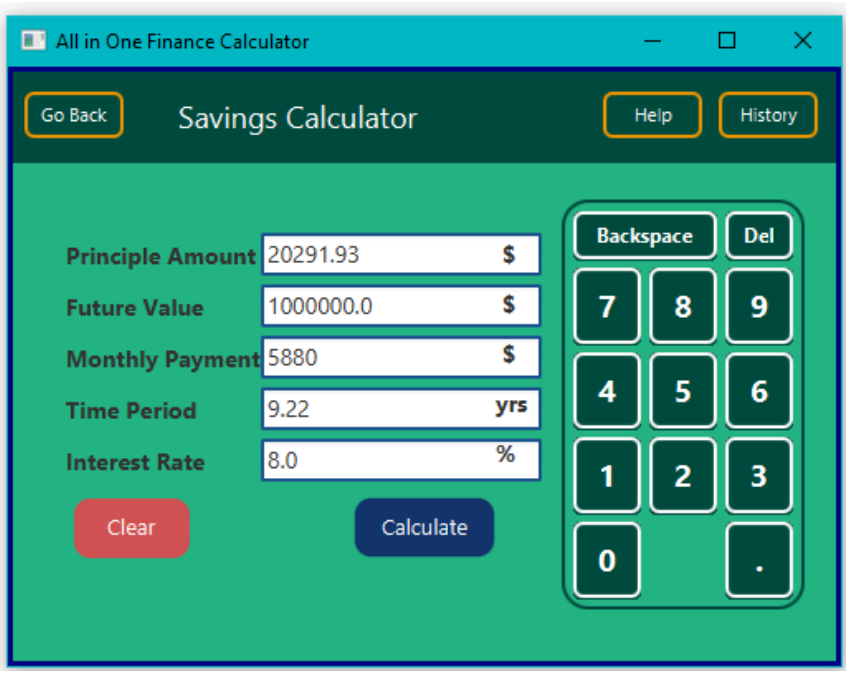
```
/* background layout */
#root{
	-fx-border-color: #000080;
	-fx-border-width: 3;
	-fx-background-color:#23b382;
}

/* main menu image */

#image{
	-fx-background-color: transparent;
}

/* main menu exit button */
#btnExit{
	-fx-background-color:  #edf0ee;
	-fx-font-size: 13px;
	-fx-text-alignment: center;
	-fx-wrap-text: true;
	-fx-font-weight: bold;
	-fx-background-radius: 10px;
}

/* for all 4 calculators and help menus and history views */
#label-heading{
	-fx-stroke-width: 2;
	-fx-stroke-color: black;
	-fx-font-size: 18px;
	-fx-text-fill: white;
	-fx-padding: 10px 10px 10px 100px;
	-fx-background-color:#004a3d;
	-fx-text-alignment: center;

}

#heading1{
	-fx-font-size: 20px;
	-fx-font-weight: bold;
	-fx-text-alignment: center;
	-fx-text-fill: white;
}
```

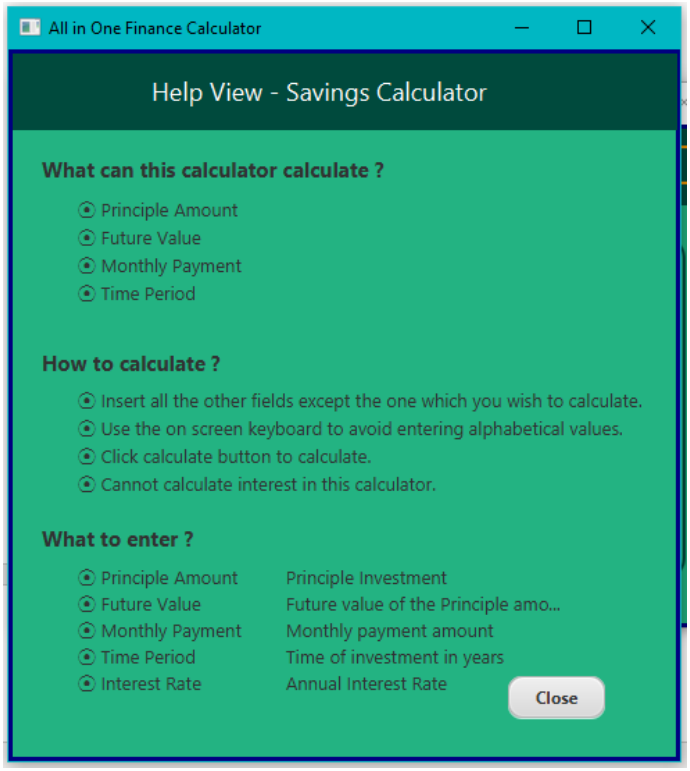
```
#btn1{
        -fx-font-weight: bold;
        -fx-background-color: #edf0ee;
        -fx-font-size: 13px;
        -fx-text-alignment: center;
        -fx-wrap-text: true;
        -fx-background-radius: 10px;
        -fx-effect: dropshadow(gaussian, rgba(1,75,60,1), 10, 0.2, 0px, 2px);

}

/* zoom in buttons when mouse moves over it */
Button:hover{
        -fx-scale-x: 1.1;
        -fx-scale-y: 1.1;
        -fx-scale-z: 1.0;
}


#label1{
        -fx-font-size: 14px;
        -fx-font-weight: bold;
}

#txt{
         -fx-font-size: 14px;
         -fx-padding: 1,1,1,1;
         -fx-border-color: #0f4f87;
         -fx-border-width: 2;
         -fx-border-radius: 1;
}


#btn2{
        -fx-font-size: 11px;
        -fx-text-alignment: center;
        -fx-border-radius: 5px;
        -fx-border-color: #f09601;
        -fx-border-width: 2;
        -fx-text-fill: white;
        -fx-background-color: transparent;
}

#btnClear{
        -fx-font-size: 13px;
        -fx-text-alignment: center;
        -fx-background-radius: 10px;
        -fx-text-fill: white;
```

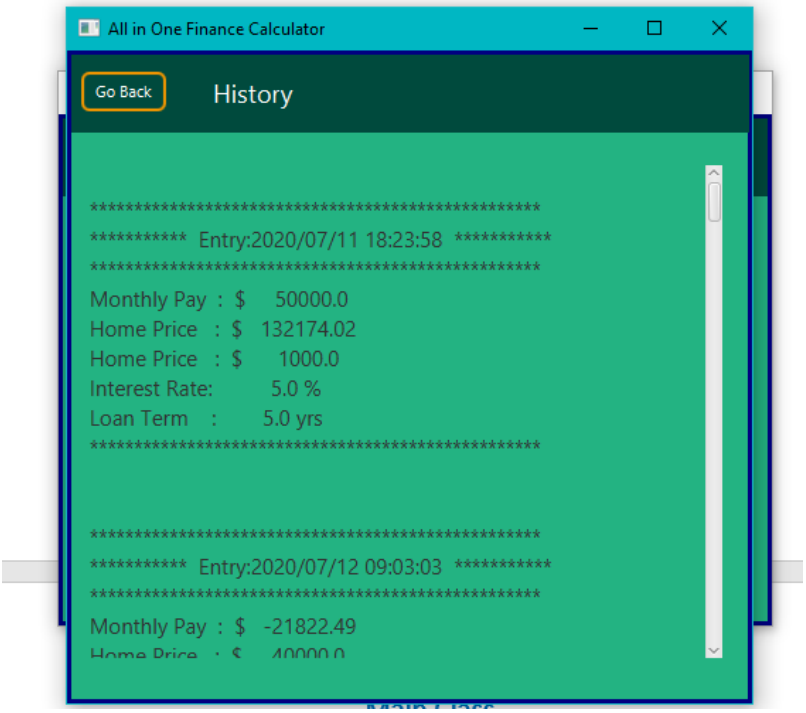
```
        -fx-background-color: #d05151;
}

#btnCalculate{
        -fx-font-size: 13px;
        -fx-text-alignment: center;
        -fx-background-radius: 10px;
        -fx-text-fill: white;
        -fx-background-color: #12346b;
}

#historyDisplay1{
        -fx-background-color:#23b382;
        -fx-font-size: 15px;
}
#scroll{
        -fx-background-color:#23b382;
}

#keyboard{
        -fx-padding: 5px;
        -fx-border-color: #004a3d;
        -fx-border-width: 2;
        -fx-border-radius: 15px;
        -fx-background-radius: 15px;
}


#backSP{
        -fx-font-size: 12px;
        -fx-font-weight: bold;
        -fx-background-radius: 5px;
        -fx-background-color:#004a3d;
        -fx-text-fill: white;
        -fx-border-color:white;
        -fx-border-radius: 5px;
        -fx-border-width: 2;
}


#period{
        -fx-font-size: 19px ;
        -fx-font-weight: bold;
        -fx-background-radius: 5px;
        -fx-background-color:#004a3d;
        -fx-text-fill: white;
        -fx-border-color:white;
        -fx-border-radius: 5px;
```

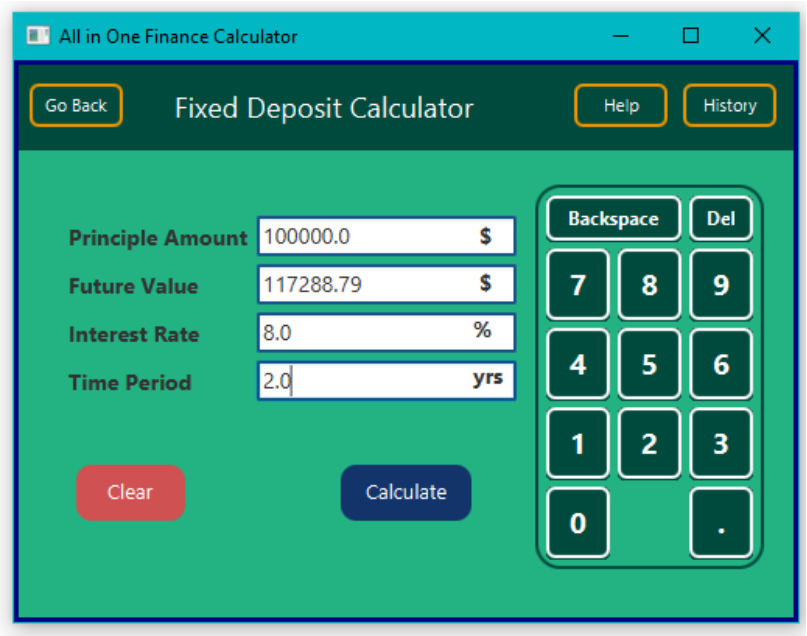
```
        -fx-border-width: 2;

}

#keyboardButton{
        -fx-font-size: 19px;
        -fx-font-weight: bold;
        -fx-background-radius: 5px;
        -fx-background-color:#004a3d;
        -fx-text-fill: white;
        -fx-border-color:white;
        -fx-border-radius: 5px;
        -fx-border-width: 2;
}

#question{
        -fx-font-weight: bold;
        -fx-font-size: 15px;
}

#answer{
        -fx-font-size: 13px;
}
#btnClose{
        -fx-font-size: 12px;
        -fx-text-alignment: center;
        -fx-wrap-text: true;
        -fx-font-weight: bold;
        -fx-background-radius: 10px;

}
/*****************************************************************/
```

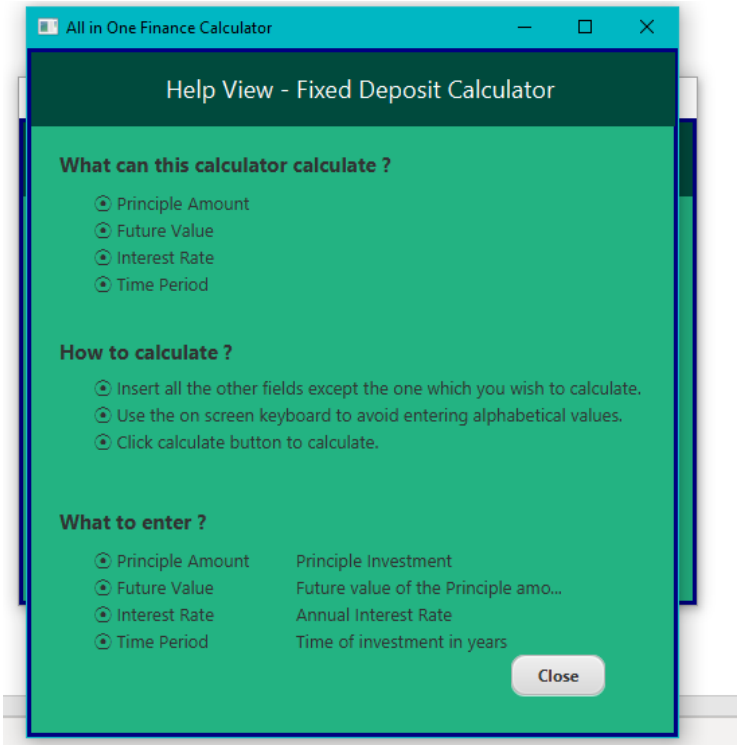
Screenshots

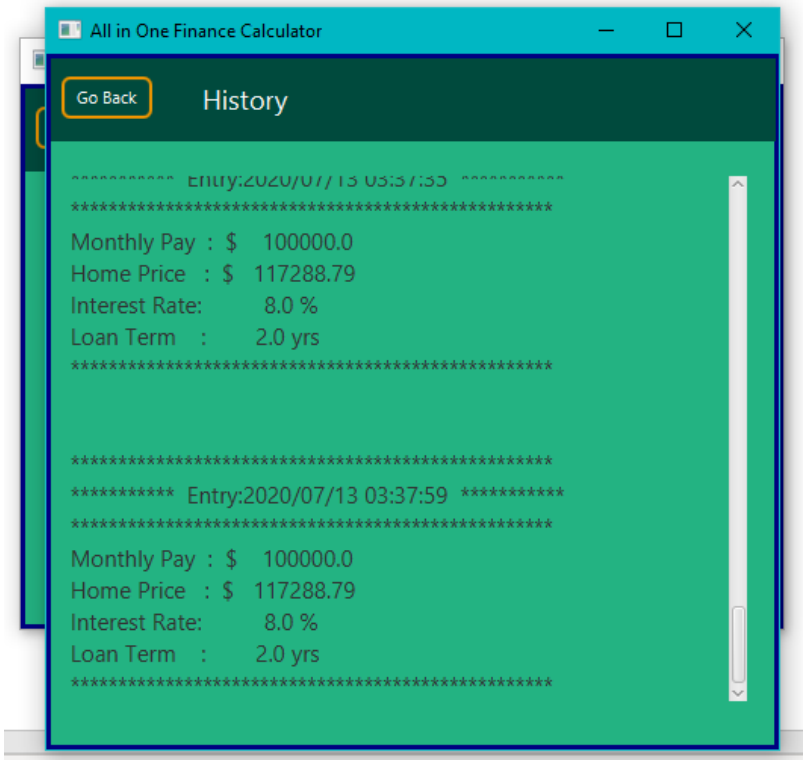


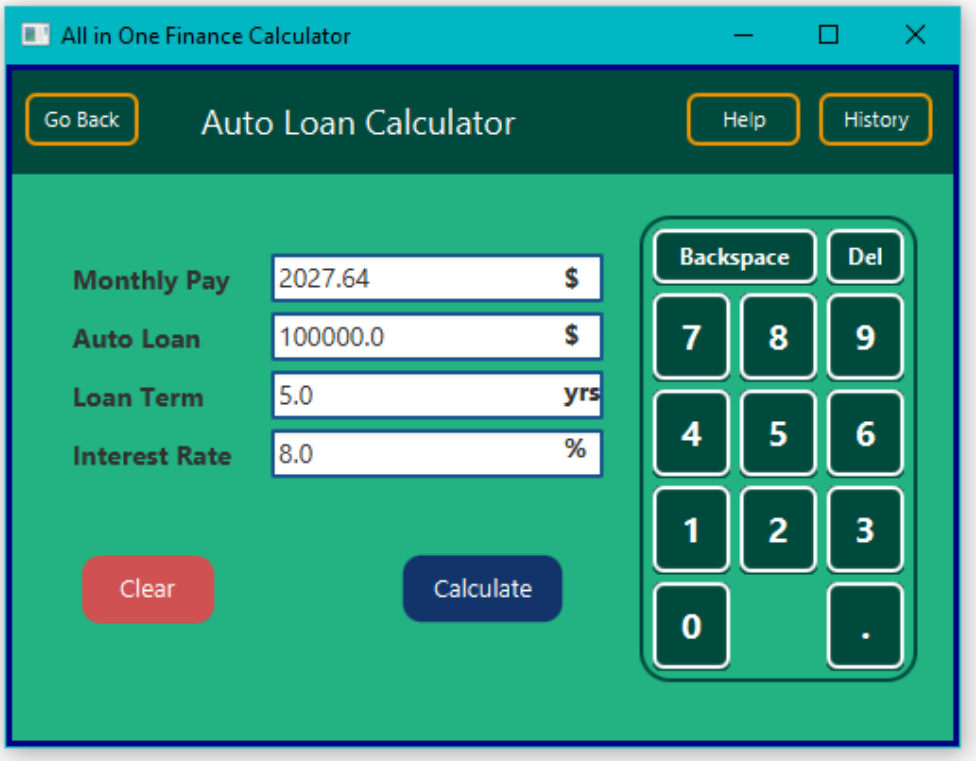

All in One Finance Calculator
Help View - Savings Calculator
What can this calculator calculate ?
Principle Amount
Future Value
Monthly Payment
Time Period
How to calculate ?
Insert all the other fields except the one which you wish to calculate.
Use the on screen keyboard to avoid entering alphabetical values.
Click calculate button to calculate.
Cannot calculate interest in this calculator.
What to enter ?
Principle Amount
Principle Investment
Future Value
Future value of the Principle amo...
Monthly Payment
Monthly payment amount
Time Period
Time of investment in years
Interest Rate
Annual Interest Rate
Close

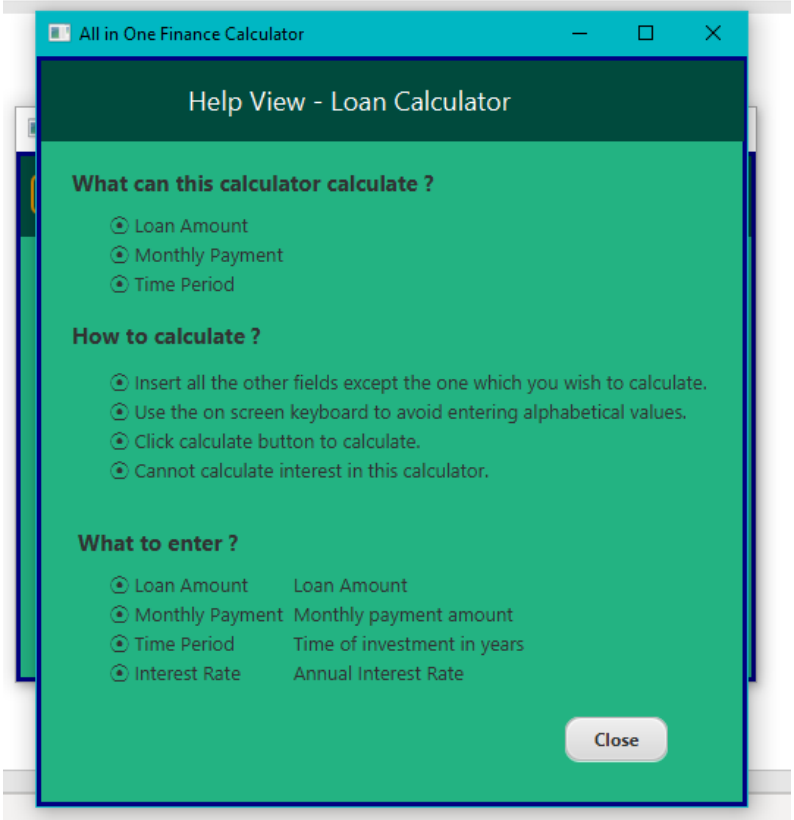

All in One Finance Calculator
Go Back
History
****
********** Entry:2020/07/11 18:23:58 **********
****
Monthly Pay : $ 50000.0
Home Price : $ 132174.02
Home Price : $ 1000.0
Interest Rate: 5.0 %
Loan Term : 5.0 yrs
****
****
********** Entry:2020/07/12 09:03:03 **********
****
Monthly Pay : $ -21822.49

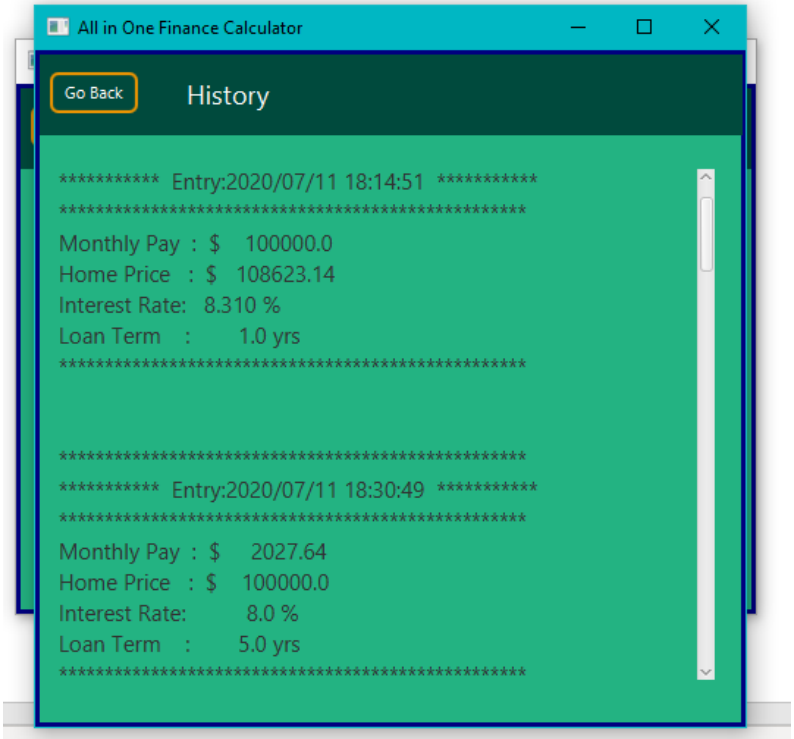

All in One Finance Calculator
Go Back
Fixed Deposit Calculator
Help
History
Principle Amount
100000.0
$
Future Value
117288.79
$
Interest Rate
8.0
%
Time Period
2.0
yrs
Clear
Calculate
Backspace
Del
7
8
9
4
5
6
1
2
3
0
.

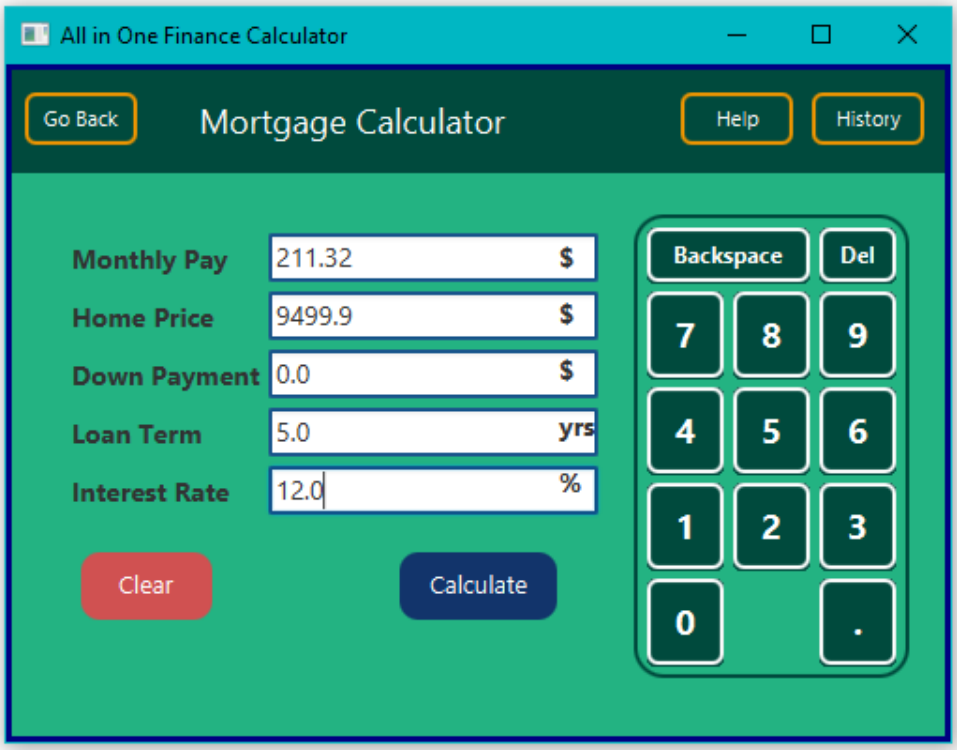

All in One Finance Calculator
Help View - Fixed Deposit Calculator
What can this calculator calculate ?
Principle Amount
Future Value
Interest Rate
Time Period
How to calculate ?
Insert all the other fields except the one which you wish to calculate.
Use the on screen keyboard to avoid entering alphabetical values.
Click calculate button to calculate.
What to enter ?
Principle Amount
Principle Investment
Future Value
Future value of the Principle amo...
Interest Rate
Annual Interest Rate
Time Period
Time of investment in years
Close

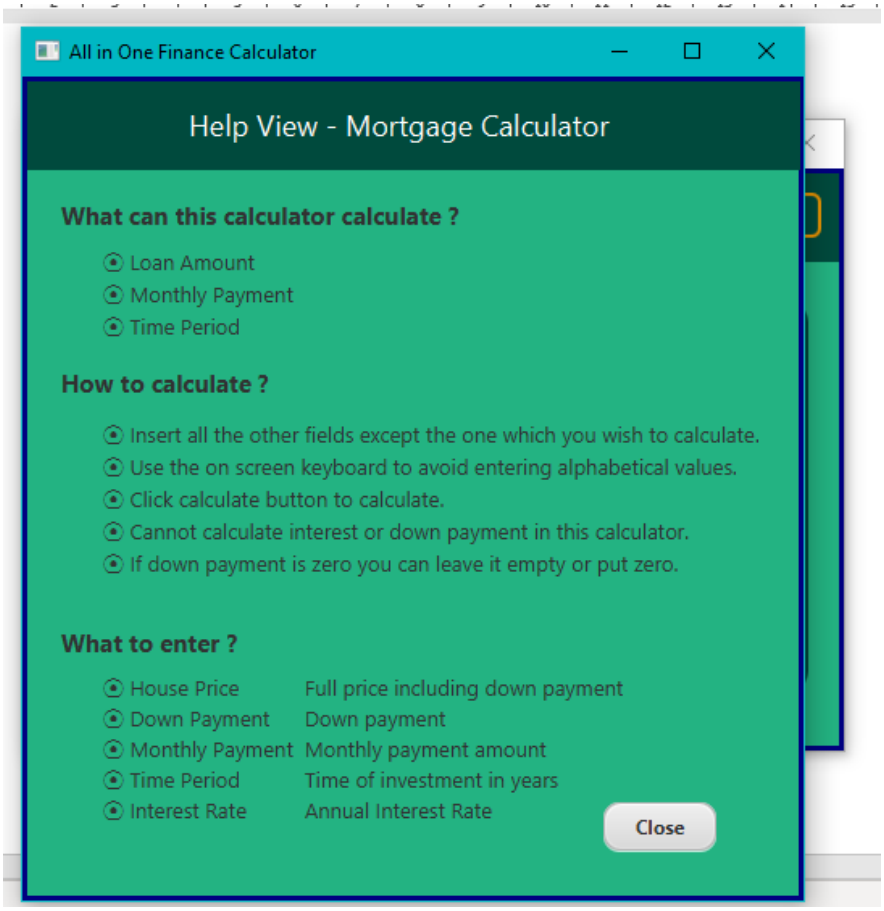

All in One Finance Calculator
Go Back
History
Monthly Pay : $ 100000.0
Home Price : $ 117288.79
Interest Rate: 8.0 %
Loan Term : 2.0 yrs
********** Entry:2020/07/13 03:37:59 **********
Monthly Pay : $ 100000.0
Home Price : $ 117288.79
Interest Rate: 8.0 %
Loan Term : 2.0 yrs

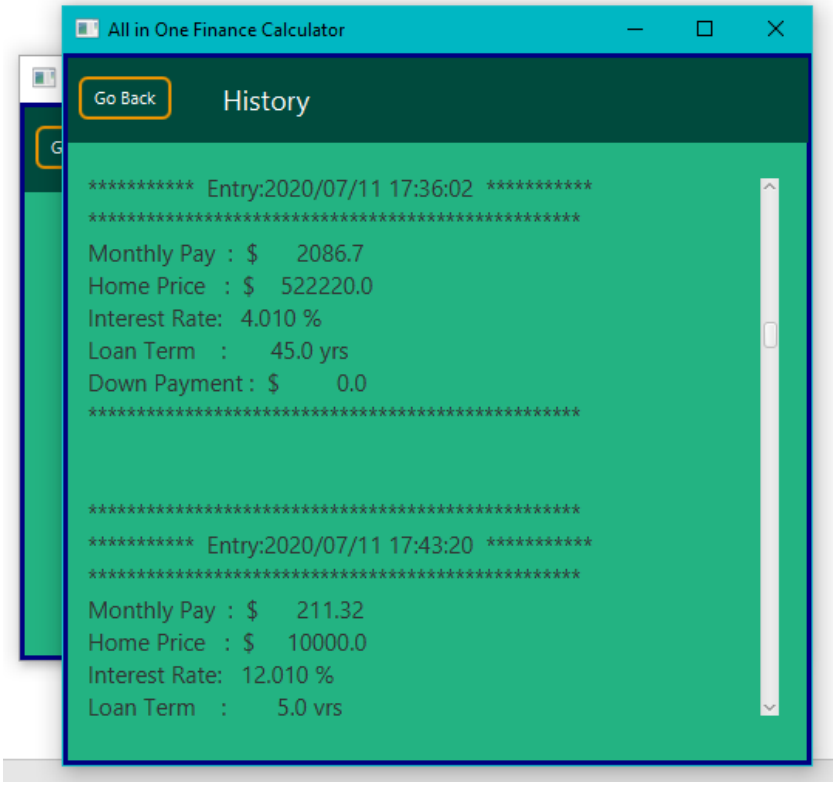

All in One Finance Calculator
Go Back
Auto Loan Calculator
Help
History
Monthly Pay
2027.64
$
Auto Loan
100000.0
$
Loan Term
5.0
yrs
Interest Rate
8.0
%
Clear
Calculate
Backspace
Del
7
8
9
4
5
6
1
2
3
0
.

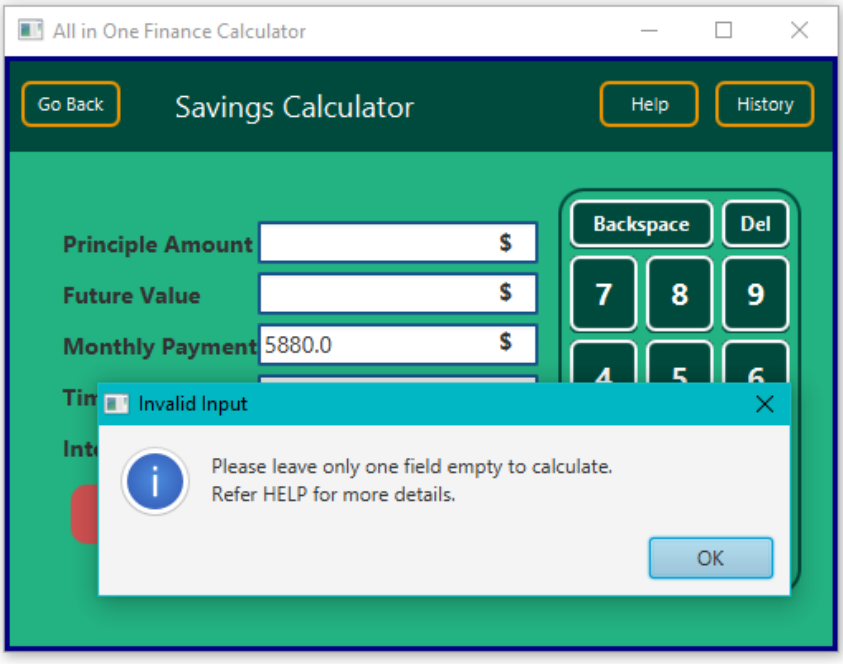

All in One Finance Calculator
Help View - Loan Calculator
What can this calculator calculate ?
Loan Amount
Monthly Payment
Time Period
How to calculate ?
Insert all the other fields except the one which you wish to calculate.
Use the on screen keyboard to avoid entering alphabetical values.
Click calculate button to calculate.
Cannot calculate interest in this calculator.
What to enter ?
Loan Amount Loan Amount
Monthly Payment Monthly payment amount
Time Period Time of investment in years
Interest Rate Annual Interest Rate
Close


All in One Finance Calculator
Go Back
History
********** Entry:2020/07/11 18:14:51 **********
Monthly Pay : $ 100000.0
Home Price : $ 108623.14
Interest Rate: 8.310 %
Loan Term : 1.0 yrs
********** Entry:2020/07/11 18:30:49 **********
Monthly Pay : $ 2027.64
Home Price : $ 100000.0
Interest Rate: 8.0 %
Loan Term : 5.0 yrs

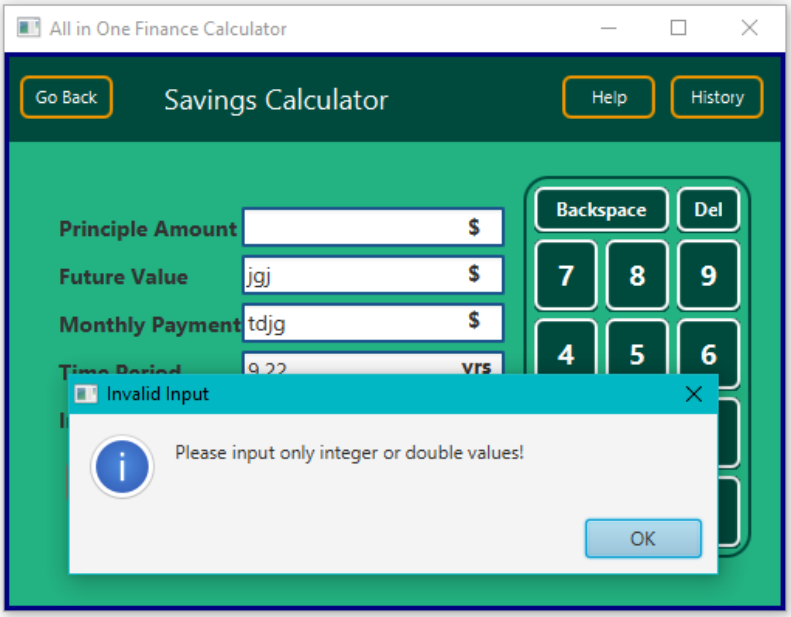

All in One Finance Calculator
Go Back
Mortgage Calculator
Help
History
Monthly Pay
211.32
$
Home Price
9499.9
$
Down Payment
0.0
$
Loan Term
5.0
yrs
Interest Rate
12.0
%
Clear
Calculate
Backspace
Del
7
8
9
4
5
6
1
2
3
0
.

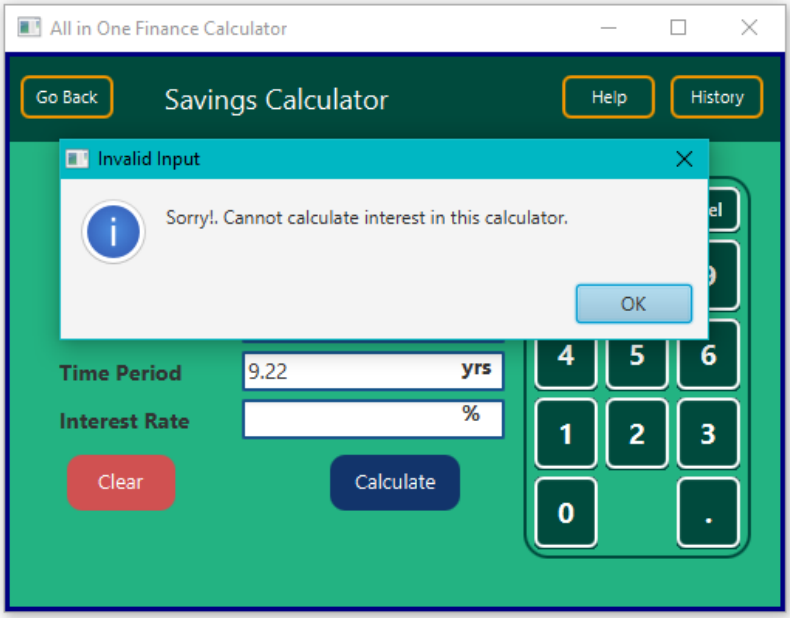

All in One Finance Calculator
Help View - Mortgage Calculator
What can this calculator calculate ?
Loan Amount
Monthly Payment
Time Period
How to calculate ?
Insert all the other fields except the one which you wish to calculate.
Use the on screen keyboard to avoid entering alphabetical values.
Click calculate button to calculate.
Cannot calculate interest or down payment in this calculator.
If down payment is zero you can leave it empty or put zero.
What to enter ?
House Price Full price including down payment
Down Payment Down payment
Monthly Payment Monthly payment amount
Time Period Time of investment in years
Interest Rate Annual Interest Rate
Close


All in One Finance Calculator
Go Back
History
********** Entry:2020/07/11 17:36:02 **********
Monthly Pay : $ 2086.7
Home Price : $ 522220.0
Interest Rate: 4.010 %
Loan Term : 45.0 yrs
Down Payment : $ 0.0
********** Entry:2020/07/11 17:43:20 **********
Monthly Pay : $ 211.32
Home Price : $ 10000.0
Interest Rate: 12.010 %
Loan Term : 5.0 vrs

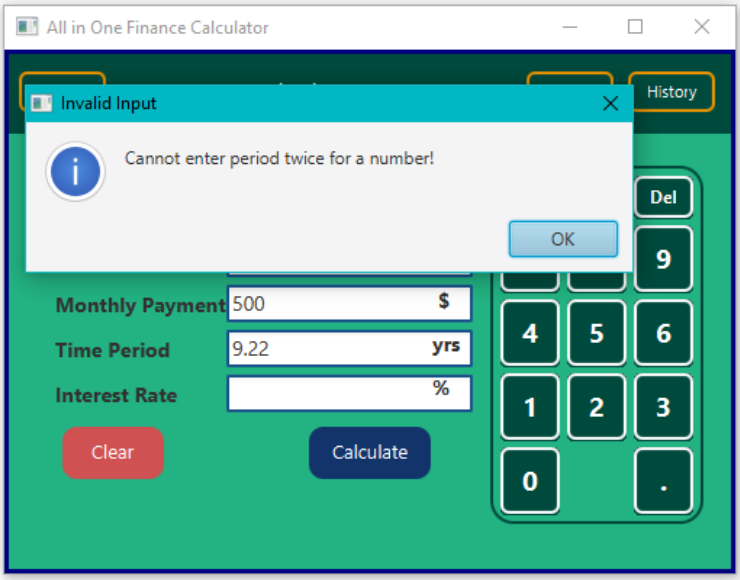

All in One Finance Calculator
Go Back
Savings Calculator
Help
History
Principle Amount
Future Value
Monthly Payment
5880.0
$
Backspace
Del
7
8
9
Invalid Input
Please leave only one field empty to calculate.
Refer HELP for more details.
OK

2


All in One Finance Calculator
Go Back
Savings Calculator
Help
History
Principle Amount
$
Future Value
jgj
$
Monthly Payment
tdjg
$
Backspace
Del
7
8
9
4
5
6
Invalid Input
Please input only integer or double values!
OK


All in One Finance Calculator
Go Back
Savings Calculator
Help
History
Invalid Input
Sorry!. Cannot calculate interest in this calculator.
OK
Time Period
9.22
yrs
Interest Rate
%
Clear
Calculate
4
5
6
1
2
3
0
.

2


All in One Finance Calculator
History
Invalid Input
Cannot enter period twice for a number!
OK
Del
9
Monthly Payment
500
$
Time Period
9.22
yrs
Interest Rate
%
4
5
6
1
2
3
Clear
Calculate
0
.

Conclusion

Calculator is working fine and behaving exactly as I expected. Using methods and reduced code duplication and breaking down to classes made the code more readable and user-friendly. Although it is working fine, my code has some duplicates which I could remove with more methods instead of copy pasting.