1598230452

Unit 15 – Object Oriented Programming

Task 5 Technical Document

Checklist

|  |  |
| --- | --- |
|  | Done? |
| Table of Contents | Y |
| Introduction – overview of program |  |
| Designs – use case – flow of events/diagram |  |
| Code |  |
| Test log |  |
| Test analysis |  |
| Evaluation |  |

Table of Contents

[Introduction 1](#_Toc472435055)

[Designs 1](#_Toc472435056)

[Code 1](#_Toc472435057)

[Test Log 1](#_Toc472435058)

[Test Analysis 2](#_Toc472435059)

[Evaluation 2](#_Toc472435060)

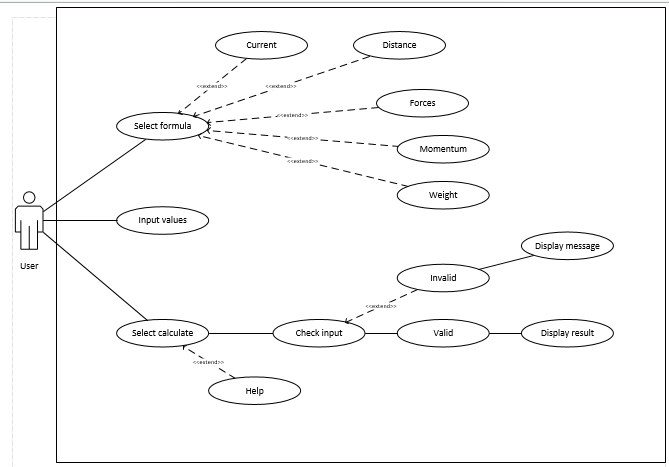
# Introduction

This program is designed for sixth form students who are studying physics. The purpose of the program is to make physics easy to learn through the use of a simple formula program. The program will be a downloadable app that students can use on their phone while in class or doing revision.

The program contains five of the most common formulas used in physics which are: current, force, momentum, distance and weight. Also, to help the program become simple and easy to learn there is a help page which guides the user through the program.

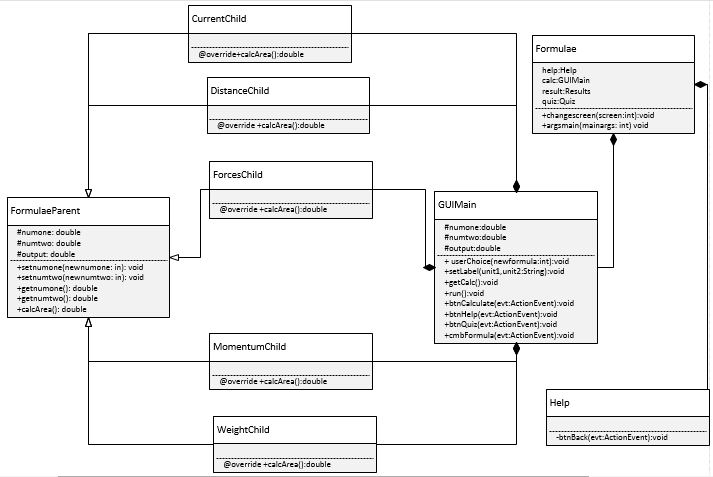
# Designs

***Use Case Diagram:***



This diagram shows the flow of events as the user goes through the program. Firstly, the user must select a formula in which they have five options: Current, Distance, Forces, Momentum, Weight. Once this is done they must input two values into the program. After that they will click calculate but if they are unsure of what to do they can click the help button which goes to the help screen. If they have chosen ‘select calculate’ the program will check the validity of the two values inputted. If the values are valid then it will display the result and if they are invalid an error message will be displayed.

***Class Diagram:***



This diagram starts at the ‘Formulae’ class. From here they have the ability to change screens through the method ‘changescreen’. Then the program can go in two directions, it can go to the ‘Help’ class or the ‘GUIMain’ class. Going to the help class will display the help screen to the user and from there they can chose to go back to the main method. Going back to the ‘GUIMain’ class the user will select the formula, enter the values and click calculate. The next class that is executed depends on the formula selected by the user. Also, for each of these classes to be ran they need to inherit data from the FormulaeParent because this class gets and sets the values entered.

# Code

*Formulae class*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class Formulae {

//setting up objects

static GUIMain calc = new GUIMain();

static Help help = new Help();

static Results result = new Results();

static Quiz quiz = new Quiz();

/\*\*

\* @param args the command line arguments

\*/

//Main Method

public static void main(String[] args) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

calc.setVisible(true);

}

});

}

//setting up a method which will allow the navigation through the GUI

public static void changescreen(int screen) {

//Each case has one object set to true which is the screen it will go

//to

switch (screen) {

case 0:

calc.setVisible(false);

result.setVisible(true);

quiz.setVisible(false);

help.setVisible(false);

break;

case 1:

calc.setVisible(true);

result.setVisible(false);

quiz.setVisible(false);

help.setVisible(false);

break;

case 2:

calc.setVisible(false);

result.setVisible(false);

quiz.setVisible(true);

help.setVisible(false);

break;

case 3:

calc.setVisible(false);

result.setVisible(false);

quiz.setVisible(false);

help.setVisible(true);

break;

default:

calc.setVisible(true);

break;

}

}

}

*FormulaeParent*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class FormulaeParent {

//Creating variables which will hold the value that the user enters

protected double numone = 0;

protected double numtwo = 0;

public FormulaeParent() {

}

//Below are get and set methods which get the value the user enters and

//put them in to the calculation

public void setnumone (double in) {

numone = in;

}

public void setnumtwo (double in) {

numtwo = in;

}

public double getnumone() {

return numone;

}

public double getnumtwo() {

return numtwo;

}

public double calcArea() {

return 0;

}

}

*CurrentChild*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class CurrentChild extends FormulaeParent {

//the word extends creates inheritance as this class is getting data from

//FormulaeParent

public CurrentChild (double in1, double in2) {

setnumone (in1);

setnumtwo (in2);

}

@Override

public double calcArea() {

return (numone / numtwo);

//How the equation is calculated

}

}

*DistanceChild*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class DistanceChild extends FormulaeParent {

//the word extends creates inheritance as this class is getting data from

//FormulaeParent

public DistanceChild (double in1, double in2) {

setnumone (in1);

setnumtwo (in2);

}

@Override

public double calcArea() {

return (numone \* numtwo);

//How the equation is calculated

}

}

*ForcesChild*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class ForcesChild extends FormulaeParent {

//the word extends creates inheritance as this class is getting data from

//FormulaeParent

public ForcesChild (double in1, double in2) {

setnumone (in1);

setnumtwo (in2);

}

@Override

public double calcArea() {

return (numone \* numtwo);

//How the equation is calculated

}

}

*MomentumChild*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class MomentumChild extends FormulaeParent {

//the word extends creates inheritance as this class is getting data from

//FormulaeParent

public MomentumChild (double in1, double in2) {

setnumone (in1);

setnumtwo (in2);

}

@Override

public double calcArea() {

return (numone \* numtwo);

//How the equation is calculated

}

}

*WeightChild*

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class WeightChild extends FormulaeParent {

//the word extends creates inheritance as this class is getting data from

//FormulaeParent

public WeightChild (double in1, double in2) {

setnumone (in1);

setnumtwo (in2);

}

@Override

public double calcArea() {

//How the equation is calculated

return (numone \* numtwo);

}

}

*GUIMain*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package formulae;

import java.util.HashSet;

/\*\*

\*

\* @author Chris

\*/

public class GUIMain extends javax.swing.JFrame {

//creating objects for the child classes which is just a copy of the class

CurrentChild currentchild = new CurrentChild(0,0);

DistanceChild distancechild = new DistanceChild(0,0);

ForcesChild forceschild = new ForcesChild(0,0);

MomentumChild momentumchild = new MomentumChild(0,0);

WeightChild weightchild = new WeightChild(0,0);

//here i am creating a container

FormulaeParent formulaeparent;

/\*\*

\* Creates new form GUIMain

\*/

public GUIMain() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

lblTitle = new javax.swing.JLabel();

cmbFormula = new javax.swing.JComboBox<String>();

lblInput1 = new javax.swing.JLabel();

lblInput2 = new javax.swing.JLabel();

txtInput1 = new javax.swing.JTextField();

txtInput2 = new javax.swing.JTextField();

btnCalculate = new javax.swing.JButton();

btnHelp = new javax.swing.JButton();

btnQuiz = new javax.swing.JButton();

txtOutput = new javax.swing.JTextField();

lblOutput = new javax.swing.JLabel();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

lblTitle.setText("Formulae");

cmbFormula.setModel(new javax.swing.DefaultComboBoxModel(new String[] { "Current", "Forces", "Momentum", "Distance", "Weight" }));

cmbFormula.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

cmbFormulaActionPerformed(evt);

}

});

lblInput1.setText("Voltage");

lblInput2.setText("Watts");

btnCalculate.setText("Calculate");

btnCalculate.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnCalculateActionPerformed(evt);

}

});

btnHelp.setText("Help");

btnHelp.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnHelpActionPerformed(evt);

}

});

btnQuiz.setText("Quiz");

btnQuiz.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnQuizActionPerformed(evt);

}

});

lblOutput.setText("Output");

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(lblInput1, javax.swing.GroupLayout.PREFERRED\_SIZE, 165, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(cmbFormula, javax.swing.GroupLayout.PREFERRED\_SIZE, 87, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(33, 33, 33))

.addComponent(txtInput1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 85, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(javax.swing.GroupLayout.Alignment.LEADING, layout.createSequentialGroup()

.addGap(161, 161, 161)

.addComponent(lblTitle))

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addComponent(btnQuiz)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(btnCalculate, javax.swing.GroupLayout.PREFERRED\_SIZE, 107, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(30, 30, 30))

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblInput2, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(lblOutput, javax.swing.GroupLayout.DEFAULT\_SIZE, 165, Short.MAX\_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(txtOutput)

.addComponent(txtInput2, javax.swing.GroupLayout.DEFAULT\_SIZE, 85, Short.MAX\_VALUE))))))

.addGap(33, 33, 33)

.addComponent(btnHelp)

.addGap(18, 18, 18))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(20, 20, 20)

.addComponent(lblTitle)

.addGap(40, 40, 40)

.addComponent(cmbFormula, javax.swing.GroupLayout.PREFERRED\_SIZE, 30, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(29, 29, 29)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtInput1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblInput1))

.addGap(24, 24, 24)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(lblInput2)

.addComponent(txtInput2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(txtOutput, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(lblOutput))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 42, Short.MAX\_VALUE)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(btnQuiz)

.addComponent(btnCalculate)

.addComponent(btnHelp)))

);

pack();

}// </editor-fold>

private void btnCalculateActionPerformed(java.awt.event.ActionEvent evt) {

//Here I am getting the getCalc method which gets the user inputs

//and then displays the results in the output box

//I have also called formulaeparent because this is where the equation

//is worked out.

getCalc(formulaeparent);

}

private void btnHelpActionPerformed(java.awt.event.ActionEvent evt) {

Formulae.changescreen(3);

//Here we are calling the method changescreen from the Formulae class

//The number in the brackets is which case is called

}

private void btnQuizActionPerformed(java.awt.event.ActionEvent evt) {

Formulae.changescreen(2);

//Here we are calling the method changescreen from the Formulae class

//The number in the brackets is which case is called

}

private void cmbFormulaActionPerformed(java.awt.event.ActionEvent evt) {

userChoice((Integer) cmbFormula.getSelectedIndex());

//This gets which option the user has selected from the combobox

//and then displays the two value eg speed and time next to the input

//box

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(GUIMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(GUIMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(GUIMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(GUIMain.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new GUIMain().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnCalculate;

private javax.swing.JButton btnHelp;

private javax.swing.JButton btnQuiz;

private javax.swing.JComboBox<String> cmbFormula;

private javax.swing.JLabel lblInput1;

private javax.swing.JLabel lblInput2;

private javax.swing.JLabel lblOutput;

private javax.swing.JLabel lblTitle;

private javax.swing.JTextField txtInput1;

private javax.swing.JTextField txtInput2;

private javax.swing.JTextField txtOutput;

// End of variables declaration

//Here we are going to match the forumla selected against each case

//The corresponding case will then be executed

public void userChoice(int formula) {

switch (formula) {

case 0:

//current

setLabel("Volatage", "Watts");

formulaeparent = currentchild;

break;

case 1:

//forces

setLabel("Mass", "Acceleration");

formulaeparent = forceschild;

break;

case 2:

//momentum

setLabel("Mass", "Velocity");

formulaeparent = momentumchild;

break;

case 3:

//distance

setLabel("Speed", "Time");

formulaeparent = distancechild;

break;

case 4:

//weight

setLabel("Mass", "Acceleration of Gravity");

formulaeparent = weightchild;

break;

}

}

//this method sets the user input as unit1 and unit2 so we can store it in

//memory

public void setLabel(String unit1, String unit2) {

lblInput1.setText(unit1);

lblInput2.setText(unit2);

}

//This

public void getCalc (FormulaeParent object) {

object.setnumone(Double.parseDouble(txtInput1.getText()));

object.setnumtwo(Double.parseDouble(txtInput2.getText()));

txtOutput.setText(Double.toString(object.calcArea()));

}

}

*Help*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class Help extends javax.swing.JFrame {

/\*\*

\* Creates new form Help

\*/

public Help() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

lblTitleHelp = new javax.swing.JLabel();

pnlHelp = new javax.swing.JScrollPane();

jTextArea1 = new javax.swing.JTextArea();

btnBack = new javax.swing.JButton();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

lblTitleHelp.setText("Help");

jTextArea1.setColumns(20);

jTextArea1.setRows(5);

jTextArea1.setText("This program is designed to help you calculate\nsome of the most common physics equations.\nFirstly, select the formula from the drop down\nyou wish to calculate. Then enter in the two\nnumbers in the corresponding text fields which\nwill be clearly labelled. After this click\nthe 'submit' button which will take you to a \nseperate screen with the answer. \nFurthermore, there is also a quiz in which you\nare given a formula and two numbers and you \nmust then work out the answer.");

pnlHelp.setViewportView(jTextArea1);

btnBack.setText("Back");

btnBack.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnBackActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(pnlHelp)

.addGroup(layout.createSequentialGroup()

.addComponent(lblTitleHelp)

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

.addGroup(layout.createSequentialGroup()

.addGap(159, 159, 159)

.addComponent(btnBack)

.addContainerGap(186, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addContainerGap()

.addComponent(lblTitleHelp)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

.addComponent(pnlHelp, javax.swing.GroupLayout.PREFERRED\_SIZE, 223, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(btnBack)

.addContainerGap(12, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

Formulae.changescreen(1);

//Here we are calling the method changescreen from the Formulae class

//The number in the brackets is which case is called

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Help.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Help.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Help.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Help.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Help().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnBack;

private javax.swing.JTextArea jTextArea1;

private javax.swing.JLabel lblTitleHelp;

private javax.swing.JScrollPane pnlHelp;

// End of variables declaration

}

*Results*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class Results extends javax.swing.JFrame {

/\*\*

\* Creates new form Results

\*/

public Results() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

lblResultsTitle = new javax.swing.JLabel();

lblFormula = new javax.swing.JLabel();

lblAnswer = new javax.swing.JLabel();

btnBack = new javax.swing.JButton();

txtFormula = new javax.swing.JTextField();

txtAnswer = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

lblResultsTitle.setText("Results");

lblFormula.setText("Formula");

lblAnswer.setText("Answer");

btnBack.setText("Back");

btnBack.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnBackActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(23, 23, 23)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblAnswer)

.addComponent(lblFormula)

.addComponent(txtFormula)

.addComponent(txtAnswer, javax.swing.GroupLayout.DEFAULT\_SIZE, 291, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(170, 170, 170)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(btnBack)

.addComponent(lblResultsTitle))))

.addContainerGap(86, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(lblResultsTitle)

.addGap(40, 40, 40)

.addComponent(lblFormula)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(txtFormula, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(36, 36, 36)

.addComponent(lblAnswer)

.addGap(18, 18, 18)

.addComponent(txtAnswer, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 28, Short.MAX\_VALUE)

.addComponent(btnBack)

.addGap(43, 43, 43))

);

pack();

}// </editor-fold>

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

Formulae.changescreen(1);

//Here we are calling the method changescreen from the Formulae class

//The number in the brackets is which case is called

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Results().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnBack;

private javax.swing.JLabel lblAnswer;

private javax.swing.JLabel lblFormula;

private javax.swing.JLabel lblResultsTitle;

private javax.swing.JTextField txtAnswer;

private javax.swing.JTextField txtFormula;

// End of variables declaration

}

*Quiz*

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package formulae;

/\*\*

\*

\* @author Chris

\*/

public class Results extends javax.swing.JFrame {

/\*\*

\* Creates new form Results

\*/

public Results() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

lblResultsTitle = new javax.swing.JLabel();

lblFormula = new javax.swing.JLabel();

lblAnswer = new javax.swing.JLabel();

btnBack = new javax.swing.JButton();

txtFormula = new javax.swing.JTextField();

txtAnswer = new javax.swing.JTextField();

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

lblResultsTitle.setText("Results");

lblFormula.setText("Formula");

lblAnswer.setText("Answer");

btnBack.setText("Back");

btnBack.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

btnBackActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(23, 23, 23)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(lblAnswer)

.addComponent(lblFormula)

.addComponent(txtFormula)

.addComponent(txtAnswer, javax.swing.GroupLayout.DEFAULT\_SIZE, 291, Short.MAX\_VALUE)))

.addGroup(layout.createSequentialGroup()

.addGap(170, 170, 170)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(btnBack)

.addComponent(lblResultsTitle))))

.addContainerGap(86, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(lblResultsTitle)

.addGap(40, 40, 40)

.addComponent(lblFormula)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addComponent(txtFormula, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(36, 36, 36)

.addComponent(lblAnswer)

.addGap(18, 18, 18)

.addComponent(txtAnswer, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 28, Short.MAX\_VALUE)

.addComponent(btnBack)

.addGap(43, 43, 43))

);

pack();

}// </editor-fold>

private void btnBackActionPerformed(java.awt.event.ActionEvent evt) {

Formulae.changescreen(1);

//Here we are calling the method changescreen from the Formulae class

//The number in the brackets is which case is called

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Results.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Results().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton btnBack;

private javax.swing.JLabel lblAnswer;

private javax.swing.JLabel lblFormula;

private javax.swing.JLabel lblResultsTitle;

private javax.swing.JTextField txtAnswer;

private javax.swing.JTextField txtFormula;

// End of variables declaration

}

# Test Log & Analysis

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | 001 | | | Test designed by | | Chris Bollard | |
| Test priority | 1 | | | Test date designed | | 15/15/2016 | |
| Test title | Loading the program | | | Test executed by | | Chris Bollard | |
| Test objective | Testing if the program loads correctly | | | Test executed date | | 15/15/2016 | |
| Step No. | Test Steps | Test Data | Expected Result | | Actual result | Status (pass/fail) | Notes and screenshot. |
| 1 | Running the program and looking if the GUI loads | N/A | The program should open to the calc GUI screen | | The program opened with the calc GUI screen showing | Pass | The GUI correctly opened. |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | 002 | | | Test designed by | | Chris Bollard | |
| Test priority | 2 | | | Test date designed | | 15/15/2016 | |
| Test title | Combo box options | | | Test executed by | | Chris Bollard | |
| Test objective | Making sure that the user can select each option | | | Test executed date | | 15/15/2016 | |
| Step No. | Test Steps | Test Data | Expected Result | | Actual result | Status (pass/fail) | Notes and screenshot. |
| 1 | Clicking the ‘current’ option | N/A | The user should be able to select ‘current’ | | The user can select ‘current’ from the list | Pass | The user can select ‘Current’ |
| 2 | Clicking the ‘forces’ option | N/A | The user should be able to select ‘forces’ | | The user can select ‘forces’ from the list | Pass | The user can select ‘forces’ |
| 3 | Clicking the ‘momentum’ option | N/A | The user should be able to select ‘momentum’ | | The user can select ‘momentum’ from the list | Pass | The user can select ‘momentum’ |
| 4 | Clicking the ‘distance’ option | N/A | The user should be able to select ‘distance’ | | The user can select ‘distance’ from the list | Pass | The user can select ‘distance’ |
| 5 | Clicking the ‘weight’ option | N/A | The user should be able to select ‘weight’ | | The user can select ‘weight’ from the list | Pass | The user can select ‘weight’ |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | 003 | | | Test designed by | | Chris Bollard | |
| Test priority | 2 | | | Test date designed | | 15/15/2016 | |
| Test title | Navigation of the program | | | Test executed by | | Chris Bollard | |
| Test objective | Testing if the buttons go to the correct screens | | | Test executed date | | 15/15/2016 | |
| Step No. | Test Steps | Test Data | Expected Result | | Actual result | Status (pass/fail) | Notes and screenshot. |
| 1 | Clicking the ‘help’ button on the main page | N/A | The program should go to the help screen | | The program went to the help screen | Pass |  |
| 2 | Clicking the ‘quiz’ button on the main page | N/A | The program should go to the quiz screen. | | The program went to the quiz screen. | Pass |  |
| 3 | Clicking the ‘back’ button on the help screen | N/A | The program should go to the main page. | | The program went to the main page. | Pass. |  |
| 4 | Clicking the ‘back’ button on the quiz page | N/A | The program should go to the main page | | The program went to the main page. | Pass. |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | 004 | | | Test designed by | | Chris Bollard | |
| Test priority | 2 | | | Test date designed | | 15/15/2016 | |
| Test title | Performing calculations | | | Test executed by | | Chris Bollard | |
| Test objective | Testing if the program calculates the formula correctly. | | | Test executed date | | 15/15/2016 | |
| Step No. | Test Steps | Test Data | Expected Result | | Actual result | Status (pass/fail) | Notes and screenshot. |
| 1 | Calculating the current | Voltage:10  Watts:5 | 10/5 = 2  This has been checked through a calculator and by asking a peer. | | The output was 2 | Pass |  |
| 2 | Calculating the force | Mass: 10  Acceleration: 5 | 10+10+10+10+10= 50  This has been checked through a calculator and by asking a peer. | | The output was 50 | Pass |  |
| 3 | Calculating the momentum | Mass:12  Velocity:6 | 12+12+12+12+12+12= 72  This has been checked through a calculator and by asking a peer. | | The output was 72 | Pass |  |
| 4 | Calculating the distance | Speed: 14  Time:7 | 14x7= 98 | | The output was 98 | Pass |  |
| 5 | Calculating the weight | Mass:16 Acceleration of gravity:8 | 16x2=32x4=128 | | The output was 128 | Pass |  |
| 6 | Calculating the current from the start | Voltage: 10  Watts: 20 | 10+10=20  There are two tens so answer equals 2 | | The program threw up an exception | Fail |  |

# Test Analysis (robust, reliable)

My first test involved testing whether the program loaded successfully, which it did. Also, because the program is relatively small in terms of file size it means that the program loaded up in a rather quick time of two seconds. One way in which this program could be improved is by repeating the test more times. This is because the program loading could have been an anomaly without me knowing.

The second group of tests I tested was to see whether the user could select each formula. Fortunately, each of the tests were successful. Furthermore, selecting each formula was very quick to do and it was easy to find out how to change the formula. One way in which I could improve these tests is to select the formulas in a random order to see if any errors occur.

My third group of tests were related to the navigation of the program. From these tests I gathered the knowledge that navigation through the program was efficient as only a maximum of two clicks were needed to get to anywhere in the program. I believe that these tests were very well thought out and therefore cannot see any way of improving the tests.

The final group of tests are related to performing calculations. As shown above when testing the calculations an error was thrown up. This error occurred when trying to calculate the default formula, which is ‘current’. Although this error did not crash the program an attempt to fix it must be made as it could prove an inconvenience among users. If I was to do this test again then I would input irregular values such as letters and symbols as a user could input them by mistake.

# Evaluation

The first functional requirement was for the user to be able to select a formula from a menu. I accomplished this by using a combo box in which the user could select a formula from the list.

The second functional requirement was for the user to be able to input the values required to perform the calculations. This was accomplished through the use of text boxes in which the user could type in the two values. Furthermore, the textboxes were labelled with which value needed to be entered into which box.

The third functional requirement was to output the answer on screen. Once the user enters in the two values and clicks calculate the program will validate the two values to make sure they are valid. After that the program will calculate the formula and then display the answer through a text box.

The final functional requirement was for the program to contain a separate help screen. This help screen is easily found through the use of a button labelled ‘Help’. The help screen contains information on how to use the program if the user is struggling.

One improvement that could have been implemented into my program is animations. I started to practice and learn how to create animations in other projects as I had the intent to include them into this project because it would make the program seem friendlier.

Another improvement that could have been made was to implement a quiz and results into the program. Again this would make the program friendlier and therefore would make physics more enjoyable to learn.

I believe that the program was very user friendly. Firstly, to be able to get to any part of the program a maximum of two clicks were required which is a very small amount. This means that the program was quick and efficient to use which in turn meant that the user would not get annoyed with the time it takes to get their answer.

As mentioned I included a help screen which guides the user through the program if they are unsure of how to use the program. Again this means that if there is a problem then it could be easily resolved which in turn means that the user will not get annoyed with the program and therefore put off by it.

One way in which the user interface could have been improved was to show an animation explaining each formula when the appropriate formula was selected. This is because the audience for the program is 16-18 year olds and therefore it needs to look visually appealing for them to take an interest in the application.

Unfortunately, I was not able to implement any features to assist accessibility due to time restraints. This means that users who have disabilities would find it hard to use or not be able to use it at all.

One way in which accessibility could be improved is through the use of a magnifying tool. The magnifying tool would enable partially sighted users to zoom in so that the text is larger and therefore easier to read.

Furthermore, if the user was fully blind then I could have implemented a speech input and output feature. This would work by the program outputting “select a formula” through the speakers to which the user would say a formula and so on.

Unfortunately the program was not very robust when an invalid input was entered. Although the program did not crash there was no explanation as to why no answer was displayed on screen. I established this by entering letters and symbols into each input box. If I was to update the program then I would include a message saying ‘only use numbers 0-9’ next to each input box. (Needs work)

I believe that the answers displayed were very accurate. I believe this because I cross checked the answer displayed with that of a calculator as well as asking a peer for the answer.

The first technique that I used was inheritance. Inheritance can be shown through the classes ‘FormulaeParent’ as well as the individual children. Within the parent class the program contains code which gets and sets the numbers inputted by the user into the textboxes. This is inherited by the children as they only contain code as to how the equation is formatted. This means that code does not need to be written out five times which in turn saves times and reduces the chances of errors.

The second technique that I used was encapsulation. Encapsulation involves setting methods to either public, private or protected depending on which methods I want to see each other. In my code I have set input 1 and 2 to protected. This is because protected allows children of the class to see it but no one else as the children need the numbers to work out the equation.

Polymorphism was another technique I used in my program. Polymorphism involves sending the same message to different child classes. An example of this in my code would be requesting a calculation and it being sent through a particular class such as ‘Current’ which would then calculate the current.

One technique which I could have used is try and catch. Try and catch method is associated with validation of the program. Effectively, if a correct input is made then the program will try something else and catch the original error. I could of implemented this in my program whereby if the user entered ‘five’ then it would try to convert it to integer this way the program wouldn’t crash.

Within the program I used comments throughout the program. These comments helped me understand what each line of code was doing and therefore if I have a problem with the program then it will be easier to locate the problem. Also, if I was to create another program then I could look back at this program for help or to reuse the code.

I also made use of indenting my code. Indenting my code allowed me to easily see where each method started and ended. Similarly to the comments this will allow me to easily locate a problem if there is one.

Yes the program was broken up into methods and classes with them being appropriately named as well. This made maintenance easy because if there was a problem with the program then I would be able to easily locate it through the name of the method or class. For example, if there was a problem with the help screen then it is most likely to be located in the ‘Help’ screen class.

One improvement that I could make to the program was to make it more visually appealing. As mentioned I could have implemented animations into the program whereby each time a formula selected an animation related to it will appear. Additionally, a wider use of colours could have been used to add a more vibrant feel to the program.

Another improvement that I could have made to my program was to add validity and verification. If the user would enter data which was not supported by the datatype double then the app would crash. One method of implementing this would be to have a note next to the textboxes stating only enter numbers 0-9. Also, using a try/catch method would enable the user to enter ‘five’ without any errors. This would improve my program because it would mean no crashes and therefore users do not get annoyed at the program.