MULTIMEDIA UNIVERSITY

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY

TOP2121 OBJECT ORIENTED PROGRAMMING

SESSION: TRIMESTER 1, 2021/2022

PROJECT TITLE SUBMISSION FORM

Project title:	Children Math Learning System		
Lecture/tutorial group:	OP1/1AV/1BV		
Lab tutor:	Md. Shohel Sayeed		
Submission date/time:	8-Nov-2021		
Project members:	**Name**	**ID**	**Major**
	1. Ahmed Aldughaither 2. Lujain Naser Ahmad Alaydie 3. Mayar Abdulmalik M Shenawi 4. Alsunnary Ahmed Jalal A	1. 1181102166 2. 1191302804 3. 1191302861 4. 1191302875	1.ST 2. AI 3.ST 4. AI

Introduction

In 2021, we saw many schools and universities use online learning to teach their students. So, we have decided to create a Children Math Learning System that help teach kids how to do simple math, learn about shapes, and lastly a tool to convert cm to m and vice versa. The student will have to compile Login.java file before starting the system. After that, our system will ask the user to login by keying in username and the password “student” then it will direct the student to the Main Menu, which has 4 features currently which are Simple Math, Shapes, and Converter and Exit button.

In simple math, the student will be able to practice how to add, subtract, multiply, divide and if the student has a problem solving the question, he can click the Hint button for help. If the student is done with practicing simple math, he can click the back button which will take him back to the Main Menu.

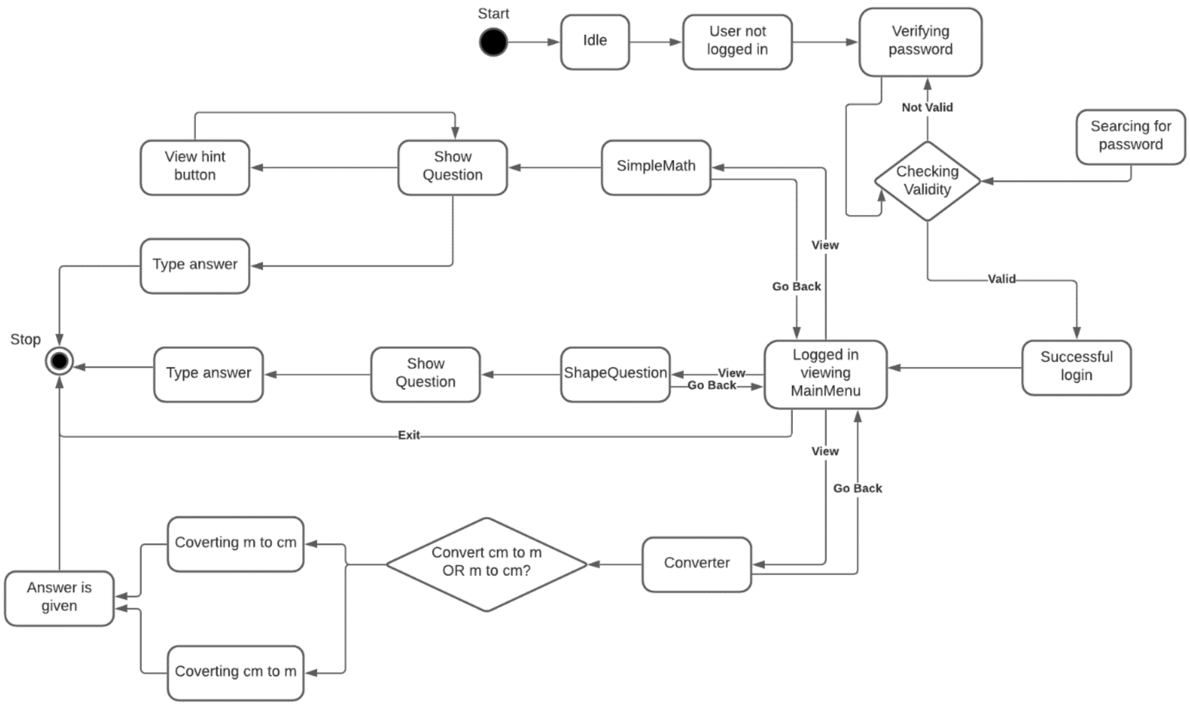
If the student choses to practice shapes, he can proceed to Shape questions which will showcase a rectangle with 2 randomized numbers of height and width for him to get the perimeter with the formula provided.

We also have a converter which student can choose whether he converts cm to m or m to cm and it will display the answer for him.

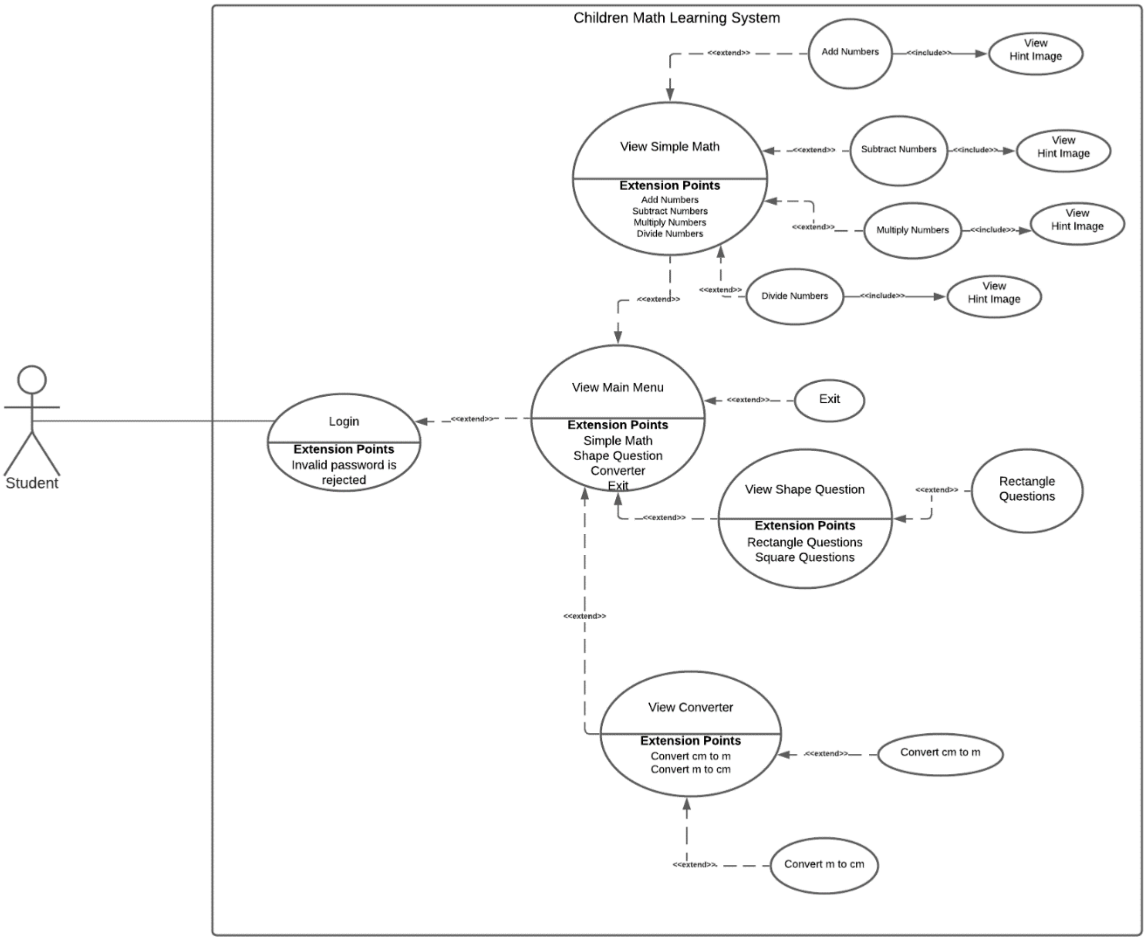
In our system we used inheritance in shape class via trial, polymorphism via simpleMath class, GUI is in every class, exception handling in simpleMath/shape/Converter if you input anything but numbers.

UML Diagrams

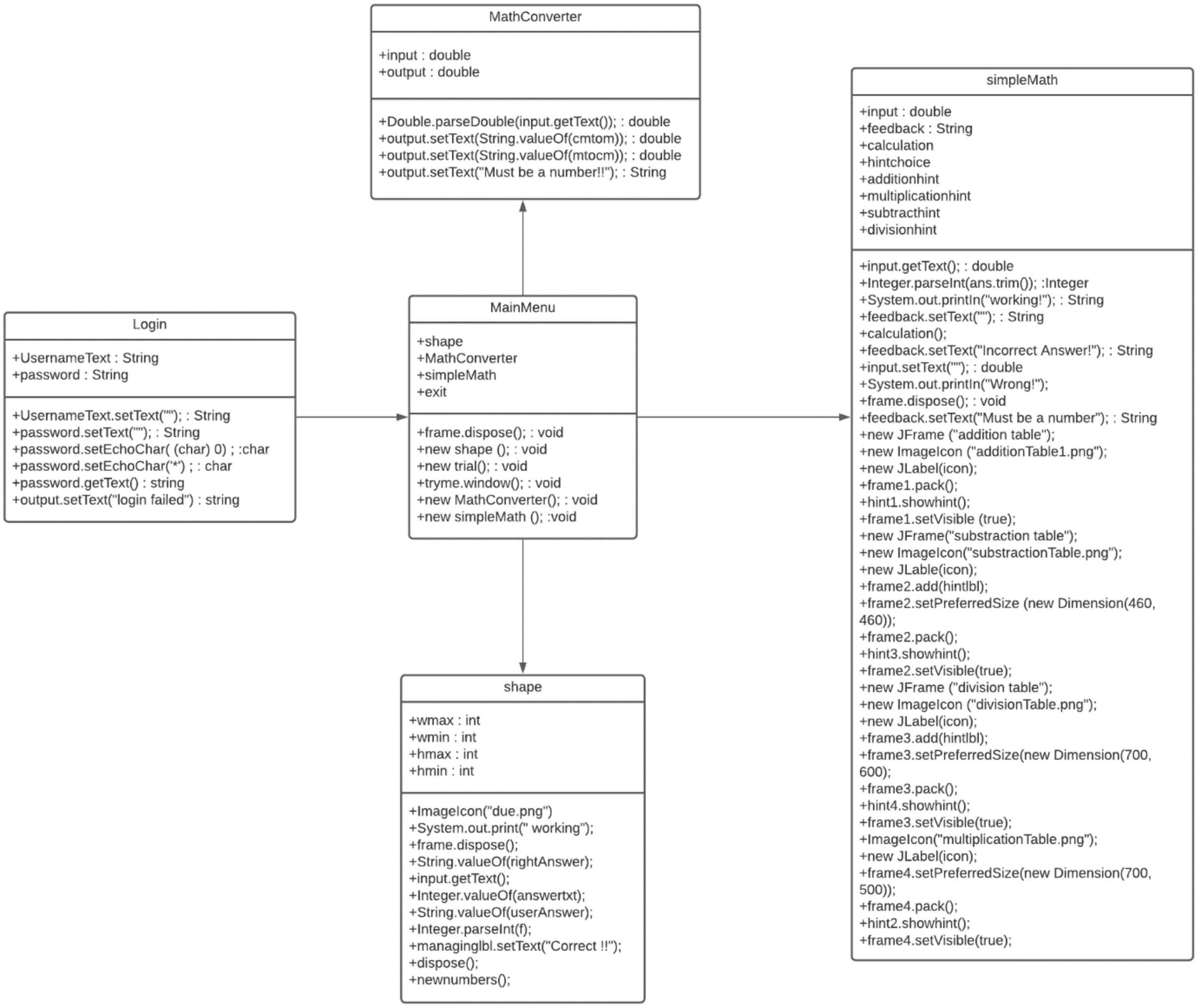
State Diagram



Case Diagram

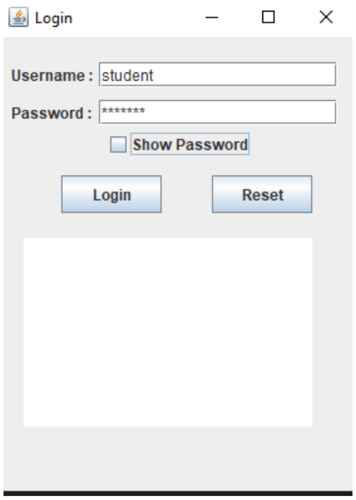


Class Diagram:


MathConverter
+input : double
+output : double
+Double.parseDouble(input.getText()); : double
+output.setText(String.valueOf(cmtom)); : double
+output.setText(String.valueOf(mtocm)); : double
+output.setText("Must be a number!!"); : String
simpleMath
+input : double
+feedback : String
+calculation
+hintchoice
+additionhint
+multiplicationhint
+subtracthint
+divisionhint
+input.getText(); : double
+Integer.parseInt(ans.trim()); :Integer
+System.out.println("working!"); : String
+feedback.setText(""); : String
+calculation();
+feedback.setText("Incorrect Answer!"); : String
+input.setText(""); : double
+System.out.println("Wrong!");
+frame.dispose(); : void
+feedback.setText("Must be a number"); : String
+new JFrame ("addition table");
+new ImageIcon ("additionTable1.png");
+new JLabel(icon);
+frame1.pack();
+hint1.showhint();
+frame1.setVisible (true);
+new JFrame("substraction table");
+new ImageIcon("substractionTable.png");
+new JLable(icon);
+frame2.add(hintlbl);
+frame2.setPreferredSize (new Dimension(460, 460));
+frame2.pack();
+hint3.showhint();
+frame2.setVisible(true);
+new JFrame ("division table");
+new ImageIcon ("divisionTable.png");
+new JLabel(icon);
+frame3.add(hintlbl);
+frame3.setPreferredSize(new Dimension(700, 600);
+frame3.pack();
+hint4.showhint();
+frame3.setVisible(true);
+ImageIcon("multiplicationTable.png");
+new JLabel(icon);
+frame4.setPreferredSize(new Dimension(700, 500));
+frame4.pack();
+hint2.showhint();
+frame4.setVisible(true);
Login
+UsernameText : String
+password : String
+UsernameText.setText(""); : String
+password.setText(""); : String
+password.setEchoChar((char) 0) ; :char
+password.setEchoChar('*') ; : char
+password.getText() : string
+output.setText("login failed") : string
MainMenu
+shape
+MathConverter
+simpleMath
+exit
+frame.dispose(); : void
+new shape (); : void
+new trial(); : void
+tryme.window(); : void
+new MathConverter(); : void
+new simpleMath (); :void
shape
+wmax : int
+wmin : int
+hmax : int
+hmin : int
+ImageIcon("due.png")
+System.out.print(" working");
+frame.dispose();
+String.valueOf(rightAnswer);
+input.getText();
+Integer.valueOf(answertxt);
+String.valueOf(userAnswer);
+Integer.parseInt(f);
+managinglbl.setText("Correct !!");
+dispose();
+newnumbers();

Screenshots of the System

Login:

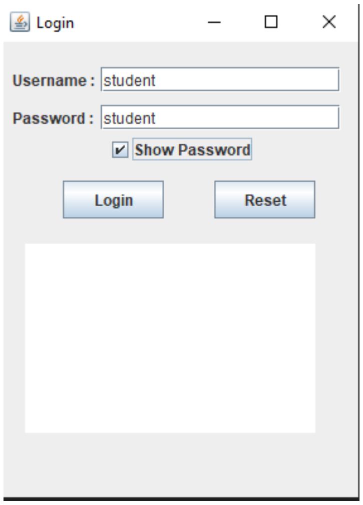


Username : student

Password : *******

□ Show Password

Login>Show Password

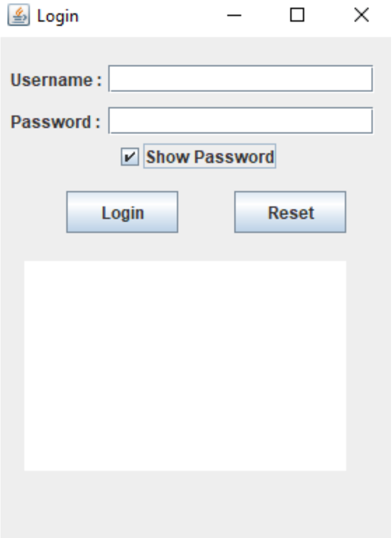


Username : student

Password : student

☑ Show Password

Login>Reset

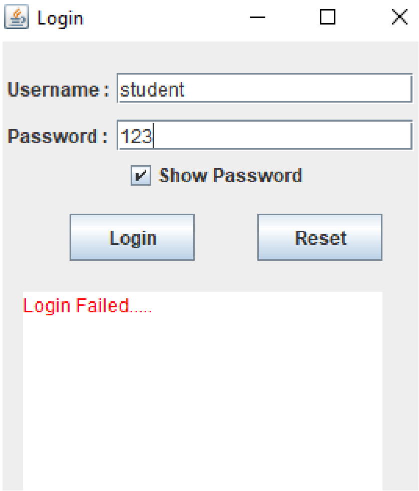


Username :

Password :

☑ Show Password

Login>Wrong password



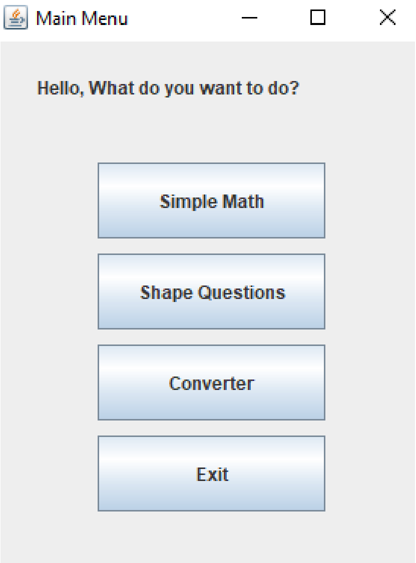
Username : student

Password : 123

☑ Show Password

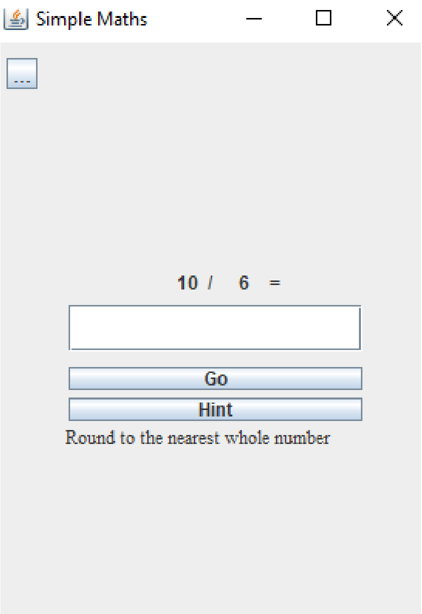
Login Failed.....

Main Menu



Hello, What do you want to do?

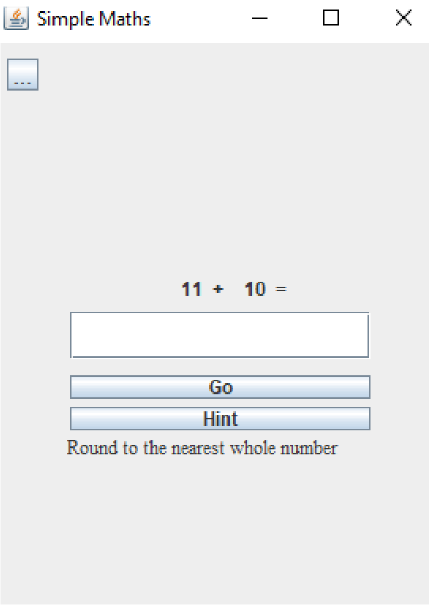
Simple Math>Division



10 /   6   =

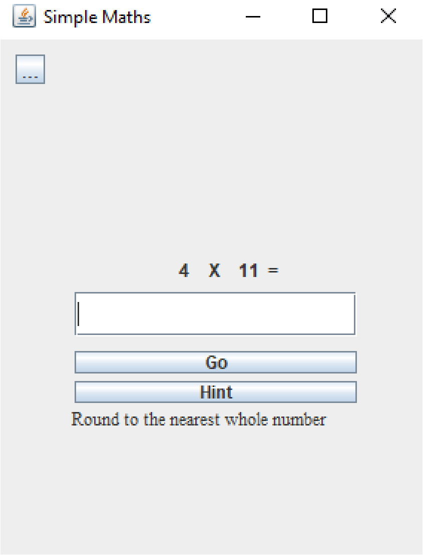
Round to the nearest whole number

Simple Math>Addition



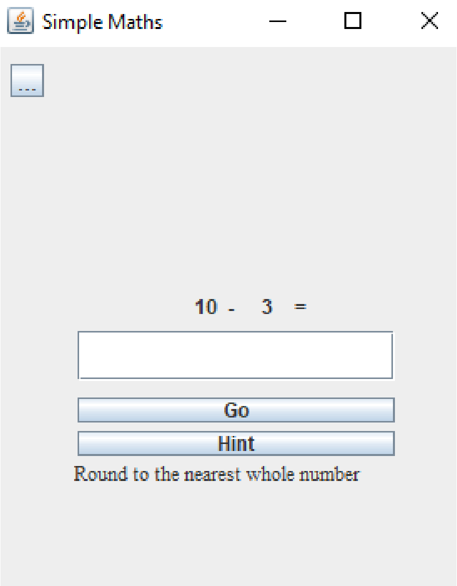
Round to the nearest whole number

Simple Maths>Multiplication



Round to the nearest whole number

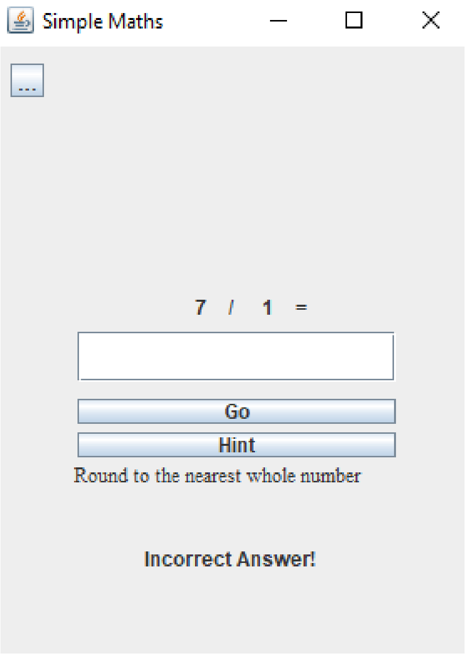
Simple Math>Subtraction



10 - 3 =

Round to the nearest whole number

Simple Math>Wrong answer

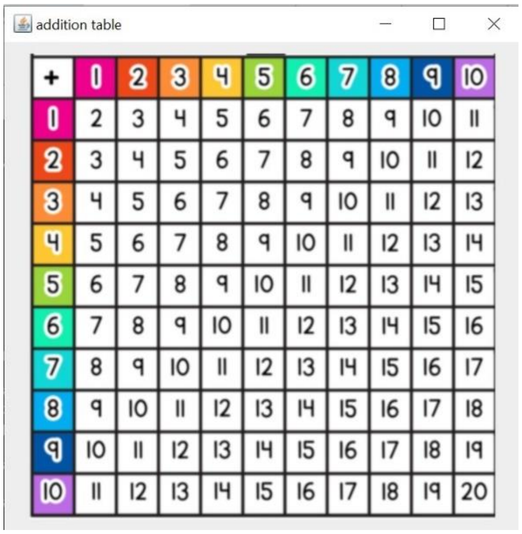


7 / 1 =

Round to the nearest whole number

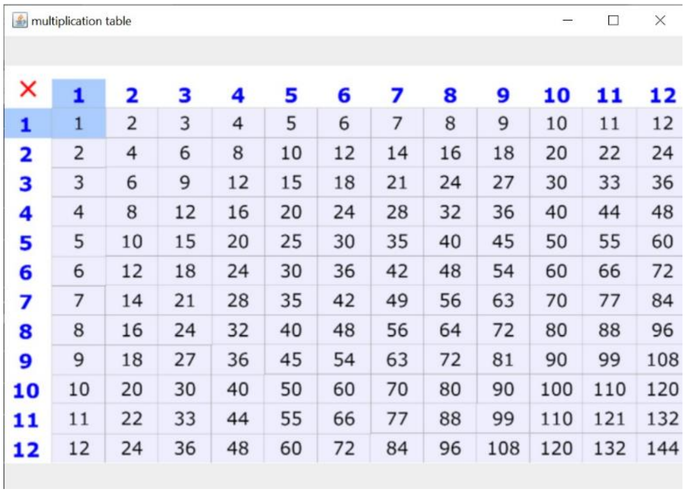
Incorrect Answer!

Simple Math>Addition>Hint



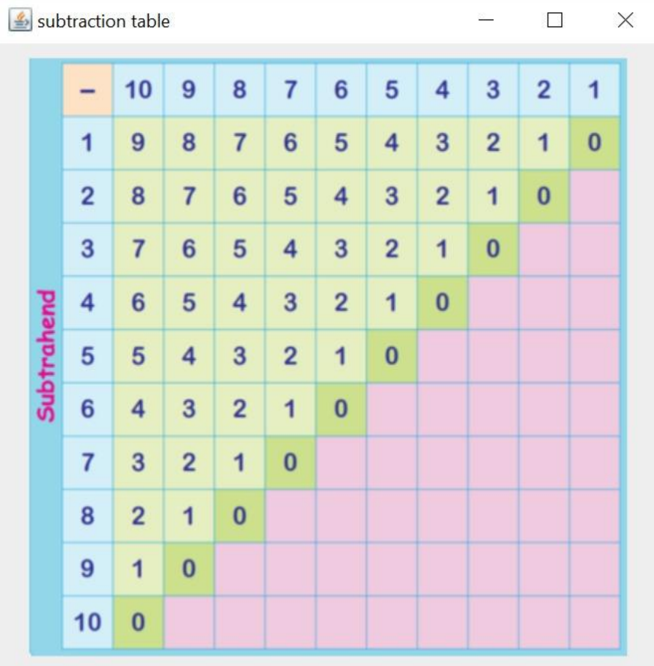
+	1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11
2	3	4	5	6	7	8	9	10	11	12
3	4	5	6	7	8	9	10	11	12	13
4	5	6	7	8	9	10	11	12	13	14
5	6	7	8	9	10	11	12	13	14	15
6	7	8	9	10	11	12	13	14	15	16
7	8	9	10	11	12	13	14	15	16	17
8	9	10	11	12	13	14	15	16	17	18
9	10	11	12	13	14	15	16	17	18	19
10	11	12	13	14	15	16	17	18	19	20

Simple Math>Multiplication>Hint



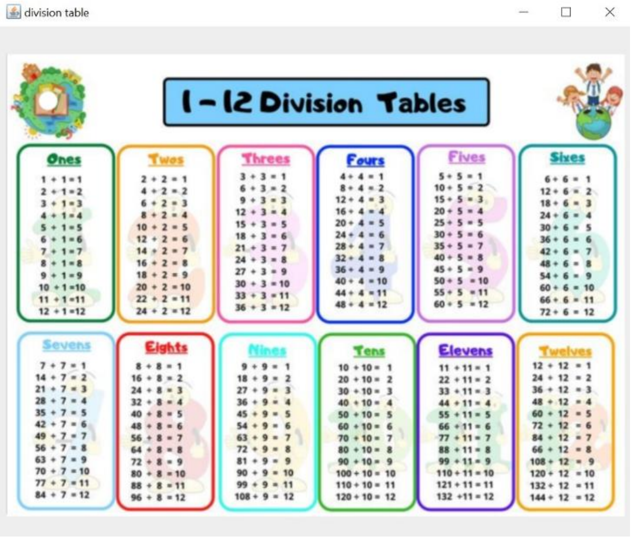
×	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	14	16	18	20	22	24
3	3	6	9	12	15	18	21	24	27	30	33	36
4	4	8	12	16	20	24	28	32	36	40	44	48
5	5	10	15	20	25	30	35	40	45	50	55	60
6	6	12	18	24	30	36	42	48	54	60	66	72
7	7	14	21	28	35	42	49	56	63	70	77	84
8	8	16	24	32	40	48	56	64	72	80	88	96
9	9	18	27	36	45	54	63	72	81	90	99	108
10	10	20	30	40	50	60	70	80	90	100	110	120
11	11	22	33	44	55	66	77	88	99	110	121	132
12	12	24	36	48	60	72	84	96	108	120	132	144

Simple Math>Subtraction>Hint



–	10	9	8	7	6	5	4	3	2	1
1	9	8	7	6	5	4	3	2	1	0
2	8	7	6	5	4	3	2	1	0	
3	7	6	5	4	3	2	1	0		
4	6	5	4	3	2	1	0			
5	5	4	3	2	1	0				
6	4	3	2	1	0					
7	3	2	1	0						
8	2	1	0							
9	1	0								
10	0									

Simple Math>Division>Hint



1 – 12 Division Tables

Ones

1 ÷ 1 = 1
2 ÷ 1 = 2
3 ÷ 1 = 3
4 ÷ 1 = 4
5 ÷ 1 = 5
6 ÷ 1 = 6
7 ÷ 1 = 7
8 ÷ 1 = 8
9 ÷ 1 = 9
10 ÷ 1 = 10
11 ÷ 1 = 11
12 ÷ 1 = 12

Twos

2 ÷ 2 = 1
4 ÷ 2 = 2
6 ÷ 2 = 3
8 ÷ 2 = 4
10 ÷ 2 = 5
12 ÷ 2 = 6
14 ÷ 2 = 7
16 ÷ 2 = 8
18 ÷ 2 = 9
20 ÷ 2 = 10
22 ÷ 2 = 11
24 ÷ 2 = 12

Threes

3 ÷ 3 = 1
6 ÷ 3 = 2
9 ÷ 3 = 3
12 ÷ 3 = 4
15 ÷ 3 = 5
18 ÷ 3 = 6
21 ÷ 3 = 7
24 ÷ 3 = 8
27 ÷ 3 = 9
30 ÷ 3 = 10
33 ÷ 3 = 11
36 ÷ 3 = 12

Fours

4 ÷ 4 = 1
8 ÷ 4 = 2
12 ÷ 4 = 3
16 ÷ 4 = 4
20 ÷ 4 = 5
24 ÷ 4 = 6
28 ÷ 4 = 7
32 ÷ 4 = 8
36 ÷ 4 = 9
40 ÷ 4 = 10
44 ÷ 4 = 11
48 ÷ 4 = 12

Fives

5 ÷ 5 = 1
10 ÷ 5 = 2
15 ÷ 5 = 3
20 ÷ 5 = 4
25 ÷ 5 = 5
30 ÷ 5 = 6
35 ÷ 5 = 7
40 ÷ 5 = 8
45 ÷ 5 = 9
50 ÷ 5 = 10
55 ÷ 5 = 11
60 ÷ 5 = 12

Sixes

6 ÷ 6 = 1
12 ÷ 6 = 2
18 ÷ 6 = 3
24 ÷ 6 = 4
30 ÷ 6 = 5
36 ÷ 6 = 6
42 ÷ 6 = 7
48 ÷ 6 = 8
54 ÷ 6 = 9
60 ÷ 6 = 10
66 ÷ 6 = 11
72 ÷ 6 = 12

Sevens

7 ÷ 7 = 1
14 ÷ 7 = 2
21 ÷ 7 = 3
28 ÷ 7 = 4
35 ÷ 7 = 5
42 ÷ 7 = 6
49 ÷ 7 = 7
56 ÷ 7 = 8
63 ÷ 7 = 9
70 ÷ 7 = 10
77 ÷ 7 = 11
84 ÷ 7 = 12

Eights

8 ÷ 8 = 1
16 ÷ 8 = 2
24 ÷ 8 = 3
32 ÷ 8 = 4
40 ÷ 8 = 5
48 ÷ 8 = 6
56 ÷ 8 = 7
64 ÷ 8 = 8
72 ÷ 8 = 9
80 ÷ 8 = 10
88 ÷ 8 = 11
96 ÷ 8 = 12

Nines

9 ÷ 9 = 1
18 ÷ 9 = 2
27 ÷ 9 = 3
36 ÷ 9 = 4
45 ÷ 9 = 5
54 ÷ 9 = 6
63 ÷ 9 = 7
72 ÷ 9 = 8
81 ÷ 9 = 9
90 ÷ 9 = 10
99 ÷ 9 = 11
108 ÷ 9 = 12

Tens

10 ÷ 10 = 1
20 ÷ 10 = 2
30 ÷ 10 = 3
40 ÷ 10 = 4
50 ÷ 10 = 5
60 ÷ 10 = 6
70 ÷ 10 = 7
80 ÷ 10 = 8
90 ÷ 10 = 9
100 ÷ 10 = 10
110 ÷ 10 = 11
120 ÷ 10 = 12

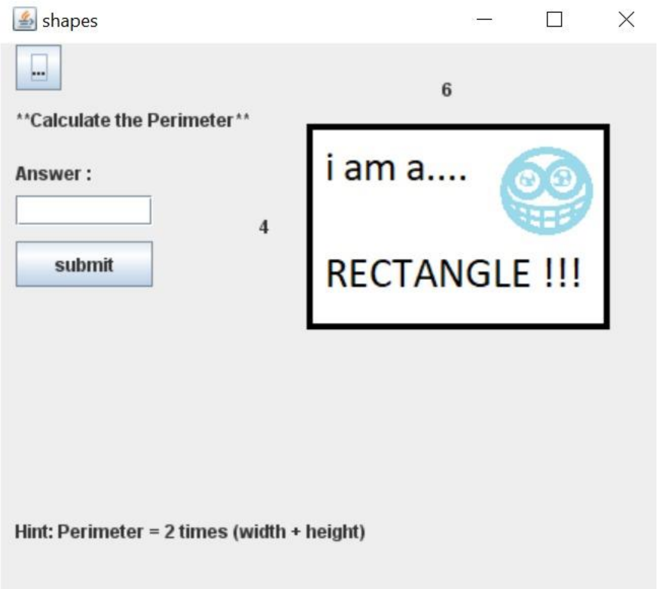
Elevens

11 ÷ 11 = 1
22 ÷ 11 = 2
33 ÷ 11 = 3
44 ÷ 11 = 4
55 ÷ 11 = 5
66 ÷ 11 = 6
77 ÷ 11 = 7
88 ÷ 11 = 8
99 ÷ 11 = 9
110 ÷ 11 = 10
121 ÷ 11 = 11
132 ÷ 11 = 12

Twelves

12 ÷ 12 = 1
24 ÷ 12 = 2
36 ÷ 12 = 3
48 ÷ 12 = 4
60 ÷ 12 = 5
72 ÷ 12 = 6
84 ÷ 12 = 7
66 ÷ 12 = 8
108 ÷ 12 = 9
120 ÷ 12 = 10
132 ÷ 12 = 11
144 ÷ 12 = 12

Shape Questions

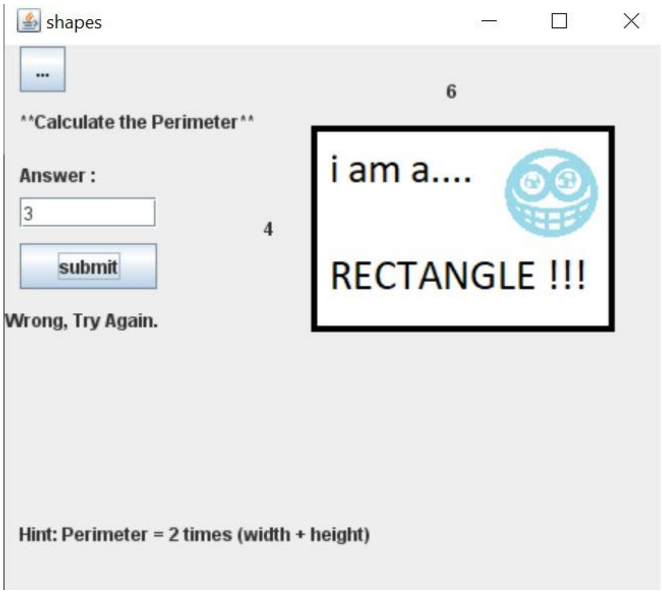


Calculate the Perimeter

Answer :

Hint: Perimeter = 2 times (width + height)

Shape Questions>Correct answer



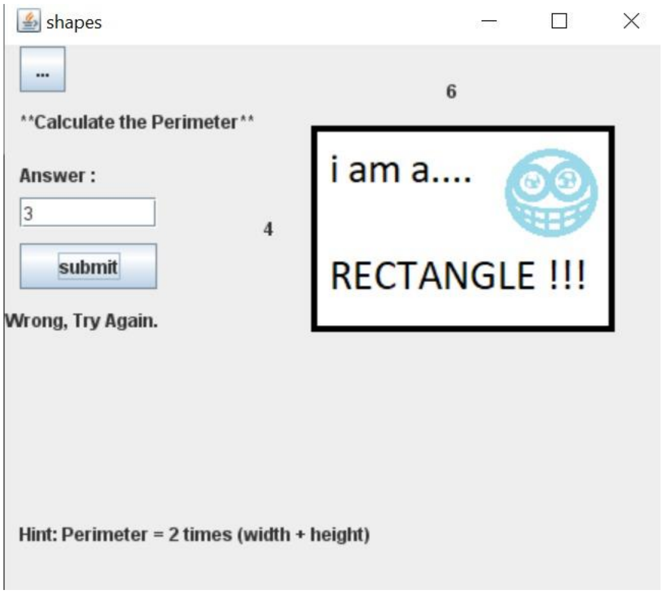
Calculate the Perimeter

Answer : 3

Wrong, Try Again.

Hint: Perimeter = 2 times (width + height)

Shape>Wrong answer



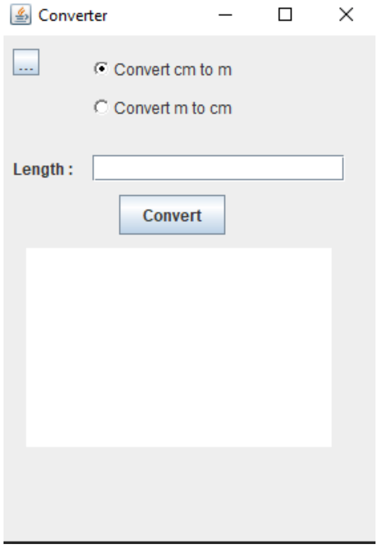
Calculate the Perimeter

Answer : 3

Wrong, Try Again.

Hint: Perimeter = 2 times (width + height)

Converter

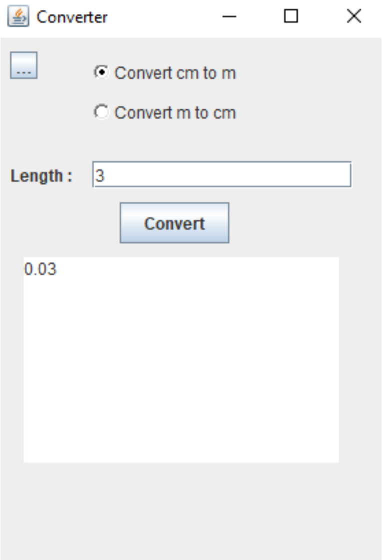


● Convert cm to m

○ Convert m to cm

Length :

Converter>cm to m



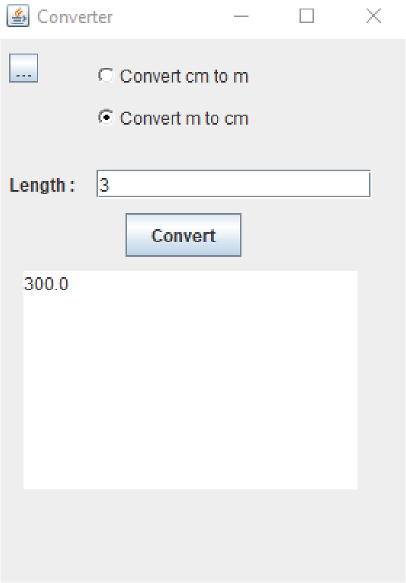
● Convert cm to m

○ Convert m to cm

Length : 3

```
0.03
```

Converter>m to cm



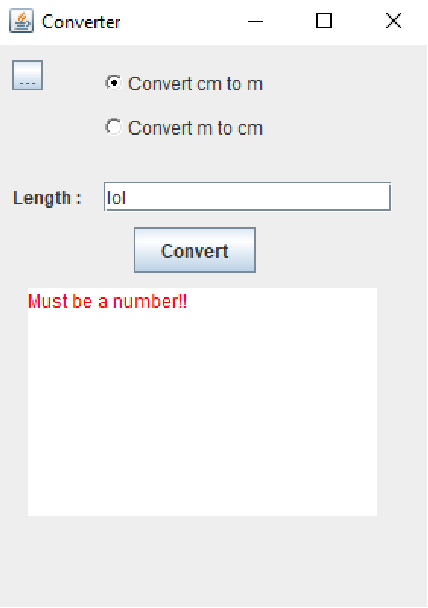
○ Convert cm to m

● Convert m to cm

Length : 3

```
300.0
```

Converter>Wrong Format



lol

Must be a number!!

Source Code

Login Source Code:

```
import java.awt.Color;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.Scanner;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPasswordField;
import javax.swing.JTextArea;
import javax.swing.JTextField;


public class Login extends JFrame implements ActionListener
      {
   static Scanner sc = new Scanner(System.in);
   static Scanner scStr = new Scanner(System.in);
   JFrame frame = new JFrame("Login");
   JLabel label = new JLabel("Username : "); //student username
   JTextField UsernameText = new JTextField();      //input student username
   JLabel label2 = new JLabel("Password : "); //student password
   JPasswordField password = new JPasswordField();     //input student password
   JButton Loginbutton = new JButton("Login");      //button to login
   JButton Resetbutton = new JButton("Reset"); // reset button
   JTextArea output = new JTextArea(10,15);    //login succesful or failed
   JCheckBox showPassword=new JCheckBox("Show Password");    //Password checkbox
   String username;
```

```
      public static void main(String[] args){

      new Login();

   }
public Login(){
       label.setBounds(10,-20, 100,100);
       UsernameText.setBounds(80,20,190,20);
       label2.setBounds(10,50,190,20);
       password.setBounds(80,50,190,20);
       Loginbutton.setBounds(50,110,80,30);
       Resetbutton.setBounds(170,110,80,30);
       output.setBounds(20,160,230,150);
       showPassword.setBounds(85,70,150,30);



       frame.add(label);
       frame.add(label2);
       frame.add(UsernameText);
       frame.add(password);
       frame.add(Loginbutton);
       frame.add(output);
       frame.add(showPassword);
       frame.add(Resetbutton);

      Loginbutton.addActionListener(this); //login button
       Loginbutton.setFocusable(false);

       Resetbutton.addActionListener(this); //reset button
```

```
    Resetbutton.setFocusable(false);

    showPassword.addActionListener(this); //check password box
    frame.setSize(300,400);
    frame.setLayout(null);
    frame.setVisible(true);
    frame.setDefaultCloseOperation(EXIT_ON_CLOSE);

}

@Override
public void actionPerformed(ActionEvent evt){
    if (evt.getActionCommand().equalsIgnoreCase("Reset")) {
      UsernameText.setText("");
      password.setText("");
    }
    if (evt.getSource() == showPassword) {
      if (showPassword.isSelected()) {
        password.setEchoChar((char) 0);
      } else {
        password.setEchoChar('*');
      }
    }
    if(evt.getActionCommand().equalsIgnoreCase("Login")){
      if(password.getText().equalsIgnoreCase("student")) {
        frame.dispose();
        MainMenu mainMenu = new MainMenu();
      }
    else{
      output.setForeground(Color.red);
```

```
            output.setText("Login Failed.....") ;
            }
        }
    }
}
```

Main Menu Source Code

```
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;



public class MainMenu extends shape implements ActionListener{
   JFrame frame = new JFrame("Main Menu");
   JLabel label = new JLabel("Hi"); //student username
   JButton Mathbutton = new JButton("Simple Math");      // button to go to simple math
   JButton Shapesbutton = new JButton("Shape Questions");      //button to go to shapes questions
   JButton Converterbutton = new JButton("Converter");      //button to go to converter
   JButton ExitButton = new JButton ("Exit");            //exit button




  public MainMenu(){

   label.setBounds(30,-20,200,100);
   Mathbutton.setBounds(70,80,150,50);
   Shapesbutton.setBounds(70,140,150,50);
   Converterbutton.setBounds(70,200,150,50);
   ExitButton.setBounds(70,260,150,50);
```

```
label.setText("Hello, What do you want to do? ");

frame.add(label);
frame.add(Mathbutton);
Mathbutton.setFocusable(false);
frame.add(Shapesbutton);
Shapesbutton.setFocusable(false);
frame.add(Converterbutton);
Converterbutton.setFocusable(false);
frame.add(ExitButton);
ExitButton.setFocusable(false);


Mathbutton.addActionListener(this);
Shapesbutton.addActionListener(this);
Converterbutton.addActionListener(this);
ExitButton.addActionListener(this);
frame.setSize(300,400);
frame.setLayout(null);
frame.setVisible(true);
frame.setDefaultCloseOperation(EXIT_ON_CLOSE);

}



@Override
public void actionPerformed(ActionEvent evt) {
    if (evt.getActionCommand().equalsIgnoreCase("Shape questions")) {
                                frame.dispose();
```

```
          shape shapepage = new shape();


                           trial tryme = new trial();


                           tryme.window();
                                             if(close == true){


                   System.out.print("works");
          }



      }
      if (evt.getActionCommand().equalsIgnoreCase("Converter")){
         frame.dispose();
         MathConverter mathConveter = new MathConverter();


      }
      if (evt.getActionCommand().equalsIgnoreCase("Simple Math")){
         frame.dispose();
         simpleMath maths = new simpleMath();
      }
      if (evt.getActionCommand().equalsIgnoreCase("Exit")){
         frame.dispose();
      }
      }
     public static void main(String[] args){
   }
}
```

simpleMath Source Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.imageio.*;
import java.util.Random;
import java.util.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

 abstract class hintchoice{
        abstract void showhint();
}
 class additionhint extends hintchoice{
        void showhint(){
                System.out.print("addition hint created");
        }
}
class multiplicationhint extends hintchoice{
        void showhint(){
                System.out.print("multiplication hint created");
        }
}
class subtractionhint extends hintchoice{
                void showhint(){
                System.out.print("subtraction hint created");
        }
}
class divisionhint extends hintchoice{
                void showhint(){
                System.out.print("division hint created");
        }
}

public class simpleMath extends JFrame implements ActionListener {
        //hints
        additionhint hint1 = new additionhint();
        multiplicationhint hint3 = new multiplicationhint();
        subtractionhint hint2 = new subtractionhint();
        divisionhint hint4 = new divisionhint();
        JLabel hintlbl = new JLabel();

        //
    static int randomNum1;
    static int randomNum2;
    static int operation;
    static int answer;
    static Random randomNums = new Random();
    int min = 1;
    int max = 10;
    JFrame frame = new JFrame("Simple Maths");
    JTextField input = new JTextField(10);
    JButton button = new JButton("Go");
    JButton Hintbutton = new JButton("Hint");
    JLabel feedback = new JLabel();
    JLabel first = new JLabel();
    JLabel second = new JLabel();
    JLabel op = new JLabel();
    JLabel equals = new JLabel("=");
    JButton Backbutton = new JButton("Back");
    JLabel labelHint = new JLabel("Round to the nearest whole number");
```

```
public void calculation(){
  randomNum1 = randomNums.nextInt(max + min) + 1;
  randomNum2 = randomNums.nextInt(max + min) + 1;
  operation = randomNums.nextInt(3 + 1) + 1;
  System.out.println(operation);

  input.setText("");
  if(operation == 1){
    op.setText("+");
    answer = randomNum1 + randomNum2;
  }
  if(operation == 2){
    op.setText("-");
    if(randomNum1 < randomNum2){
      int temp;
      temp = randomNum1;
      randomNum1 = randomNum2;
      randomNum2 = temp;
    }
    answer = randomNum1 - randomNum2;


  }
  if(operation == 3){
    op.setText("/");
    if(randomNum1 < randomNum2){
      int temp;
      temp = randomNum1;
      randomNum1 = randomNum2;
      randomNum2 = temp;
      answer = randomNum1 / randomNum2;
    }

  if(randomNum2 != 0)
      answer = randomNum1 / randomNum2;
  }
  if(operation == 4){
    op.setText("X");
    answer = randomNum1 * randomNum2;
  }

  first.setText(Integer.toString(randomNum1));
  second.setText(Integer.toString(randomNum2));

}

public void swapValues(int a, int b){


}
public simpleMath(){
  frame.setLayout(null);
  frame.setSize(300,420);
  frame.add(labelHint);
  first.setBounds(120,150,190,10);
  second.setBounds(160,150,190,10);
  op.setBounds(140,150,190,10);
  equals.setBounds(180,150,190,10);
  input.setBounds(50,170,190,30);
  button.setBounds(50, 210, 190,15);
  Hintbutton.setBounds(50, 230, 190, 15);
```

```
labelHint.setBounds(48,250,190,10);
labelHint.setFont(new Font("TimesRoman", Font.PLAIN,12));
feedback.setBounds(90,230,190,10);
Backbutton.setBounds(10,10,20,20);
Backbutton.setFont(new Font("TimesRoman", Font.PLAIN,15));

frame.add(equals);
frame.add(input);
frame.add(button);
frame.add(Hintbutton);
frame.add(first);
frame.add(second);
frame.add(op);
frame.add(feedback);
frame.add(Backbutton);

    //
      frame.setSize(300,420);
first.setBounds(120,150,190,10);
second.setBounds(160,150,190,10);
op.setBounds(140,150,190,10);
equals.setBounds(180,150,190,10);
input.setBounds(50,170,190,30);
button.setBounds(50, 210, 190,15);
Hintbutton.setBounds(50, 230, 190, 15);
feedback.setBounds(90,300,190,10);
Backbutton.setBounds(10,10,20,20);
Backbutton.setFont(new Font("TimesRoman", Font.PLAIN,15));
    //
   //frame.add(hintlbl); // hint label
   Backbutton.setFocusable(false);
   Backbutton.addActionListener(this);

   Hintbutton.addActionListener(this);
   Hintbutton.setFocusable(false);
   button.setFocusable(false);
   button.addActionListener(this);
   frame.setVisible(true);
   frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
   calculation();

 }

 @Override
public void actionPerformed(ActionEvent e1){
     if (e1.getActionCommand().equals("Go")){
       try{
         String ans = input.getText();
         int userAns = Integer.parseInt(ans.trim());
         if(answer == userAns){
           System.out.println("working!");
           feedback.setText("");

           calculation();

         }else{
           feedback.setText("Incorrect Answer!");
           input.setText("");
           System.out.println("Wrong!");
         }
```

```
  }catch (NumberFormatException ex) {
    feedback.setText("Must be a number");
  }
}
if (e1.getActionCommand().equals("Back")){
        frame.dispose();
        MainMenu mainMenu = new MainMenu();
}
if(e1.getActionCommand().equals("Hint")){// hint button action
                if(operation == 1){// addition hint
                JFrame frame1 = new JFrame("addition table");
                ImageIcon icon = new ImageIcon("additionTable1.png");
                            hintlbl = new JLabel(icon);
                frame1.add(hintlbl);
                frame1.setPreferredSize(new Dimension(460, 460));
                            frame1.pack();

                            hint1.showhint();
                            frame1.setVisible(true);
                }
                if(operation == 2){// subtraction hint
                JFrame frame2 = new JFrame("subtraction table");
                 ImageIcon icon = new ImageIcon("subtractionTable.png");
                            hintlbl = new JLabel(icon);
                frame2.add(hintlbl);
                frame2.setPreferredSize(new Dimension(460, 460));
                            frame2.pack();

                            hint3.showhint();
                            frame2.setVisible(true);
        }

        if(operation == 3){ // division hint
            JFrame frame3 = new JFrame("division table");
             ImageIcon icon = new ImageIcon("divisionTable.png");
            hintlbl = new JLabel(icon);
            frame3.add(hintlbl);
            frame3.setPreferredSize(new Dimension(700, 600));
            frame3.pack();



            hint4.showhint();
            frame3.setVisible(true);

        }
            if(operation == 4){// multiplication hint
            JFrame frame4 = new JFrame("multiplication table");
            ImageIcon icon = new ImageIcon("multiplicationTable.png");
            hintlbl = new JLabel(icon);
            frame4.add(hintlbl);
            frame4.setPreferredSize(new Dimension(700, 500));
            frame4.pack();

            hint2.showhint();
            frame4.setVisible(true);
        }



}
}
```

```
  public static void main(String[] args){
    new simpleMath();
 }
}
```

shape source code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.imageio.*;
import java.util.Random;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;
public class shape extends JFrame implements ActionListener {

        JFrame frame = new JFrame("shapes");



JLabel label = new JLabel("Answer : ");
JTextField input = new JTextField(10);
JButton button = new JButton("submit");
JTextArea output = new JTextArea(10,15);
int wmax = 10;
int wmin = 6;
int hmax = 5;
int hmin = 1;
 int widthNumber = new Random().nextInt(wmax - wmin + 1) + wmin;
 int heightNumber = new Random().nextInt(hmax - hmin + 1) + hmin;

JLabel shapeWidth =  new JLabel();
JLabel shapeHeight =  new JLabel();
JLabel managinglbl = new JLabel(); // correct/wrong feedback.
JLabel hint = new JLabel("Hint: Perimeter = 2 times (width + height)");
JLabel question = new JLabel("**Calculate the Perimeter**");
JLabel exH = new JLabel("                                        ");
JButton mainmenub = new JButton("Back");
```

```
//
boolean close = false;
//circumference right answer
int rightAnswer = 2 * (widthNumber + heightNumber);

//JLabel shapeHeight = new JLabel(heightNumber);
public  void newnumbers(){
        widthNumber = new Random().nextInt(wmax - wmin + 1) + wmin;
        heightNumber = new Random().nextInt(hmax - hmin + 1) + hmin;

        rightAnswer = 2 * (widthNumber + heightNumber);

        shapeWidth.setText(String.valueOf(widthNumber));
        shapeHeight.setText(String.valueOf(heightNumber));
}


 //image

  ImageIcon icon = new ImageIcon("due.png");
  JLabel label20 = new JLabel(icon);




  public static void main(String[] args){
   /*
   trial tryme = new trial();

        tryme.window();
        */
```

```
}



        public /*static*/ class trial extends shape{


        public void window()
        {

        //frame



  frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);


  frame.setPreferredSize(new Dimension(450, 400));
  frame.pack();



        //Layout
        managinglbl.setBounds(10,300,20,20);


        shapeWidth.setText(String.valueOf(widthNumber));
        shapeHeight.setText(String.valueOf(heightNumber));
                shapeWidth.setBounds(290,20,20,20);        // immovable
                shapeHeight.setBounds(170,110,20,20); // right number status: movable


                label.setBounds(10,70,90,30);
                input.setBounds(10,100,90,20);
                button.setBounds(10,130,90,30);
                hint.setBounds(10,310,300,20);
                label20.setBounds(200,50,200,140);
```

```
    question.setBounds(10,40,250,20);
                exH.setBounds(10,290,300,20);
                mainmenub.setBounds(10,1,30,30);


                frame.add(shapeHeight);
                frame.add(shapeWidth);
frame.add(button);
                frame.add(input);
frame.add(label);
                frame.add(hint);
                frame.add(question);
                 frame.add(label20);
                 frame.add(mainmenub);
                 frame.add(exH);

                frame.add(managinglbl);


button.addActionListener(this);
                mainmenub.addActionListener(this);
     frame.setVisible(true);



                }

    }

    // get number from textfield store it in a variable/ convert, then output
    //Action Listener

 public void actionPerformed(ActionEvent e) {

        if(e.getActionCommand().equals("Back")){

                    MainMenu mainMenu = new MainMenu();
                    close = true;
                    System.out.print(" working");
                    frame.dispose();

                }
        //change this to fit shape class
        String rightAnswertxt = String.valueOf(rightAnswer); // right answer
        String answertxt = input.getText(); // user input
            try{  //handle exception
                        int userAnswer = Integer.valueOf(answertxt);
                String f = String.valueOf(userAnswer);
                int ff = Integer.parseInt(f);
                }catch (NumberFormatException ex) {

                    exH.setText("##invalid input: please answer with a
number##");

                    }



                if(answertxt.equals(rightAnswertxt)){
                    managinglbl.setText("Correct !!");
                    //change numbers if correct
```

```
                           dispose();
                                     newnumbers();
                               //



                         }


                         else{
                               managinglbl.setText("Wrong, Try Again.");
                         }






  }


}
```

MathConverter Source Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import static javax.swing.WindowConstants.EXIT_ON_CLOSE;

public class MathConverter extends JFrame implements ActionListener {
    JFrame frame = new JFrame("Converter");
    JLabel label = new JLabel("Length : ");
    JTextField input = new JTextField(10);
    JButton button = new JButton("Convert");
    JButton Backbutton = new JButton("Back");
    JTextArea output = new JTextArea(10,15);
    CheckboxGroup cbg = new CheckboxGroup();
    Checkbox cb1 = new Checkbox("Convert cm to m", cbg, true);
    Checkbox cb2 = new Checkbox("Convert m to cm", cbg, false);



public MathConverter(){
    cb1.setBounds(70,20,190,10);
    cb2.setBounds(70,30,190,50);
    label.setBounds(10,50, 100,100);
    input.setBounds(70,90,190,20);
    button.setBounds(90,120,80,30);
    Backbutton.setBounds(10,10,20,20);
    output.setBounds(20,160,230,150);
    Backbutton.setFont(new Font("TimesRoman", Font.PLAIN,15));

        frame.add(cb1);
```

```
     cb1.setFocusable(false);
     frame.add(cb2);
     cb2.setFocusable(false);
     frame.add(label);
     frame.add(input);
     frame.add(button);
     button.setFocusable(false);
     frame.add(Backbutton);
     Backbutton.setFocusable(false);
     frame.add(output);

 button.addActionListener(this);
 Backbutton.addActionListener(this);
 frame.setSize(300,420);
 frame.setLayout(null);
 frame.setVisible(true);
 frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
 }

 @Override
public void actionPerformed(ActionEvent e1){
     if (e1.getActionCommand().equals("Convert")){
         try{
         if(cb1.getState()){
         double l = Double.parseDouble(input.getText());
         double cmtom = 0;
         cmtom = l / 100;
         output.setText(String.valueOf(cmtom));
         }
             else if(cb2.getState()){
```

```
        double l = Double.parseDouble(input.getText());
        double mtocm = 0;
         mtocm = l * 100;
        output.setText(String.valueOf(mtocm));
        }
        }catch(NumberFormatException ex) {
        output.setForeground(Color.red);
        output.setText("Must be a number!!");
      }
        }
    if (e1.getActionCommand().equals("Back")){
            frame.dispose();
            MainMenu mainMenu = new MainMenu();
        }
 }
public static void main(String[] args){
    new MathConverter();
    }
}
```