**Name: Hamad Nasir**

**Roll No: 120312**

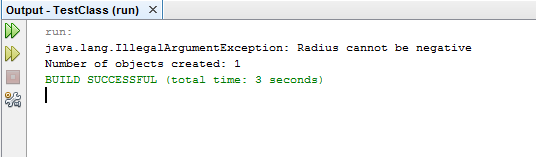
**Section: BSCS-6C**

**Lab No: 9**

**Activity No 1:**

**Q1.What will be the output?**

**Ans.** The output is as follows

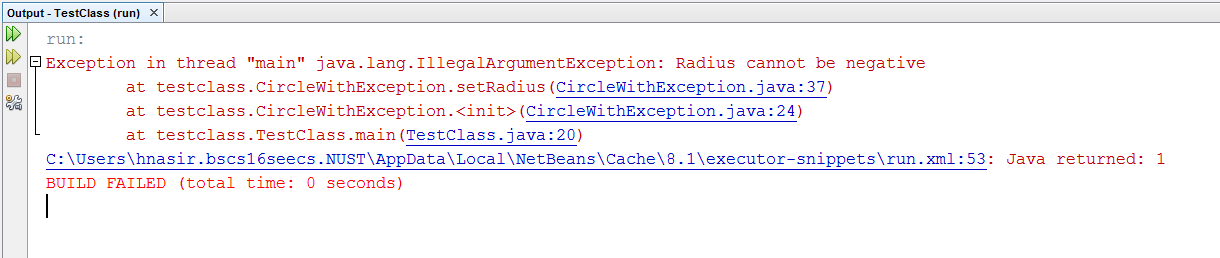
****

**Q2. What happens if we remove the clause throws IllegalArgumentException from the setRadius method declaration, and re-compile the CircleWithException class? Would it compile? If so, why?**

**Ans.** When IllegalArgumentException clause from the setRadius method was removed then it does not show any error and the program was compiled because doing this we tell the program that this method does not show any exception and it was skip by the compiler exception check. But know whenever there is negative value of radius in setRadius method the program will not compile and it will show exception.

**Q3. What happens if we do not handle the IllegalArgumentException in the TestCircleWithException class by not using the try statement?**

**Ans.** If we do not handle the IllegalArgumentException in the TestCircleWithException class by not using the try statement then the compiler will show exception because know the negative value -5 will be assigned to setRadiusMethod which will be a exception case because setRadiusMethod throws exception and it is not handled.

****

**Activity No:2**

**Q1. If no exception occurs, will statement4 be executed, and will statement5 be executed?**

**Ans.** If no exception occurs then catch block will be skipped and control will pass to finally block thus executing the statement 4. After this control will pass to code just after finally block thus executing statement 5. In this way both the statements will be executed.

**Q2. If the exception is of type Exception1, will statement4 and statement5 be executed?**

**Ans.** If the exception is of type Exception1 then as exception is not handled in catch block the control will pass to finally block but it will never pass to the code after the finally block and it will return to try block. Thus in this case statement 4 will be executed but statement 5 will not be executed.

**Q3.** **If the exception is of type Exception2, will statement4 and statement5 be executed?**

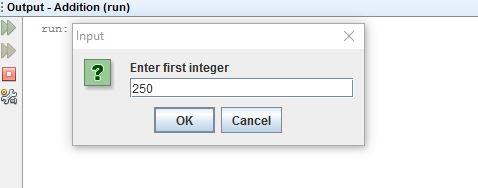
**Ans.** If the exception is of type Exception2 then in this case as the catch block of Exception 2 has keyword throws it means it can’t handle the Exception 2 and it is throwing it outside the catch block. Thus in this case statement 4 will be executed in the finally block but as the exception is rethrown so statement 5 will not be executed.

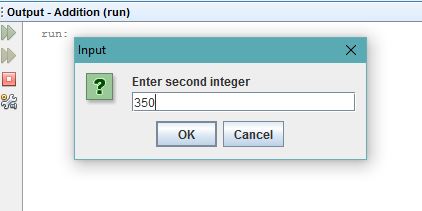
**Q4. If the exception is not Exception1 nor Exception2, will statement4 and statement5 be executed?**

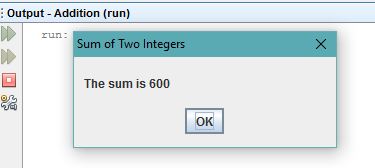
**Ans.** If the exception is not Exception1 nor Exception2 then catch block will be skipped and control will pass to finally block thus executing the statement 4. After this control will pass to code just after finally block thus executing statement 5. In this way both the statements will be executed.

**Activity No 3:**

The output of original code is:







**Source code for Three integers:**

package addition;

/\*\*

\*

\* @author Hamad nasir

\*/

import javax.swing.JOptionPane; // program uses JOptionPane

public class Addition

{

public static void main( String[] args )

{

// obtain user input from JOptionPane input dialogs

String firstNumber =

JOptionPane.showInputDialog( "Enter first integer" );

String secondNumber =

JOptionPane.showInputDialog( "Enter second integer" );

String thirdNumber =

JOptionPane.showInputDialog( "Enter third integer" );

// convert String inputs to int values for use in a

// calculation

int number1 = Integer.parseInt( firstNumber );

int number2 = Integer.parseInt( secondNumber );

int number3 = Integer.parseInt( thirdNumber );

int sum = number1 + number2+ number3; // add numbers

// display result in a JOptionPane message dialog

JOptionPane.showMessageDialog( null, "The sum is " + sum,

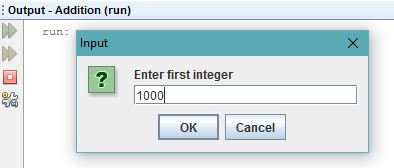
"Sum of Three Integers",

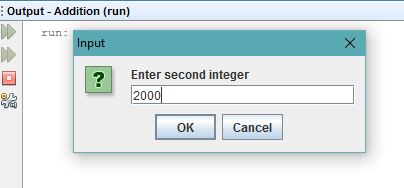
JOptionPane.PLAIN\_MESSAGE );

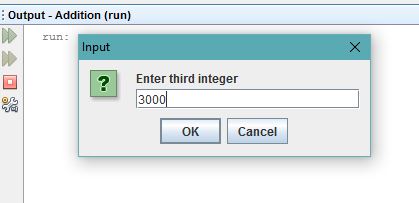
} // end method main

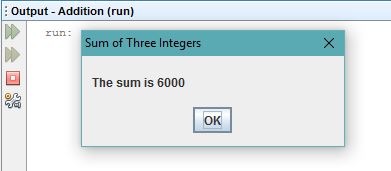
} // end class Addition

**Output:**

****

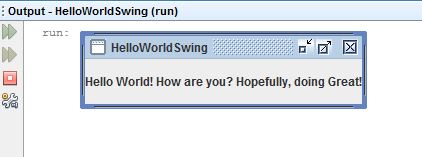
****

****

****

**Activity No 4:**

**Output:**

****

**Task 1:**

**Source Code:**

package unitsconverter;

/\*\*

\*

\* @author Hamad nasir

\*/

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

public class UnitsConverter implements ActionListener {

JFrame converterFrame;

JPanel converterPanel;

JTextField enteredNumber;

JLabel Jlabel1Conversion, conversionToJlabel;

JButton convertTemp;

JComboBox numberSequence;

public UnitsConverter() {

//formation and setting up the window

converterFrame = new JFrame("Convert between Different Quantities");

converterFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

converterFrame.setSize(new Dimension(400, 150));

//formation and setting up the panel.

converterPanel = new JPanel(new GridLayout(3, 3));

String[] units = {"Celsius To Fahrenheit" , "Meters to Feet",

"Kilograms(Kgs) to Pounds(LBs)", "Radians to Degrees(Angles)"};

numberSequence = new JComboBox(units);

//Adding the widgets.

addWidgets();

//Setting default button.

converterFrame.getRootPane().setDefaultButton(convertTemp);

//Adding panel to window.

converterFrame.getContentPane().add(converterPanel, BorderLayout.CENTER);

//Displaying window.

converterFrame.pack();

converterFrame.setVisible(true);

//Using the anonymous class to perform some action when individual items

//are selected in Drop Down List.

numberSequence.addItemListener(

new ItemListener(){

@Override

public void itemStateChanged(ItemEvent event){

if(event.getStateChange() == ItemEvent.SELECTED){

switch (numberSequence.getSelectedIndex()) {

case 0:

conversionToJlabel.setText(" Fahrenheit");

conversionToJlabel.setText("Celsius");

break;

case 1:

conversionToJlabel.setText(" Feet");

conversionToJlabel.setText("Meters");

break;

case 2:

conversionToJlabel.setText(" Pounds");

conversionToJlabel.setText("Kilograms(KGs)");

break;

default:

conversionToJlabel.setText(" Degrees");

conversionToJlabel.setText("Radians");

break;

}

}

}

}

);

}

/\*\*

\* Create and add the widgets.

\*/

private void addWidgets() {

//Create widgets.

enteredNumber = new JTextField(2);

conversionToJlabel = new JLabel("Celsius", SwingConstants.LEFT);

convertTemp = new JButton("Convert");

conversionToJlabel = new JLabel("Fahrenheit", SwingConstants.LEFT);

//Listen to events from the Convert button.

convertTemp.addActionListener(this);

//Add the widgets to the container.

converterPanel.add(enteredNumber);

converterPanel.add(conversionToJlabel);

converterPanel.add(convertTemp);

converterPanel.add(conversionToJlabel);

converterPanel.add(numberSequence);

conversionToJlabel.setBorder(BorderFactory.createEmptyBorder(9,9,9,9));

conversionToJlabel.setBorder(BorderFactory.createEmptyBorder(9,9,9,9));

}

public void actionPerformed(ActionEvent event) {

//Parse degrees Celsius as a double and convert to Fahrenheit.

//Performing specific functioning on specific items of Drop Down List

if(numberSequence.getSelectedIndex() == 0){

int tempFahr = (int)((Double.parseDouble(enteredNumber.getText())) \* 1.8 + 32);

conversionToJlabel.setText(tempFahr + " Fahrenheit");

}

if(numberSequence.getSelectedIndex() == 1){

double feet;

feet = Double.parseDouble((enteredNumber.getText())) \* 3.28;

conversionToJlabel.setText(feet + " feet");

}

if(numberSequence.getSelectedIndex() == 2){

double pounds = Double.parseDouble((enteredNumber.getText())) \* 2.204;

conversionToJlabel.setText(pounds + " pounds");

}

if(numberSequence.getSelectedIndex() == 3){

double degrees = Double.parseDouble((enteredNumber.getText())) \* 57.2958;

conversionToJlabel.setText(degrees + " Degrees");

}

}

public static void main(String[] args) {

//Make sure we have nice window decorations.

JFrame.setDefaultLookAndFeelDecorated(true);

UnitsConverter converter = new UnitsConverter();

}

}

**Output:**

