Lab # 9

# Activity # 1

## Output:

**What will be the output?**

run:

java.lang.IllegalArgumentException: Radius cannot be negative

Number of objects created: 1

BUILD SUCCESSFUL (total time: 0 seconds)

**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?**

It would compile but the ide won’t warn to use try catch block when the method is called. But in this case try and catch are still used so the program won’t crash.

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

The program crashes. (if we use throws attribute with method the ide won’t compile the code and would warn us to use try-catch block while if the throws attribute is not used the program crashes.

# Activity # 2

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

Statement4 would be executed

Statement5 would be executed in all cases unless the program is shutdown

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

Statement4 would be executed.

Statement 5 would be executed in all cases unless the program is shutdown

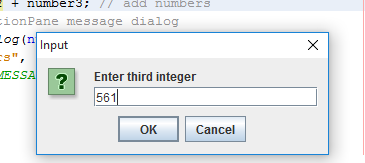
**If the exception is of type Exception2, will statement4 and statement5 be executed?**

Statement 4 would be executed but not the statement 5. The program would be crashed if the throwed exception is not handled somewhere other.

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

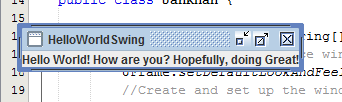
Since the exception is not handled so the program would crash so no statements would be executed except the one in the finally block.

# Activity #3



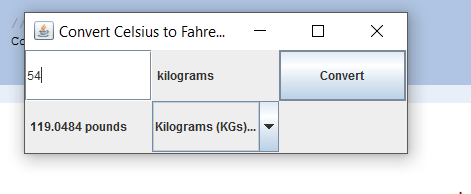
import javax.swing.JOptionPane; // program uses JOptionPane  
  
public class JanKhan {  
  
 public static void main(String[] args) {  
 // obtain user input from JOptionPane input dialogs  
 String firstNumber  
 = JOptionPane.showInputDialog("Enter first integer");  
 String secondNumber  
 = JOptionPane.showInputDialog("Enter 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 Two Integers",  
 JOptionPane.PLAIN\_MESSAGE);  
 } // end method main  
} // end class Addition

# Activity #4



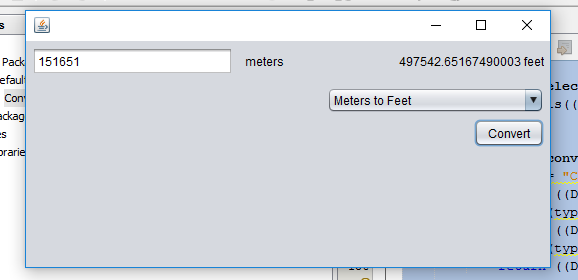
# TASK

## Method #1



package jankhan;  
  
  
import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*;  
  
class Converter implements ActionListener {  
  
 JFrame converterFrame;  
 JPanel converterPanel;  
 JTextField inputField;  
 JLabel inputLabel, outputLabel;  
 JButton convertButton;  
 String type;  
 String inputLabelString="Celcius";  
 String outputLabelString="Fahrenheit";  
   
 String converterTypes[]=new String[] {"Celsius to Fahrenheit","Meters to Feet","Kilograms (KGs) to Pounds (LBs)","Radians to Degrees (Angles)"};  
 JComboBox selectConverter;  
  
 public Converter() {  
   
   
   
 //Create and set up the window.  
 converterFrame = new JFrame("Convert Celsius to Fahrenheit");  
  
 converterFrame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 converterFrame.setSize(new Dimension(400, 150));  
 //Create and set up the panel.  
 converterPanel = new JPanel(new GridLayout(2, 2));  
 //Add the widgets.  
 addWidgets();  
 //Set the default button.  
 converterFrame.getRootPane().setDefaultButton(convertButton);  
 //Add the panel to the window.  
 converterFrame.getContentPane().add(converterPanel,  
 BorderLayout.CENTER);  
 //Display the window.  
 //converterFrame.pack();  
 converterFrame.setVisible(true);  
 }  
  
 /\*\*  
 \* Create and add the widgets.  
 \*/  
 private void addWidgets() {  
 //Create widgets.  
 inputField = new JTextField(2);  
 inputLabel = new JLabel(inputLabelString, SwingConstants.LEFT);  
 convertButton = new JButton("Convert");  
 outputLabel = new JLabel(outputLabelString,  
 SwingConstants.LEFT);  
 selectConverter = new JComboBox(converterTypes);  
   
   
   
   
 //Listen to events from the Convert button.  
 convertButton.addActionListener(this);  
 selectConverter.addActionListener(this);  
 //Add the widgets to the container.  
 converterPanel.add(inputField);  
 converterPanel.add(inputLabel);  
 converterPanel.add(convertButton);  
 converterPanel.add(outputLabel);  
 converterPanel.add(selectConverter);  
  
  
 inputLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));  
  
 outputLabel.setBorder(BorderFactory.createEmptyBorder(5, 5, 5, 5));  
 }  
  
 @Override  
 public void actionPerformed(ActionEvent event) {  
   
   
   
 if(event.getSource()==selectConverter)  
 {  
   
 updateLabels((String) selectConverter.getSelectedItem());  
 }   
 else if(event.getSource()==convertButton){  
 double answer=convert((String) selectConverter.getSelectedItem());  
 outputLabel.setText(answer + " " + outputLabelString);  
 }  
   
 }  
   
   
  
   
   
 public double convert(String type){  
 if(type=="Celsius to Fahrenheit"){  
 return((Double.parseDouble(inputField.getText())) \* 1.8 + 32);  
 }  
 else if(type=="Meters to Feet"){  
 return((Double.parseDouble(inputField.getText())) \* 3.2808399);  
 }  
 else if(type=="Kilograms (KGs) to Pounds (LBs)"){  
 return((Double.parseDouble(inputField.getText())) \* 2.2046);  
 }  
 else if(type=="Radians to Degrees (Angles)"){  
 return((Double.parseDouble(inputField.getText())) \* 57.295);  
 }  
 return 0;  
 }  
   
 public void updateLabels(String type){  
 if(type=="Celsius to Fahrenheit"){  
 inputLabelString="Degrees Celsius";  
 outputLabelString="Degrees Fahrenheit";  
   
 }  
 else if(type=="Meters to Feet"){  
 inputLabelString="meters";  
 outputLabelString="feet";  
 }  
 else if(type=="Kilograms (KGs) to Pounds (LBs)"){  
 inputLabelString="kilograms";  
 outputLabelString="pounds";  
 }  
 else if(type=="Radians to Degrees (Angles)"){  
 inputLabelString="radians";  
 outputLabelString="degrees";  
 }  
 inputLabel.setText(inputLabelString);  
 outputLabel.setText(outputLabelString);  
 }  
   
   
   
}  
  
public class JanKhan{  
   
   
 public static void main(String[] args) {  
 //Make sure we have nice window decorations.  
   
 //JFrame.setDefaultLookAndFeelDecorated(true);  
 Converter converter = new Converter();  
 }  
}

## Alternative method...jforms



/\*  
 \* 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.  
 \*/  
  
/\*\*  
 \*  
 \* @author ajan.bscs16seecs  
 \*/  
public class Converter extends javax.swing.JFrame {  
  
 /\*\*  
 \* Creates new form Converter  
 \*/  
 String converterTypes[] = new String[]{"Celsius to Fahrenheit", "Meters to Feet", "Kilograms (KGs) to Pounds (LBs)", "Radians to Degrees (Angles)"};  
 String type;  
 String inputLabelString = "Celcius";  
 String outputLabelString = "Fahrenheit";  
  
 public Converter() {  
 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() {  
  
 jPanel1 = new javax.swing.JPanel();  
 inputField = new javax.swing.JTextField();  
 inputLabel = new javax.swing.JLabel();  
 convertButton = new javax.swing.JButton();  
 selectConverter = new javax.swing.JComboBox<>();  
 outputLabel = new javax.swing.JLabel();  
  
 setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);  
  
 inputField.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 inputFieldActionPerformed(evt);  
 }  
 });  
  
 inputLabel.setText("celcius");  
  
 convertButton.setText("Convert");  
 convertButton.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 convertButtonActionPerformed(evt);  
 }  
 });  
  
 selectConverter.setModel(new javax.swing.DefaultComboBoxModel<>(new String[] { "Celsius to Fahrenheit", "Meters to Feet", "Kilograms (KGs) to Pounds (LBs)", "Radians to Degrees (Angles)" }));  
 selectConverter.addActionListener(new java.awt.event.ActionListener() {  
 public void actionPerformed(java.awt.event.ActionEvent evt) {  
 selectConverterActionPerformed(evt);  
 }  
 });  
  
 outputLabel.setText("fahrenheit");  
  
 javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);  
 jPanel1.setLayout(jPanel1Layout);  
 jPanel1Layout.setHorizontalGroup(  
 jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addGroup(jPanel1Layout.createSequentialGroup()  
 .addContainerGap()  
 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addGroup(jPanel1Layout.createSequentialGroup()  
 .addComponent(inputField, javax.swing.GroupLayout.DEFAULT\_SIZE, 156, Short.MAX\_VALUE)  
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
 .addComponent(inputLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 148, javax.swing.GroupLayout.PREFERRED\_SIZE)  
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
 .addComponent(outputLabel))  
 .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()  
 .addGap(0, 0, Short.MAX\_VALUE)  
 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addComponent(selectConverter, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  
 .addComponent(convertButton, javax.swing.GroupLayout.Alignment.TRAILING))))  
 .addContainerGap())  
 );  
 jPanel1Layout.setVerticalGroup(  
 jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addGroup(jPanel1Layout.createSequentialGroup()  
 .addContainerGap()  
 .addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
 .addComponent(inputField, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  
 .addComponent(inputLabel)  
 .addComponent(outputLabel))  
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
 .addComponent(selectConverter, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  
 .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  
 .addComponent(convertButton)  
 .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))  
 );  
  
 javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  
 getContentPane().setLayout(layout);  
 layout.setHorizontalGroup(  
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addComponent(jPanel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)  
 );  
 layout.setVerticalGroup(  
 layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
 .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  
 );  
  
 pack();  
 }// </editor-fold>   
  
 private void inputFieldActionPerformed(java.awt.event.ActionEvent evt) {   
 // TODO add your handling code here:  
 }   
  
 private void convertButtonActionPerformed(java.awt.event.ActionEvent evt) {   
 double answer = convert((String) selectConverter.getSelectedItem());  
 outputLabel.setText(answer + " " + outputLabelString);  
  
 }   
  
 private void selectConverterActionPerformed(java.awt.event.ActionEvent evt) {   
 updateLabels((String) selectConverter.getSelectedItem());  
 }   
  
 public double convert(String type) {  
 if (type == "Celsius to Fahrenheit") {  
 return ((Double.parseDouble(inputField.getText())) \* 1.8 + 32);  
 } else if (type == "Meters to Feet") {  
 return ((Double.parseDouble(inputField.getText())) \* 3.2808399);  
 } else if (type == "Kilograms (KGs) to Pounds (LBs)") {  
 return ((Double.parseDouble(inputField.getText())) \* 2.2046);  
 } else if (type == "Radians to Degrees (Angles)") {  
 return ((Double.parseDouble(inputField.getText())) \* 57.295);  
 }  
 return 0;  
 }  
  
 public void updateLabels(String type) {  
 if (type == "Celsius to Fahrenheit") {  
 inputLabelString = "Degrees Celsius";  
 outputLabelString = "Degrees Fahrenheit";  
  
 } else if (type == "Meters to Feet") {  
 inputLabelString = "meters";  
 outputLabelString = "feet";  
 } else if (type == "Kilograms (KGs) to Pounds (LBs)") {  
 inputLabelString = "kilograms";  
 outputLabelString = "pounds";  
 } else if (type == "Radians to Degrees (Angles)") {  
 inputLabelString = "radians";  
 outputLabelString = "degrees";  
 }  
 inputLabel.setText(inputLabelString);  
 outputLabel.setText(outputLabelString);  
 }  
  
 /\*\*  
 \* @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(Converter.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
 } catch (InstantiationException ex) {  
 java.util.logging.Logger.getLogger(Converter.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
 } catch (IllegalAccessException ex) {  
 java.util.logging.Logger.getLogger(Converter.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  
 } catch (javax.swing.UnsupportedLookAndFeelException ex) {  
 java.util.logging.Logger.getLogger(Converter.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 Converter().setVisible(true);  
 }  
 });  
 }  
  
 // Variables declaration - do not modify   
 private javax.swing.JButton convertButton;  
 private javax.swing.JTextField inputField;  
 private javax.swing.JLabel inputLabel;  
 private javax.swing.JPanel jPanel1;  
 private javax.swing.JLabel outputLabel;  
 private javax.swing.JComboBox<String> selectConverter;  
 // End of variables declaration   
}