**Assignment 6 Turn in Sheet Name: Tyler Quayle**

Lab Questions (**Total 50 Pts.**)

Big Java, Late Objects / Java for Everyone, 2e

**Chapter Number: 7 Exceptions**

1.1) (5 pts) Start with the code below and complete the getInt method. The method should prompt the user to enter an integer. Scan the input the user types. If the input is not an int, throw an IllegalArgumentException; otherwise, return the int.

|  |
| --- |
| import java.util.Scanner;  public class Throwing  {  public static void main(String[] args)  {  int x = getInt();  System.out.println(x);  }  public static int getInt()  {  Scanner input = new Scanner(System.in);  System.out.print("Please Enter an Integer: ");  if (!input.hasNextInt()) { throw new IllegalArgumentException(); }  int temp = input.nextInt();  return temp;  }  }  **OUTPUT: INVALID INPUT**  **Please Enter an Integer: test**  **Exception in thread "main" java.lang.IllegalArgumentException**  **OUTPUT: VALID INPUT**  **Please Enter an Integer: 123**  **123** |

1.2) (5 pts) Modify the program from Exercise 1.1 so that getInt throws an IOException instead of an IllegalArgumentException. Modify the main program so that it catches and prints the IOException.

|  |
| --- |
| import java.util.Scanner;  public class Throwing  {  public static void main(String[] args)  {  int x = getInt();  System.out.println(x);  }  public static int getInt()  {  Scanner input = new Scanner(System.in);  System.out.print("Please Enter an Integer: ");  try  {  if (!input.hasNextInt()) { throw new IOException("Invalid Input"); }  }  catch(IOException e)  {  System.out.println(e);  }  int temp = input.nextInt();  return temp;    }  }  **OUTPUT: INVALID INPUT**  **Please Enter an Integer: TEST**  **java.io.IOException: Invalid Input**  **OUTPUT: VALID INPUT**  **Please Enter an Integer: 123**  **123** |

2.1) (5 pts) Write a program that prompts the user to enter an int. Print the integer the user enters. If the user doesn’t enter an int, keep looping, prompting the user to enter the required int. Write this solution without exceptions.

|  |
| --- |
| public static void main(String[] args)  {  Scanner input = new Scanner(System.in);    System.out.print("Please Enter an Integer: ");    while (!input.hasNextInt()) {  System.out.print("ERROR - Please enter a number: ");  input.next();}    int x = input.nextInt();  System.out.println(x);  }  **OUTPUT: INVALID INPUT**  **Please Enter an Integer: Test**  **ERROR - Please enter a number: Testing**  **ERROR - Please enter a number: 123**  **123**  **OUTPUT: VALID INPUT**  **Please Enter an Integer: 123**  **123** |

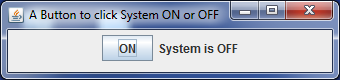
2.2) (5 pts) Modify/Rewrite the program in 2.1 so that it does the same task using exceptions.

|  |
| --- |
| public static void main(String[] args)  {  Scanner input = new Scanner(System.in);  int x;  while(true)  {  System.out.print("Please Enter an Integer: ");  try  {  if (!input.hasNextInt()) { throw new Exception();}  x = input.nextInt();  System.out.println(x);  break;  }    catch(Exception e)  {  System.out.println("Error: Invalid input");  input.next();  }  }  }  **OUTPUT: INVALID INPUT**  **Please Enter an Integer: TEST**  **Error: Invalid input**  **Please Enter an Integer: 123**  **123** |

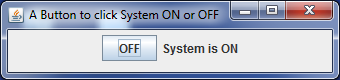
**Chapter Number: 10 Graphical User Interfaces**

3) (30 pts) Write a main program that implements the following functionality:

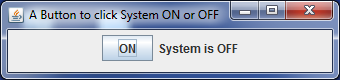
A Button is shown and a Label next to it telling about the system status:



If the button is clicked the button text AND the system status changes.



This toggles every time you click the button:



Use a class “MyCustomFrame” that extends JFrame in a separate file from the main class file! Use the nesting recommendations discussed in class.

|  |
| --- |
| **IN MAIN JAVA CLASS**  **public static void main(String[] args) {**  **MyCustomFrame myFrame = new MyCustomFrame();**  **myFrame.setButtonText("ON");**  **myFrame.setButtonToggleText("OFF");**  **myFrame.setLabelText("System is OFF");**  **myFrame.setLabelToggleText("System is ON");**  **myFrame.addButton();**  **myFrame.addLabel();**  **}** |
| **IN MYCUSTOMFRAME.JAVA**  **public class MyCustomFrame extends JFrame implements ActionListener{**    **private JFrame frame = new JFrame();**  **private JButton newButton = new JButton();**  **private JPanel panel = new JPanel();**  **private boolean switchToggle = true;**  **private String buttonText, buttonToggleText, labelText, labelToggleText;**  **private final int FRAME\_WIDTH = 350, FRAME\_HEIGHT = 100;**    **// DEFAULT CONSTRUCTOR TO SET THE FRAME UP**  **public MyCustomFrame()**  **{**  **frame.setSize(FRAME\_WIDTH, FRAME\_HEIGHT);**  **frame.setTitle("A Button to click System ON or OFF");**  **frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);**  **frame.setVisible(true);**  **}**    **//SETS THE BUTTON TEXT AND THE TEXT WHEN BUTTON IS PRESSED**  **public void setButtonText(String t) { buttonText = t; }**  **public void setButtonToggleText(String t) { buttonToggleText = t; }**    **//SETS THE LABEL TEXT AND THE TEXT WHEN BUTTON IS PRESSED**  **public void setLabelText(String t) { labelText = t; }**  **public void setLabelToggleText(String t) { labelToggleText = t; }**    **//USED TO CLEAR AND UPDATE PANELS WHEN BUTTON IS PRESSED, OTHERWISE IT**  **//WILL KEEP ADDING MORE AND MORE TEXT WITHOUT CLEARING OLD TEXT**  **public void updatePanel() { frame.add(panel);}**  **public void clearPanel(){panel.removeAll(); panel.updateUI();}**    **//SETS THE INTIAL STATE OF THE BUTTON**  **public void addButton()**  **{**  **newButton.setText(buttonText);**  **//used to create button of constant width**  **//newButton.setPreferredSize(new Dimension(80,30));**  **newButton.addActionListener(this);**  **panel.add(newButton);**  **updatePanel();**  **}**    **//SETS THE INTIAL STATE OF THE LABEL**  **public void addLabel() {panel.add(new JLabel(labelText));}**    **// ACTIONLISTENER TO CHECK IF BUTTON IS PRESSED AND UPDATES ACCORDINGLY**  **@Override**  **public void actionPerformed(ActionEvent e)**  **{**  **clearPanel();**  **if(switchToggle)**  **{**  **newButton.setText(buttonToggleText);**  **panel.add(newButton);**  **panel.add(new JLabel(labelToggleText));**  **}**  **else**  **{**  **newButton.setText(buttonText);**  **panel.add(newButton);**  **panel.add(new JLabel(labelText));**  **}**  **updatePanel();**  **switchToggle = !switchToggle;**  **}**  **}**  **OUTPUT:** |

Will also upload the 4 Java Files

* Assignment\_06\_Question\_01.java
* Assignment\_06\_Question\_02.java
* Assignment\_06\_Question\_03.java
  + MyCustomFrame.java