C:\Users\benkimz\Desktop\Java\projects\InputApproaches.java

```java
// SCANNER INPUT APPROACH

/**
 *
 * @author benkimz
 */

import java.util.Scanner;

public class ScannerInput {

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {

        // Create an instance of the Scanner class
        Scanner userinput = new Scanner(System.in);
        /*
            STEPS
            ----------------------
            1). initialize a variable
            2). Prompt the user for a value
            3). assign value to the variable
            4). display the entered value
        */

        //Initializing a variable `username`
        String username;

        // displaying a prompt
        System.out.print("Enter a username: ");

        /*
            We use the instance of the Scanner class to take user's
            input and assign the value to our variable `username`
        */
        username = userinput.nextLine();

        //Print the `username` on screen in a welcome message
        System.out.println("Welcome, " + username);

        /*
            NOTE
            ---------------
            1). You can collapse variable initialization and
                value assignment into one statement as follows.
        */

        // display the prompt
        System.out.print("Enter username 2: ");
        //initialize and assign value
        String username2 = userinput.nextLine();
        //print the value
        System.out.println("Welcome, " + username2);

        //Prompting the user for numbers Example
        //variables initialization
        int x, y;
        //prompt the user for `x`
        System.out.print("Enter [x]: ");
        x = userinput.nextInt();
        //prompt the user for `y`
        System.out.print("Enter [y]: ");
        y = userinput.nextInt();

```

```
67          //do some operations if you like
68          int sum = x + y;
69          int product = x * y;
70          //display results
71          System.out.println("The sum is: " + sum);
72          System.out.println("The product is: " + product);
73
74      }
75  }
76
77
78
79  /*
80  ******************************************************************************
81  */
82
83  // BUFFERED READER APPROACH
84
85  /**
86   *
87   * @author benkimz
88   */
89
90  import java.io.*;
91
92  public class BufferedReaderApproach {
93
94      /**
95       * @param args the command line arguments
96       */
97      public static void main(String args[]) {
98
99          //create an instance of the BufferedReader class
100         BufferedReader reader = new BufferedReader(
101                 new InputStreamReader(System.in)
102         );
103
104         /*
105             STEPS
106             ----------------------
107             1). initialize a variable
108             2). Prompt the user for a value
109             3). assign value to the variable
110             4). display the entered value
111         */
112
113         //Initializing a variable `username`
114         String username;
115         // displaying a prompt
116         System.out.print("Enter a username: ");
117
118         //We must catch unexpected IOExceptions when using this approach
119         try{
120             /*
121                 We use the instance of the BuffredReader class to take user's
122                 input and assign the value to our variable `username`
123             */
124             username = reader.readLine();
125             //Print the `username` on screen in a welcome message
126             System.out.println("Welcome, " + username);
127         }catch(IOException error){
128             // Incase of an exception
129             String errorMessage = error.getMessage();
130             System.out.println("Oops! An error occured! Info: "+errorMessage);
131         }
132
133         //Prompting the user for numbers Example
134         /*
135             STEPS
```

```
136             ----------------------
137             1). initialize a variable
138             2). Prompt the user for an input
139             3). Parse the raw input to be of the desired data-type
140             4). assign result to the variable
141             5). Do operations
142         */
143 
144         //initialize two integer variables `x` & `y`
145         int x, y;
146 
147         //We must catch unexpected IOExceptions when using this approach
148         try{
149             /*
150                 We use the instance of the BuffredReader class to take user's
151                 input and assign the value to our variable `username`
152             */
153 
154             //prompt for the value of `x`
155             System.out.print("Enter [x]: ");
156             x = Integer.parseInt(reader.readLine());
157 
158             // prompt for the value of `y`
159             System.out.print("Enter [y]: ");
160             y = Integer.parseInt(reader.readLine());
161 
162             // take the sum
163 
164             int sum = x + y;
165             //Print the sum of `x` & `y` on screen
166             System.out.println("The sum of " + x + " and " + y + " is " + sum);
167         }catch(IOException error){
168             // Incase of an exception
169             String errorMessage = error.getMessage();
170             System.out.println("Oops! An error occured! Info: "+errorMessage);
171         }
172 
173     }
174 }
175 
176 
177 
178 /*
179 ********************************************************************************
180 */
181 
182 // GUI APPROACH
183 /**
184  *
185  * @author benkimz
186  */
187 
188 import java.awt.*;
189 import java.awt.event.*;
190 
191 public class GUIApproach extends Frame implements MouseListener{
192     /*
193         STEPS
194         ---------------------
195         1). Create the textfields and their labels
196         2). Set a suitable bounding rectangle (Position on canvas) for each.
197         3). Add the above controls to the Frame object
198         4). Take the user's input
199         5). Perform operations
200     */
201 
202     // `username` input controls
203     TextField username = new TextField(255);
204     Label usernameLabel = new Label("Enter username: ");
```

```
205
206     // `age` input controls
207     TextField age = new TextField(3);
208     Label ageLabel = new Label("Enter age: ");
209
210     // `button` control
211     Button button = new Button("done");
212
213     // Method to be invoked when button is clicked
214     @Override
215     public void mouseClicked(MouseEvent e) {
216         // initialize value holders
217         String usernameValue; int ageValue;
218         // get the value for `username`
219         usernameValue = username.getText();
220         // get the value for `age` and parse to be of [int] data-type
221         ageValue = Integer.parseInt(age.getText());
222         System.out.println("Welcome, " + usernameValue + ".");
223         System.out.println("Age received: " + ageValue + ", have a great day!");
224     }
225
226     // constructor method
227     public GUIApproach(String arg){
228         // call the parent class constructor method (Frame class constructor)
229         super(arg);
230         setLayout(null); //let's use a null layout for now
231         // set the bounding rectangle for username input controls
232         usernameLabel.setBounds(20,50,100,30);
233         username.setBounds(150,50,100,30);
234         // Add the controls to the Frame object
235         add(usernameLabel, 0);
236         add(username, 1);
237         // set the bounding rectangle for age input controls
238         ageLabel.setBounds(20,90,100,30);
239         age.setBounds(150,90,100,30);
240         // Add the controls to the Frame object
241         add(ageLabel, 2);
242         add(age, 3);
243         //Add a button
244         button.setBounds(150,150,100,30);
245         //Add an event listener to the button
246         button.addMouseListener(this);
247         add(button);
248         /*
249             The code for the close button
250         */
251         this.addWindowListener(new WindowAdapter() {
252             @Override
253             public void windowClosing(WindowEvent event){
254                 System.exit(0);
255             }
256         });
257     }
258
259     public static void main(String args[]) {
260
261         /*
262             We want a simple form with a title: `User Input-Form`
263             STEPS
264             --------------------------
265             1). create an instance of the `GUIApproach class (extends Frame)
266             2). set the width and height to display on screen
267             3). set visibility to [true]
268         */
269
270         GUIApproach gui = new GUIApproach("User Input-Form");
271         gui.setSize(450, 250);
272         gui.setVisible(true);
273     }
```

```
274
275     // consume these methods of the MouseListener interface because we
276     // don't really need them
277     @Override
278     public void mousePressed(MouseEvent e) {
279         e.consume();
280     }
281
282     @Override
283     public void mouseReleased(MouseEvent e) {
284         e.consume();
285     }
286
287     @Override
288     public void mouseEntered(MouseEvent e) {
289         e.consume();
290     }
291
292     @Override
293     public void mouseExited(MouseEvent e) {
294         e.consume();
295     }
296 }
```