An input dialog box returns the user’s input as a string. If the program expects numeric or Boolean input then your program must scan the string and convert its contents to the equivalent values in binary. If the input string contains only one token, this is most easily done using static methods from the wrapper classes.

Each of the eight wrapper classes except char has a static method that scans a string into the primitive datum represented by the string.

<table>
<thead>
<tr>
<th>Location in the API</th>
<th>The Parse Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>java.lang.Byte</td>
<td>static byte parseByte( String s )</td>
</tr>
<tr>
<td>java.lang.Short</td>
<td>static short parseShort( String s )</td>
</tr>
<tr>
<td>java.lang.Integer</td>
<td>static int parseInt( String s )</td>
</tr>
<tr>
<td>java.lang.Long</td>
<td>static long parseLong( String s )</td>
</tr>
<tr>
<td>java.lang.Boolean</td>
<td>static boolean parseBoolean( String s )</td>
</tr>
<tr>
<td>java.lang.Float</td>
<td>static float parseFloat( String s )</td>
</tr>
<tr>
<td>java.lang.Double</td>
<td>static double parseFloat( String s )</td>
</tr>
</tbody>
</table>

**Examples**

```java
int k = Integer.parseInt( "100" );  k = 100

boolean b = Boolean.parseBoolean( "true" );  b = true

double d = Double.parseDouble( "2.5" );  d = 2.5
```
To use these to scan dialog box input, simply pass the input string to the appropriate parse method.

**Example**
The code fragment below reads a `double` value from the input dialog shown to the right and stores it into variable `weight`.

```java
import static javax.swing.JOptionPane.*;
import static java.lang.Double.*;

String input;
input = showInputDialog( "Enter patient's weight" );
double weight = parseDouble( input );
```

If the argument to `parseX` does not correctly represent a datum of primitive data type `X`, the method throws a `NumberFormatException`.

**Example**
The code fragment of the previous example, if given input as shown to the right, produces the run-time error message shown below.

```
Exception in thread "main" java.lang.NumberFormatException
  For input string: "135.5 lbs."
```
Exercises

Three common errors that users make using an input dialog box are:

- Canceling the dialog box when the program expects input
- Clicking OK without entering input in the text field.
- Entering a string that cannot scan to the primitive data type that the program expects.

Enter the application given below into jGRASP, save it to a file and compile it. Do the exercises that follow.

```java
import static javax.swing.JOptionPane.*;
import static java.lang.Integer.*;
import java.text.DecimalFormat;

public class Percent {
    public static void main( String [] args ) {
        String prompt, input, output;
        int score;
        double pctg;
        prompt = "Enter student's score";
        input = showInputDialog( prompt );
        score = parseInt( input );
        pctg = score / 100.0;
        DecimalFormat df = new DecimalFormat( "##.0%" );
        output = score + "/100 = " + df.format( pctg );
        showMessageDialog( null, output );
    }
}
```

1. Run the program; enter 82 in the text field; click the OK button. Observe the output.
2. Run the program; enter 82 in the text field; press the Enter key. Observe the output.
3. Run the program; enter 82 points in the text field; press the Enter key. Observe the run-time error message. Explain the error.
4. Run the program; enter 82.5 in the text field; press the Enter key. Observe the run-time error message. Explain the error.
5. Run the program; click the OK button without entering anything in the text field. Observe the run-time error message. Explain the error.

6. Run the program; click the Cancel button. Observe the run-time error message. Explain the error.

For each of the following, write the Java code that does it using a `JOptionPane` input dialog and the appropriate parse method from the wrapper classes. Invent and declare any variables that you need.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>Read the height of a building (a floating-point value).</td>
</tr>
<tr>
<td>8.</td>
<td>Read whether or not an employee is married (a Boolean value).</td>
</tr>
<tr>
<td>9.</td>
<td>Read the number of exemptions an employee has filled out on his or her W-4 form (an integer).</td>
</tr>
<tr>
<td>10.</td>
<td>Read a person’s age (an integer).</td>
</tr>
</tbody>
</table>