

## Interactive Input

- ❖ Swing Graphical User Interface (GUI)
  - ◆ Via Input Dialog Box (Chapter 4.2)
  - ◆ Via Input Controls (Chapter 10)
  - ◆ Via Console Window (Chapter 4.1)
    - ◆ Pre-Version 5.0 use System.in
    - ◆ Version 5 JDK use new Scanner Class
- ❖ Interactive input data prompts user and is entered with keyboard
- ❖ Batch input data is input to program as file

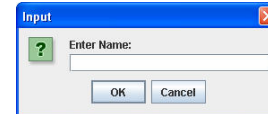
Copyright © 2006 R.M. Laurie 1

## Interactive Dialog Input

- ❖ GUI method of entering user data:
  - ◆ Method named `showInputDialog()` in `JOptionPane` class
    - ◆ Creates dialog box user user to enter string
- ❖ Syntax:
 

```
String JOptionPane.showInputDialog(string);
```
- ❖ Example:
 

```
sName=JOptionPane.showInputDialog("Enter Name:");
```

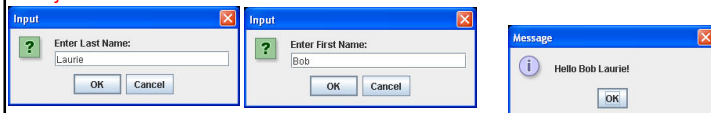


Copyright © 2006 R.M. Laurie 2

## String Input Example

- ❖ This program prompts for user Last Name and then First Name and displays greets user

```
import javax.swing.*;
public class InNames
{
    public static void main(String args[])
    {
        String sFirstName, sLastName, sOutput;
        sLastName = JOptionPane.showInputDialog("Enter Last Name:");
        sFirstName = JOptionPane.showInputDialog("Enter First Name:");
        sOutput = "Hello " + sFirstName + " " + sLastName + "!";
        JOptionPane.showMessageDialog(null, sOutput);
        System.exit(0);
    }
}
```



Copyright © 2006 R.M. Laurie 3

## String to Number Conversion

- ❖ Keyboard Input is always String Data
  - ◆ Entered numerical data requires parsing the entered String to the specified numerical data type
  - ◆ Parsing must be done prior to performing mathematical operations

Table 4.2: Java Conversion Routines

Class	Method	Description	Example	Returned Value
Integer	<code>parseInt(string)</code>	Converts a string to a primitive type int.	<code>Integer.parseInt("1234")</code>	1234 (an int value)
Long	<code>parseLong(string)</code>	Converts a string to a primitive type long.	<code>Long.parseLong("128365489")</code>	128365489L (a long value)
Float	<code>parseFloat(string)</code>	Converts a string to a primitive type float.	<code>Float.parseFloat("345.89")</code>	345.89f (a float value)
Double	<code>parseDouble(string)</code>	Converts a string to a primitive type double.	<code>Double.parseDouble("2.3456789")</code>	2.3456789 (a double value)

4

### Dialog Box Input/Output

```

import javax.swing.*;
public class InOperations
{
    public static void main(String args[])
    {
        int nNum1, nNum2;
        String sEntry, sOutput="Operators Example\nby Bob Laurie";
        JOptionPane.showMessageDialog(null,sOutput);
        sEntry = JOptionPane.showInputDialog("Enter 1st Number:");
        nNum1 = Integer.parseInt(sEntry);
        sEntry = JOptionPane.showInputDialog("Enter 2nd Number:");
        nNum2 = Integer.parseInt(sEntry);
        sOutput = nNum1+" + " + nNum2 + " = " + (nNum1 + nNum2);
        JOptionPane.showMessageDialog(null, sOutput);
        sOutput = nNum1+" - " + nNum2 + " = " + (nNum1 - nNum2);
        JOptionPane.showMessageDialog(null, sOutput);
        sOutput = nNum1+" x " + nNum2 + " = " + (nNum1 * nNum2);
        JOptionPane.showMessageDialog(null, sOutput);
        System.exit(0);
    }
}
    
```

### Dialog Box Input & Console Output

```

import javax.swing.*;
public class InOperationsOut
{
    public static void main(String args[])
    {
        int nNum1, nNum2;
        String sEntry;
        sEntry = JOptionPane.showInputDialog("Enter 1st Number:");
        nNum1 = Integer.parseInt(sEntry);
        sEntry = JOptionPane.showInputDialog("Enter 2nd Number:");
        nNum2 = Integer.parseInt(sEntry);
        System.out.println("Operators Output\nby Bob Laurie");
        System.out.println(nNum1+" + "+nNum2+" = "+(nNum1+nNum2));
        System.out.println(nNum1+" - "+nNum2+" = "+(nNum1-nNum2));
        System.out.println(nNum1+" x "+nNum2+" = "+(nNum1*nNum2));
        System.exit(0);
    }
}
    
```

## Formatted Output

❖ Integer and floating-point numbers can be controlled by `format()` method

- ◆ Especially useful in printing columns with numbers
- ◆ Import statement for `java.text` package of classes
- ◆ In class `java.text.DecimalFormat`
- ◆ Statement within `main()` uses new operator `DecimalFormat num = new DecimalFormat("000");`
- ◆ `format()` method applies format string to number

Symbol	Description
#	A digit placeholder; zero shows as absent and not as a space
0	A digit placeholder and automatic fill character
.	Decimal placeholder
,	Grouping separator
;	Separate positive and negative format strings
%	Multiply by 100 and add a % sign

### Dialog Box I/O & Console Output

```

import javax.swing.*;
import java.text.*;
public class InOperationsOutDia
{
    public static void main(String args[])
    {
        double dNum1, dNum2;
        String sEntry, sOutput;
        DecimalFormat oDlr = new DecimalFormat("$,##0.00");
        DecimalFormat oPercent = new DecimalFormat("###0.00%");
        sEntry = JOptionPane.showInputDialog("Enter 1st Price:");
        dNum1 = Double.parseDouble(sEntry);
        sEntry = JOptionPane.showInputDialog("Enter 2nd Price:");
        dNum2 = Double.parseDouble(sEntry);
        sOutput = "Difference and Percent Program\n\n"+ oDlr.format(dNum1)
            +" - " + oDlr.format(dNum2)+" = "+oDlr.format(dNum1-dNum2)+"\n\n"
            +"$"+dNum1+"/$"+dNum2+" = " + oPercent.format(dNum1/dNum2)+"\n\n";
        JOptionPane.showMessageDialog(null, sOutput);
        System.out.println(sOutput);
        System.exit(0);
    }
}
    
```