



Lecture (03)

String Variables, formatted String & Handling user input

Dr. Ahmed ElShafee



Agenda

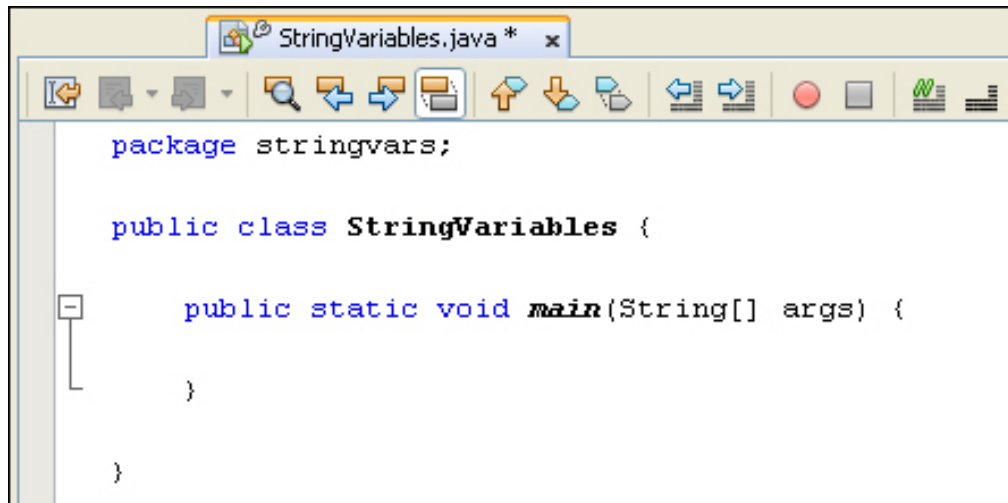
- Introduction
- Mode advanced printing methods
- Accepting Input from a User
- Option Panes (GUI)
- Exercises
- Assignments

Introduction

- As well as storing number values, variables can hold text.
- You can store just one character, or lots of characters.
- To store just one character, the **char variable** is used.
- Usually, though, you'll want to store more than one character.
- To do so, you need the **string variable type**.

StringVars01 example

- Start a new project from NetBeans.
- Make sure **Java and Java Application** are selected.
- Project Name is StringVars, and the Class name is StringVariables.



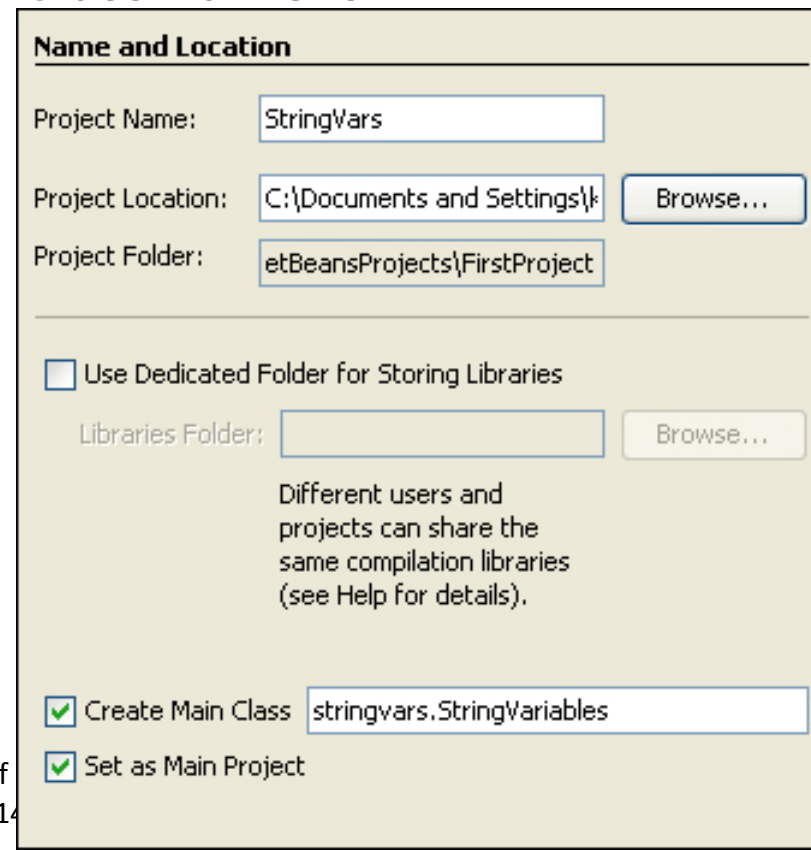
```
package stringvars;

public class StringVariables {

    public static void main(String[] args) {

    }

}
```



Name and Location

Project Name:

Project Location:

Project Folder:

Use Dedicated Folder for Storing Libraries

Libraries Folder:

Different users and projects can share the same compilation libraries (see Help for details).

Create Main Class

Set as Main Project

-
- To set up a string variable, you type the word **String** followed **by a name for your** variable.
 - Note that there's an uppercase "S" for String. Again, a semicolon ends the line:

String first_name;

- Assign a value to your new string variable by typing an equals sign.
- After the equals sign the text you want to store goes between two sets of double quotes:

first_name = "Ahmed";

- If you prefer, you can have all that on one line:

String first_name = "Ahmed";

-
- Set up a second string variable to hold a surname/family name:

String family_name = "Soliman";

- To print both names, add the following `println()`:

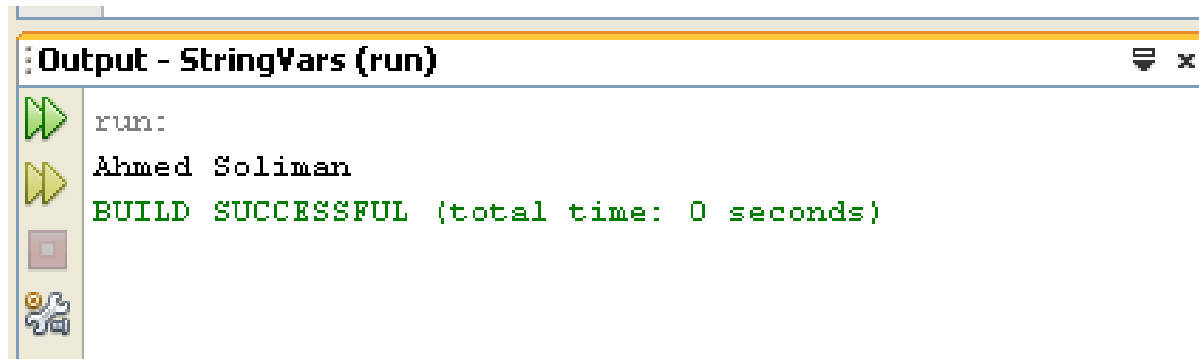
System.out.println(first_name + " " + family_name);

- In between the round brackets of `println()`, we have this:

first_name + " " + family_name

```
1  +  /*...*/  
5  package stringvars01;  
6  
7  +  /**...*/  
11 public class StringVars01 {  
12  
13  +   /**...*/  
16  -   public static void main(String[] args) {  
17     String first_name="Ahmed";  
18     String second_name="Soliman";  
19     System.out.println(first_name + " | " +second_name );  
20  }  
21 }  
22
```

Run your program and you should see this in the Output window:



```
Output - StringVars (run)  
run:  
Ahmed Soliman  
BUILD SUCCESSFUL (total time: 0 seconds)
```

StringVars02 example

- If you are storing just a single character, then the variable you need is **char** (lowercase “c”).
- To store the character, you use single quotes instead of double quotes.
- Here’s our program again, but this time with the **char variable**:

```
Output - StringVars02 (run)
run:
A S
BUILD SUCCESSFUL (total time: 0 seconds)
```

More advanced printing method

- Printf() method enables you to print formatted string, by placing a certain symbols inside printed strings.
- These symbols refer to variables passed as parameters after the formatted string and separated by commas.
- Example:

Data type	symbol	function
char	%c	Printf("this is char: %c", ch);
String	%s	Printf("my name is %s %s", first_name, last_name);
integers	%d	Printf("%d + %d = %d", num1, num2, answer);
double	%f	Printf("%f + %f = %f", num1, num2, answer);

printfexample01

```
1  +  /**...*/
5  package printfexample01;
6
7  +  /**...*/
11 public class PrintfExample01 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      char ch1='c';
18      char ch2='D';
19      String first_name="Ahmed";
20      String last_name="Soliman";
21      int inum1,inum2,ianswer;
22      inum1=45;
23      inum2=85;
24      ianswer=inum1+inum2;
25      double dnum1, dnum2, danswer;
26      dnum1=44.55;
27      dnum2=45.23;
28      danswer=dnum1+dnum2;
29      System.out.printf("\nFirst char is: %c, and the second char is: %c", ch1,ch2) ;
30      System.out.printf("\nMy name is: %s %s",first_name , last_name) ;
31      System.out.printf("\nInteger numbers operation: %d + %d = %d",inum1 ,inum2,ianswer) ;
32      System.out.printf("\nDouble numbers operation: %f + %f = %f\n",dnum1 ,dnum2, danswer) ;
33
34  }
35 }
36
```

Output - PrintfExample01 (run)

run:

First char is: c, and the second char is: D

My name is: Ahmed Soliman

Integer numbers operation: 45 + 85 = 130

Double numbers operation: 44.550000 + 45.230000 = 89.780000

BUILD SUCCESSFUL (total time: 1 second)

Data type	symbol	function
short	%d	Printf(“%d + %d = %d”, num1, num2, answer);
float	%f	Printf(“%f + %f = %f”, num1, num2, answer);
byte	%d	Printf(“%d + %d = %d”, num1, num2, answer);
Boolean	%b	Printf(“yes this is %b”, 1);
Hexadecimal	%x	Printf(“0x%x + 0x%x = 0x%x”, num1, num2, answer);
Octal	%o	Printf(“%o + %o = %o”, num1, num2, answer);

printfexample02

```
1  +  /*...*/
5  package printfexample02;
6
7  +  /**...*/
11 public class PrintfExample02 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17
18      short snum1,snum2;
19      short sanswer;
20      snum1=56; snum2=10;
21      sanswer = (short) (snum1+snum2);
22      System.out.printf("\nshort numbers operation: %s + %s = %s", snum1, snum2, sanswer);
23      float f1,f2,f3;
24      f1=3.456f;
25      f2=45.23431f;
26      f3=f1/f2;
27      System.out.printf("\nfloat numbers operation: %f / %f = %f", f1, f2, f3);
28      byte b1,b2,b3;
29      b1=34;
30      b2=22;
31      b3=(byte) (b1+b2);
32      System.out.printf("\nByte numbers operation: %d + %d = %d\n", b1, b2, b3);
33
34  }
35 }
36
```

Output - PrintfExample02 (run)

```
run:
short numbers operation: 56 + 10 = 66
float numbers operation: 3.456000 / 45.234310 = 0.076402
Byte numbers operation: 34 + 22 = 56
BUILD SUCCESSFUL (total time: 1 second)
```

printfexample03

```
1  +  /*...*/
5  package printfexample03;
6
7  +  /**...*/
11 public class Printfexample03 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      boolean b1,b2;
18      b1=true;
19      b2=false;
20      System.out.printf("%nBoolean operation: %b xor %b = %b.",b1, b1, (b1^b1));
21      System.out.printf("%nBoolean operation: %b xor %b = %b.",b1, b2, (b1^b2));
22      System.out.printf("%nBoolean operation: %b xor %b = %b.",b2, b1, (b2^b1));
23      System.out.printf("%nBoolean operation: %b xor %b = %b.\n",b2, b2, (b2^b2));
24  }
```

```
Output - printfexample03 (run)
run:
Boolean operation: true xor true = false.
Boolean operation: true xor false = true.
Boolean operation: false xor true = true.
Boolean operation: false xor false = false.
BUILD SUCCESSFUL (total time: 0 seconds)
```

printfexample04

```
1  +  /*...*/
5  package printfexample04;
6
7  +  /**...*/
11 public class Printfexample04 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      int n1,n2,n3;
18      n1=0x4c;
19      n2=526;
20      n3=067;
21      System.out.printf("%nDecimal= %d, Hexdecimal= %x, Octal= %o", n1,n1,n1);
22      System.out.printf("%nDecimal= %d, Hexdecimal= %x, Octal= %o", n2,n2,n2);
23      System.out.printf("%nDecimal= %d, Hexdecimal= %x, Octal= %o\n", n3,n3,n3);
24  }
25  }
26
```

Output - printfexample04 (run)

```
run:
Decimal= 76, Hexdecimal= 4c, Octal= 114
Decimal= 526, Hexdecimal= 20e, Octal= 1016
Decimal= 55, Hexdecimal= 37, Octal= 67
BUILD SUCCESSFUL (total time: 0 seconds)
```

- Escape sequence

Escape Sequence	Name	Description
<code>\a</code>	Bell (alert)	Makes a sound from the computer
<code>\b</code>	Backspace	Takes the cursor back
<code>\t</code>	Horizontal Tab	Takes the cursor to the next tab stop
<code>\n</code>	New line	Takes the cursor to the beginning of the next line
<code>\v</code>	Vertical Tab	Performs a vertical tab
<code>\f</code>	Form feed	
<code>\r</code>	Carriage return	Causes a carriage return
<code>\"</code>	Double Quote	Displays a quotation mark (")
<code>\'</code>	Apostrophe	Displays an apostrophe (')
<code>\?</code>	Question mark	Displays a question mark
<code>\\</code>	Backslash	Displays a backslash (\)
<code>\0</code>	Null	Displays a null character

- other parameters

Symbol	effect	example	
%(d	Encapsulate number with brackets		
%+d	Print positive sign		
%05d	Number length 5 digits with 0 padding to the left		
%,f	Add , between numeric groups (3 numbers)		
%,(f	Encapsulate number with brackets Add , between numeric groups (3 numbers)		
%,.2f	Add , between numeric groups (3 numbers), 2 numbers after decimal point .		

Printfexample05

```
1  +  /*...*/
5  package printfexample05;
6
7  +  /**...*/
11 public class Printfexample05 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17  System.out.printf("Integer formats: %d, %(d, %+d, %05d\n", 3, -3, 3, 3);
18
19  System.out.printf("Default floating-point format: %f\n", 1234567.123);
20  System.out.printf("Floating-point with commas: %,f\n", 1234567.123);
21  System.out.printf("Negative floating-point default: %,f\n", -1234567.123);
22  System.out.printf("Negative floating-point option: %, (f\n", -1234567.123);
23
24  System.out.printf("Line-up positive and negative values:\n");
25  System.out.printf("Signed & unsigned Float: % ,.2f\n% ,.2f\n", 1234567.123, -1234567.123);
26  }
27 }
28
```

```
Output - Printfexample05 (run)
run:
Integer formats: 3, (3), +3, 00003
Default floating-point format: 1234567.123000
Floating-point with commas: 1,234,567.123000
Negative floating-point default: -1,234,567.123000
Negative floating-point option: (1,234,567.123000)
Line-up positive and negative values:
Signed & unsigned Float: 1,234,567.12
-1,234,567.12
BUILD SUCCESSFUL (total time: 0 seconds)
```

EscapeSequence01

```
1  +  /*...*/
5  package escapesequence01;
6
7  +  /**...*/
11 public class EscapeSequence01 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      System.out.printf("\n*****");
18      System.out.printf("\n*\t\t*");
19      System.out.printf("\n*\t\t*");
20      System.out.printf("\n*\t\t*");
21      System.out.printf("\n*\t\t*");
22      System.out.printf("\n*****\n");
23
24
25  }
26 }
27
```

```
Output - EscapeSequence01 (run)
run:
*****
*
*
*
*
*****
BUILD SUCCESSFUL (total time: 0 seconds)
```

Accepting Input from a User

- One of the strengths of Java is the huge libraries of code available to you.
- This is code that has been written to do specific jobs.
- All you need to do is to reference which library you want to use, and then call a method into action.
- One really useful class that handles input from a user is called the Scanner class.
- The Scanner class can be found in the **java.util library**.
- **To use the Scanner class, you need to** reference it in your code. This is done with the keyword **import**.

```
import java.util.Scanner;
```

-
- The next thing you need to do is to create an object from the Scanner class.
 - (A class is just a bunch of code. It doesn't do anything until you create a new object from it.)
 - To create a new Scanner object the code is this:

Scanner user_input = new Scanner(System.in);

- So instead of setting up an int variable or a String variable, we're setting up a Scanner variable.
- We've called ours user_input.
- After an equals sign, we have the keyword new.
- This is used to create new objects from a class.

-
- **The object** we're creating is from the Scanner class.
 - In between round brackets we have to tell java that this will be System Input (**System.in**).
 - To get the user input, you can call into action one of the many methods available to your new Scanner object.
 - One of these methods is called **next**.
 - This gets the next string of text that a user types on the keyboard:

```
String first_name;  
System.out.print("Enter your first name: ");  
first_name = user_input.next( );
```

-
- Notice that we've used `print` rather than `println` like last time.
 - The difference between the two is that `println` will move the cursor to a new line after the output, but `print` stays on the same line.
 - We'll add a prompt for a family name, as well:

```
String family_name;
```

```
System.out.print("Enter your family name: ");
```

```
family_name = user_input.next( );
```

- Then

```
System.out.println("Welcome"+first_name + " " +  
family_name);
```

Scanner01

```
1  +  /*...*/
5  package scanner01;
6  -  import java.util.Scanner;
7  +  /**...*/
11 public class Scanner01 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      String first_name, last_name, full_name;
18      Scanner user_input=new Scanner(System.in);
19      System.out.printf("\nEnter Your First Name:");
20      first_name=user_input.next();
21      System.out.printf("\nEnter Your Family Name:");
22      last_name=user_input.next();
23      full_name=first_name+" "+last_name;
24      System.out.printf("\nYour full name is: "+full_name);
25
26  }
27
28 }
```

```
: Output - Scanner01 (run)
run:
Enter Your First Name:ahmed
Enter Your Family Name:alshafee
Your full name is: ahmed elshafeeBUILD SUCCESSFUL (total time: 8 seconds)
```

Most common "next" input methods.

- `s = sc.next()` Returns next "token", which is more or less a "word".
- `s = sc.nextLine()` Returns an entire input line as a String.
- `i = sc.nextInt()` Returns next integer value.
- `d = sc.nextDouble()` Returns next double value.
- `x = sc.nextXYZ()` Returns value of type XYZ (primitive value if possible), where XYZ is one of `BigDecimal`, `BigInteger`, `Boolean`, `Byte`, `Float`, or `Short`.

Scanner02

```
1 |+ /*...*/
5 package scanner02;
6 - import java.util.Scanner;
7 |+ /**...*/
11 public class Scanner02 {
12 + /**...*/
15 - public static void main(String[] args) {
16     String first_name, last_name, comment;
17     int id,level, gained_credit_hours;
18     float GPA;
19     Scanner s=new Scanner(System.in);
20     System.out.printf("\nEnter Your ID (integer):");
21     id=s.nextInt();
22     System.out.printf("\nEnter Your first name:");
23     first_name=s.next();
24     System.out.printf("\nEnter Your family name:");
25     last_name=s.next();
26     System.out.printf("\nEnter Your Level (1,2,3,4):");
27     level=s.nextInt();
28     System.out.printf("\nEnter Your Gained Credit hours till now (integer):");
29     gained_credit_hours=s.nextInt();
```

```

30     System.out.printf("\nEnter Your your GPA (float):");
31     GPA=s.nextFloat();
32     System.out.printf("\ntype your comments (ended by Enter):");
33     comment=s.next();
34     System.out.printf("\n*****");
35     System.out.printf("\nID\t%d\nname\t%s %s\nLevel\t%d\nCH\t%d\nGPA\t%f",
36         id, first_name, last_name, level, gained_credit_hours, GPA);
37     System.out.println("\nCommnet:\n"+comment);
38 }
39 }
40

```

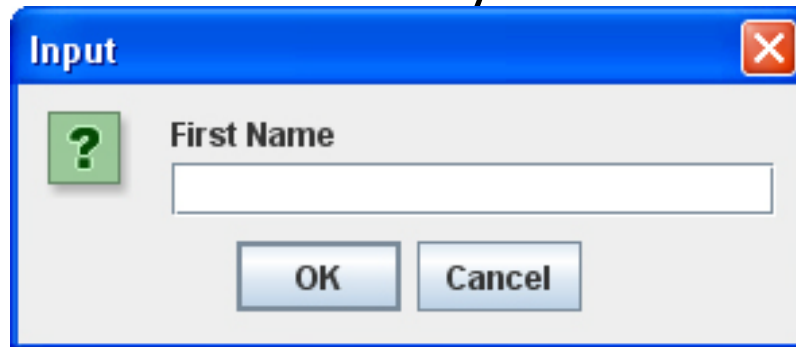
-
- Enter Your ID (integer):32
 - Enter Your first name:de
 - Enter Your family name:rf
 - Enter Your Level(1,2,3,4):1
 - Enter Your Gained Credit hours till now (integer):2
 - Enter Your your GPA (float):2.2
 - type your comments (ended by Enter):rfvergergtergrthytr htryhtyht hyhyth

•*****

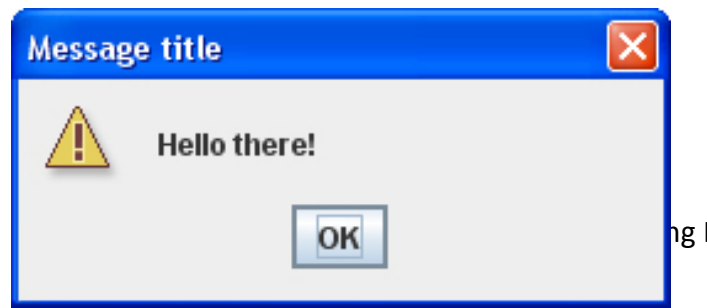
- ID 32
- name de rf
- Level 1
- CH 2
- GPA 2.200000
- Commnet:
- rfvergergtergrthytr
- BUILD SUCCESSFUL (total time: 15 seconds)

Option Panes (GUI)

- Another useful class for accepting user input, and displaying results, is the `JOptionPane` class. This is located in the **javax.swing library**.
- **The `JOptionPane` class** allows you to have input boxes like this one:



- And message boxes like this:



-
- The first thing to do is to reference the library we want to use:

```
import javax.swing.JOptionPane;
```

- This tells java that we want to use the JOptionPane class, located in the **javax.swing library**.
- In the main function type

```
String first_name;
```

```
first_name = JOptionPane.showInputDialog("First Name");
```

```
String family_name;
```

```
family_name = JOptionPane.showInputDialog("Family Name");
```

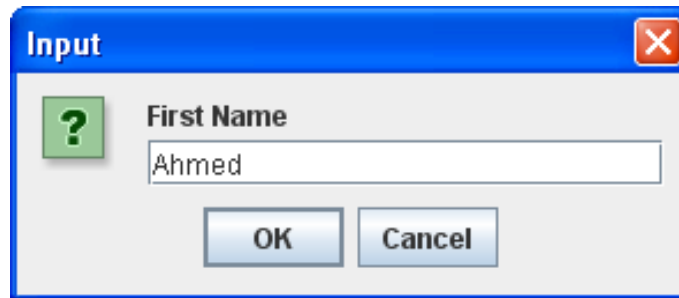
```
String full_name;
```

```
full_name = "You are " + first_name + " " + family_name;
```

-
- To display the result in a message box, add the following:
JOptionPane.showMessageDialog(null, full_name);
 - **Null means** that the message box is not associated with anything else in the program.
 - After a comma comes the text we want to display in the message box.
 - Notice the line at the bottom of the code:
System.exit(0);
 - As its name suggests, this ensures that the program exits.
 - But it also tidies up for us, removing all the created objects from memory.

JOptionPane01

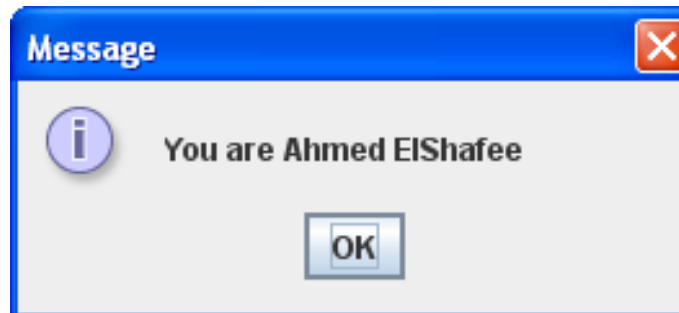
```
1 +  /*...*/
5  package joptionpane01;
6 -  import javax.swing.JOptionPane;
7 +  /**...*/
11 public class JOptionPane01 {
12
13 +  /**...*/
16 -  public static void main(String[] args) {
17     String first_name, last_name, full_name;
18     first_name=JOptionPane.showInputDialog("Your first name?");
19     last_name=JOptionPane.showInputDialog("Your last name?");
20     full_name=first_name+" "+last_name;
21     JOptionPane.showMessageDialog(null,"Welcome "+full_name);
22 }
23 }
24
```



The dialog box has a blue title bar with the text "Input" and a close button (red X) in the top right corner. The main area has a light gray background. On the left, there is a green square icon with a white question mark. To its right, the text "First Name" is displayed above a white text input field containing the text "Ahmed". Below the input field are two buttons: "OK" and "Cancel".



The dialog box has a blue title bar with the text "Input" and a close button (red X) in the top right corner. The main area has a light gray background. On the left, there is a green square icon with a white question mark. To its right, the text "Family Name" is displayed above a white text input field containing the text "ElShafee". Below the input field are two buttons: "OK" and "Cancel".



The dialog box has a blue title bar with the text "Message" and a close button (red X) in the top right corner. The main area has a light gray background. On the left, there is a circular icon with a lowercase letter 'i'. To its right, the text "You are Ahmed ElShafee" is displayed. Below the text is a single button labeled "OK".

JOptionPane02

- Input boxes and Message boxes can be formatted further. Try the following for
- your Input boxes:

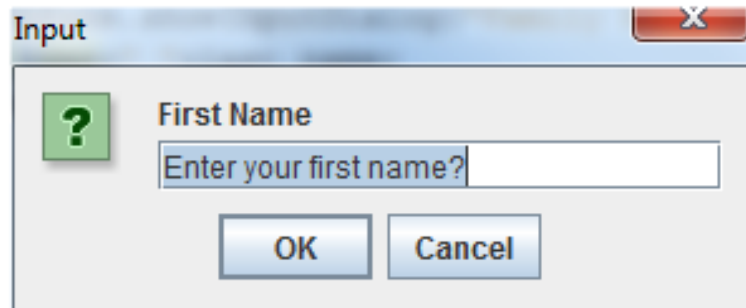
showInputDialog("First Name", "Enter Your First Name");

showInputDialog("Family", "Enter Your Family Name");

```

1  +  /*...*/
5  package joptionpane02;
6  -  import javax.swing.JOptionPane;
7  +  /**...*/
11 public class Joptionpane02 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      String first_name, last_name, full_name;
18      first_name=JOptionPane.showInputDialog("First Name","Enter your first name?");
19      last_name=JOptionPane.showInputDialog("Family Name","Enter your last name?");
20      full_name=first_name+" "+last_name;
21      JOptionPane.showMessageDialog(null,"Welcome "+full_name);
22
23  }
24 }
25

```



Input

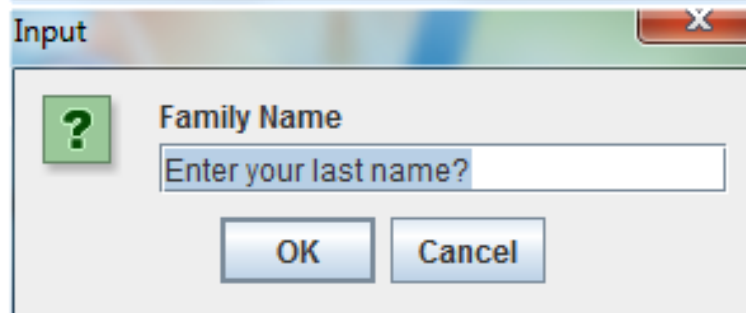
?

First Name

Enter your first name?

OK Cancel

This is a standard Windows-style input dialog box. It has a title bar with the text 'Input' and a close button (X). The main area contains a green question mark icon, the label 'First Name', and a text input field with the placeholder text 'Enter your first name?'. Below the input field are two buttons: 'OK' and 'Cancel'.



Input

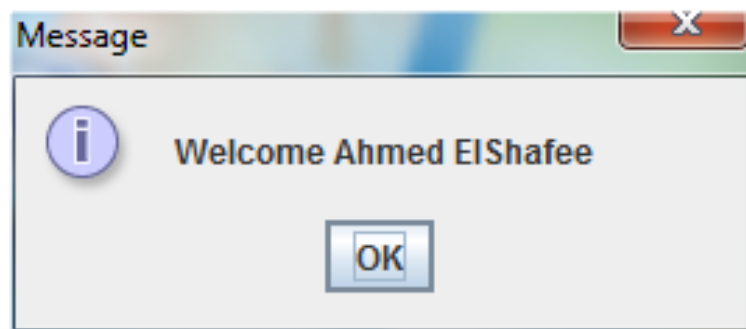
?

Family Name

Enter your last name?

OK Cancel

This is a standard Windows-style input dialog box. It has a title bar with the text 'Input' and a close button (X). The main area contains a green question mark icon, the label 'Family Name', and a text input field with the placeholder text 'Enter your last name?'. Below the input field are two buttons: 'OK' and 'Cancel'.



Message

i

Welcome Ahmed ElShafee

OK

This is a standard Windows-style message dialog box. It has a title bar with the text 'Message' and a close button (X). The main area contains a blue information icon (i), the text 'Welcome Ahmed ElShafee', and a single 'OK' button at the bottom.

JOptionPane03

- For your Message boxes try this (yours should be on one line):
**showMessageDialog(null, full_name, "Name",
JOptionPane.INFORMATION_MESSAGE);**

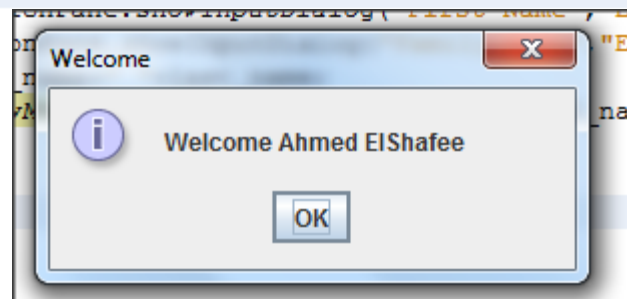
Exercise

- Instead of JOptionPane.INFORMATION_MESSAGE try these:
- **ERROR_MESSAGE**
- **PLAIN_MESSAGE**
- **QUESTION_MESSAGE**
- **WARNING_MESSAGE**

```

1  +  /*...*/
5  package joptionpane03;
6  -  import javax.swing.JOptionPane;
7  +  /**...*/
11 public class JOptionPane03 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      String first_name, last_name, full_name;
18      first_name=JOptionPane.showInputDialog("First Name","Enter your first name?");
19      last_name=JOptionPane.showInputDialog("Family Name","Enter your last name?");
20      full_name=first_name+" "+last_name;
21      JOptionPane.showMessageDialog(null, "Welcome "+full_name,"Welcome", JOptionPane.INFORMATION_MESSAGE);
22  }
23 }
24

```



JOptionPane04

Exercise

- Input boxes are not just used for text: they can accept numbers as well.
- Write a program that prompts the user for two numbers, the breadth of a rectangle and the height of a rectangle.
- Use a message box to calculate the area of the rectangle. (Remember: the area of a rectangle is its breadth multiplied by the height.)
- However, you'll need some extra help for this exercise.

Help for the Exercise

- You have to use the String variable to get your numbers from the user:

String breadth;

breadth = JOptionPane.showInputDialog("Rectangle Breadth");

- However, you can't multiply two strings together.
- You need to convert the Strings to integers.
- You can convert a string to an integer like this:

Integer.parseInt(string_to_convert)

-
- You can then multiply and assign on the same line;
int area = Integer.parseInt(string_one) * Integer.parseInt(string_two);
 - For the message box, use concatenation:
“answer = ” + area
 - You can use any of the MESSAGE symbols for your message

```

1  +  /*...*/
5  package joptionpane04;
6  import javax.swing.JOptionPane;
7  +  /**...*/
11 public class Joptionpane04 {
12
13  +  /**...*/
16  -  public static void main(String[] args) {
17      String user_input,output;
18      int breadth, hight, area;
19      user_input=JOptionPane.showInputDialog("Breadth","0");
20      breadth=Integer.parseInt(user_input);
21      user_input=JOptionPane.showInputDialog("Hight","0");
22      hight=Integer.parseInt(user_input);
23      area=breadth*hight;
24      JOptionPane.showMessageDialog(null, "Rectangle Area = "+area,"Rectangle", JOptionPane.INFORMATION_MESSAGE);
25  }
26  }
27

```

Exercise1

- The program will crash if you enter floating point values for the breadth and height.
- How would you solve this?
- When you have solved the above exercise, do you really want Integer.parseInt?
- What else do you think you can use?
- Search for solution in internet,...

JOptionPane05



```
1 | +| /*...*/
5 | package joptionpane05;
6 | -| import javax.swing.JOptionPane;
7 | +| /*...*/
11 | public class JOptionPane05 {
12 |
13 | +|     /*...*/
16 | -|     public static void main(String[] args) {
17 |         String user_input;
18 |         float breadth, hight, area;
19 |         user_input=JOptionPane.showInputDialog("Breadth","0");
20 |         breadth=Float.parseFloat(user_input);
21 |         user_input=JOptionPane.showInputDialog("Hight","0");
22 |         hight=Float.parseFloat(user_input);
23 |         area=breadth*hight;
24 |         JOptionPane.showMessageDialog(null, "Rectangle Area = "+area,"Rectangle", JOptionPane.INFORMATION_MESSAGE);
25 |
26 |     }
```

Exercise2

Rebuild the following program using JoptionPane instead of scanner

```
1  +| /*...*/
5  package scanner02;
6  -| import java.util.Scanner;
7  +| /*...*/
11 public class Scanner02 {
12 +|   /*...*/
15 -|   public static void main(String[] args) {
16       String first_name, last_name, comment;
17       int id,level, gained_credit_hours;
18       float GPA;
19       Scanner s=new Scanner(System.in);
20       System.out.printf("\nEnter Your ID (integer):");
21       id=s.nextInt();
22       System.out.printf("\nEnter Your first name:");
23       first_name=s.next();
24       System.out.printf("\nEnter Your family name:");
25       last_name=s.next();
26       System.out.printf("\nEnter Your Level(1,2,3,4):");
27       level=s.nextInt();
28       System.out.printf("\nEnter Your Gained Credit hours till now (integer):");
29       gained_credit_hours=s.nextInt();
30       System.out.printf("\nEnter Your your GPA (float):");
31       GPA=s.nextFloat();
32       System.out.printf("\ntype your comments (ended by Enter):");
33       comment=s.next();
34       System.out.printf("\n*****");
35       System.out.printf("\nID\t%d\nname\t%s %s\nLevel\t%d\nCH\t%d\nGPA\t%f",
36           id, first_name, last_name, level, gained_credit_hours, GPA);
37       System.out.println("\nCommnet:\n"+comment);
38   }
39 }
40
```

ing l,

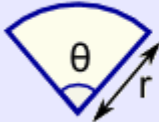
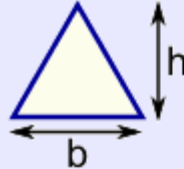

JOptionPane06

Assignments

1. Create a program accept float radius from the user then calculates the area of circle.
 - A. Use scanner class
 - B. Use JOptionPane class
2. Create a program that accept float triangle base and high than calculate its are
 - A. Use scanner class
 - B. Use JOptionPane class

3. Create a software accepts float radius and integer angle in degrees, than calculate sector area.

- A. Use scanner class
- B. Use JOptionPane class

	<p><u>Sector</u> Area = $\frac{1}{2}r^2\theta$ r = radius θ = angle in radians</p>
	<p><u>Triangle</u> Area = $\frac{1}{2}b \times h$ b = base h = vertical height</p>
	<p><u>Circle</u> Area = πr^2 Circumference = $2\pi r$ r = radius</p>



Thanks,
See you next Lecture, isA

