

CMSC 131: Chapter 2 (Supplement)

Overview of Java

The Java Programming Language

Java - Why is it special?

- An **object-oriented** programming language. (More about this later.)
 - Developed in early 1990's by James Gosling at Sun Microsystems for controlling programmable devices.
 - Unlike C++, Java programs are **portable** between machines.
-
- Has a rich set of **libraries** that allow you to create graphics, communicate over networks, interact with databases, etc.
 - The Java compiler and interpreter are part of **Java Software Development Kit (SDK)**, also referred to as **Java Development Kit (JDK)**.

Java Programming Concepts

A quick overview of fundamental Java concepts.

Object:

Class:

Method:

Main Method:

Statements:

- Declarations** -
- Assignment** -
- Method invocation** -
- Control flow** -

A Sample Java Program

```
/**
 * My first sample Java program.
 */
public class MyFirstProgram {

    public static void main( String[ ] args ) {
        int secondsPerMinute = 60;           // let's create some variables
        int minutesPerLecture = 50;
        int totalSeconds = secondsPerMinute * minutesPerLecture;
        System.out.println( "There are " + totalSeconds + " seconds in a lecture." );
    }

}
```

Java Program Organization

Class:

File name: `MyFirstProgram.java`:

```
public class MyFirstProgram {
    ... (contents of the class go here) ...
}
```

Java Program Organization

Methods:

```
public static void main( String[ ] args ) {
    ... (contents of the main method go here) ...
}
```

Variables:

```
int secondsPerMinute = 60;
int minutesPerLecture = 50;
```

Java Program Organization

Statements:

Declarations -

Assignments -

Method invocation -

Control flow -

Expressions/Operators:

Other Java Elements: Comments

Comments:

Line comments: After `//` the rest of the line is a comment.

Block comments: The text between any pair: `/* ... */`.

Java I/O and JOptionPane

Java Input/Output:

JOptionPane: A graphical mechanism for text input and output.

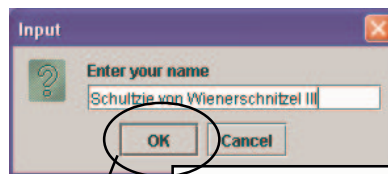
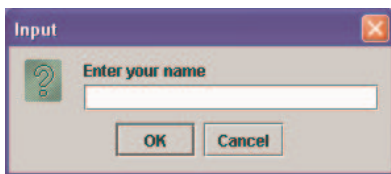
Input dialog:

`showInputDialog( String prompt )`: Prints the given prompt, and returns the string entered by the user.

Example:

String name;

name = JOptionPane.showInputDialog("Enter your name");



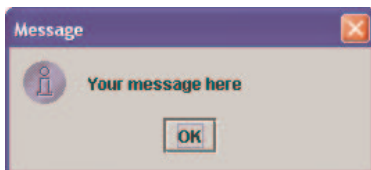
name ← "Schultzie von Wiener schnitzel III"

More JOptionPane

Message Dialog: A method for outputting a message.

The `showMessageDialog( )` method generates a message box.

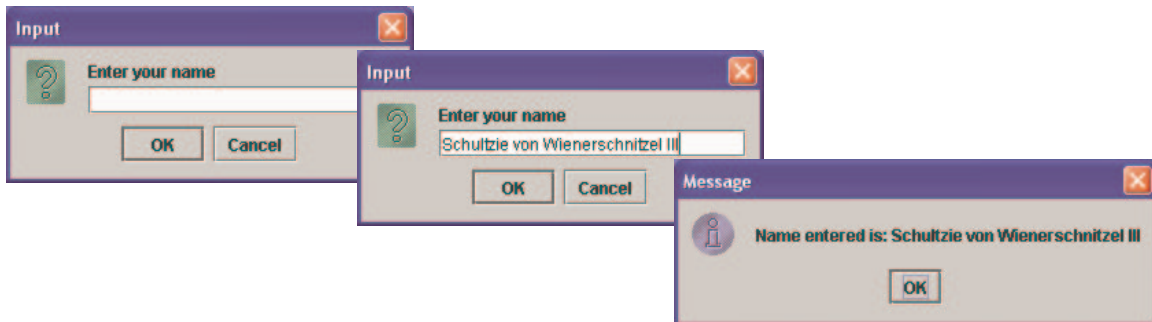
`JOptionPane.showMessageDialog( null, "Your message here" );`



Java Example with JOptionPane

```
/**
 * A simple example showing Java I/O using JOptionPane
 */
import javax.swing.*;           // needed to make JOptionPane accessible

public class JOPSample {
    public static void main( String[ ] args ) {
        String name = JOptionPane.showInputDialog( "Enter your name" );
        String reply = "Name entered is: " + name;
        JOptionPane.showMessageDialog( null, reply );
        System.exit(0);         // terminate the program
    }
}
```

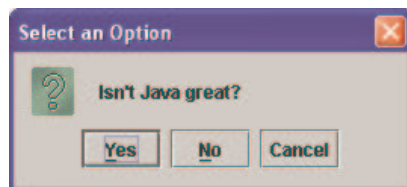


Still More on JOptionPane

Confirmation Dialog: For user confirmation (yes/no/cancel).

The `showConfirmDialog()` method generates a message box:

```
int answer;
answer = JOptionPane.showConfirmDialog( null, "Isn't Java great?" );
```



Execution resumes after the user makes a selection. The method returns one of three results (stored in variable "answer"):

- `JOptionPane.YES_OPTION` if "Yes"
- `JOptionPane.NO_OPTION` if "No"
- `JOptionPane.CANCEL_OPTION` if "Cancel"