COMP 110-003
Introduction to Programming
Console I/O, Java GUI

January 24, 2013

Haohan Li
TR 11:00 – 12:15, SN 011
Spring 2013

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

Daily Joke

- A guy is standing on the corner of the street smoking one cigarette after another. A lady walking by notices him and says, "Hey, don't you know that those things can kill you? I mean, didn't you see the giant warning on the box?!“

  "That's OK" says the guy, puffing casually, "I'm a computer programmer“

  "So? What's that got to do with anything?“

  "We don't care about warnings. We only care about errors."

Today

- More about screen output and keyboard input
- Introduction to Java Swing
  - Provides a way to use windowing for I/O in your Java programs

Screen Output

- We've seen several examples of screen output already
  - *System.out.println("Hello World!");*
  - *System.out.println("The input integer is "+n1);*
- *System.out* is an object that sends the output to the screen; *println()* is the method that performs this action for the *System.out* object.

Screen Output

- The concatenation operator (+) is useful when you want to output both texts and values of variables to the screen.
  - *int lucky_num = 13, secret_num = 11;*
  - *System.out.println("The lucky number is " + i + ", and the secret number is " + secret_num);*
- Do not break the line except immediately before or after the concatenation operator (+).

Screen Output

- Alternatively, you can use *print()* in multiple times and end with a *println()*;
  - *System.out.print("The lucky number is ");*
  - *System.out.print(i);*
  - *System.out.print(", and the secret number is ");*
  - *System.out.println(secret_num);*

Keyboard Input by Scanner Class

- Near the beginning of your program, insert
  - *import java.util.Scanner;*
- Create an object of the Scanner class
  - *Scanner keyboard = New Scanner (System.in);*
- Read data (an int or a double, for example)
  - *int n1 = keyboard.nextInt();*
  - *double d1 = keyboard,nextDouble();*

More Scanner Class Methods

FIGURE 2.7 Some Methods in the Class Scanner

Scanner_Object_Name`.next()`
Returns the `String` value consisting of the next keyboard characters up to, but not including, the first delimiter character. The default delimiters are whitespace characters.

Scanner_Object_Name`.nextLine()`
Reads the rest of the current keyboard input line and returns the characters read as a value of type `String`. Note that the line terminator `'\n'` is read and discarded; it is not included in the string returned.

Scanner_Object_Name`.nextInt()`
Returns the next keyboard input as a value of type `int`.

Scanner_Object_Name`.nextDouble()`
Returns the next keyboard input as a value of type `double`.

nextLine() Method

- The *nextLine()* method reads
  - The remainder of the current line,
  - Even if it is empty.
- Make sure to read **Gotcha** before Figure 2.7

Empty String

- A string can have any number of characters, including zero.
  - The string with zero characters is called the empty string.
  - Why is it useful?
    - Consider what happens if you want to read a line and the user input nothing but a return
    - Consider what happens if you want to find a common substring between “aaa” and “bbb”
- The empty string can be created in many ways
  - *String empty_string = “”;*

Java Swing

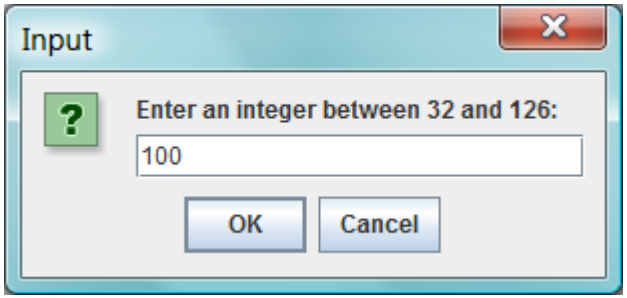
- Java makes it easy to build Graphical User Interfaces (GUIs)
  - *javax.swing* package
  - Import *javax.swing.** into your program
    - *swing.** means all classes under the swing package
- Read Sections 1.4 and 2.5 for more info (Graphics Supplement)

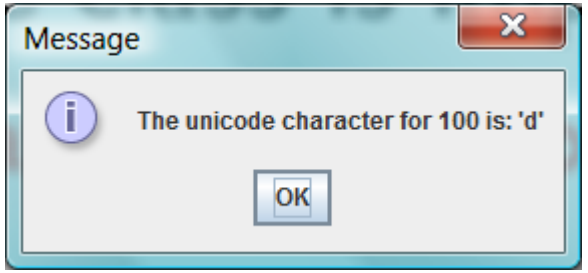
JOptionPane Class

- Our focus today: the *JOptionPane* class
  - *javax.swing.JOptionPane*
  - You will be using *JOptionPane* in your lab today
- Import *JOptionPane* class by either statement
  - *import javax.swing.JOptionPane;*
  - *import javax.swing.*;*

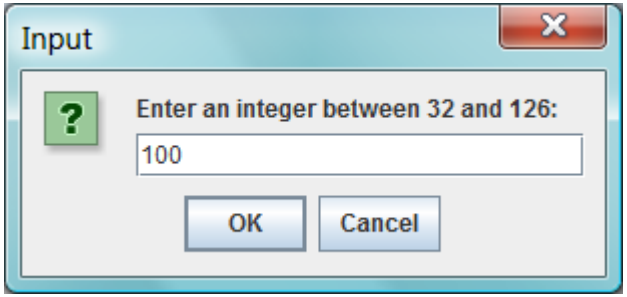
JOptionPane Class

- *JOptionPane* can be used to construct windows that interact with the user
  - The *JOptionPane* class produces windows for obtaining input or displaying output.





Get Screen Input in *JOptionPane*



- Use *showInputDialog()* for input .
- Only **string values** can be input.
- To convert an input value from a string to an integer use the *parseInt()* method from the *Integer* class
  - ***inputInt= Integer.parseInt(outputString);***
  - Integer is a default class in the system, too. Like this:
  - *System.out.println("Output");*
  - *parseInt()* is the standard way to cast a *String* to an *int*
    - Guess how to cast a *String* to a *double*?

Inputting Numeric Types from Strings

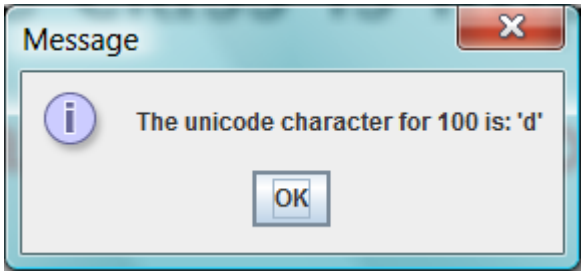
- Figure 2.9 shows methods for converting strings to numbers

FIGURE 2.9 Methods for Converting Strings to Numbers

Result Type	Method for Converting
`byte`	`Byte.parseByte(`*String_To_Convert*`)`
`short`	`Short.parseShort(`*String_To_Convert*`)`
`int`	`Integer.parseInt(`*String_To_Convert*`)`
`long`	`Long.parseLong(`*String_To_Convert*`)`
`float`	`Float.parseFloat(`*String_To_Convert*`)`
`double`	`Double.parseDouble(`*String_To_Convert*`)`

Output to Screen in *JOptionPane*

- Output is displayed using the *showMessageDialog* method.
  - The syntax rule is:
    - *JOptionPane.showMessageDialog(**null**, "Some String");*
  - *JOptionPane.showMessageDialog(**null**, **"The unicode character for "**+ inputInt + **" is: \'"** + (**char**) **inputInt + "\'");***



char vs *int*

- Character is saved as numbers in memory
  - '0' <-> 48, '1' <-> 49, '2' <-> 50, …, '9' <-> 57
  - 'A' <-> 65, 'B' <-> 66, 'C' <-> 67, …, 'Z' <-> 90
  - 'a' <-> 97, 'b' <-> 98, 'c' <-> 99, …, 'z' <-> 122
  - http://www.cs.cmu.edu/~pattis/15-1XX/common/handouts/ascii.html

Syntax Rules

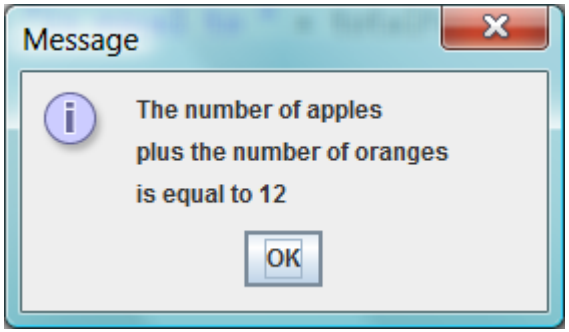
- Input
  - *String_Variable = JOptionPane.showInputDialogue (String_Expression);*
- Output
  - *JOptionPane.showMessageDialog (null, String_Expression);*
- *System.exit(0)* ends the program.

JOptionPane Cautions

- If the input is not in the correct format, the program will *crash.* (e.g asking for Integer, but input is Character)
- The output must be a string.
  - Read *GOTCHA: Displaying Only a Number*
- If you omit the last line "*System.exit(0);*", the program will not end, even when the **OK** button in the output window is clicked.

Multi-Line Output

- To output multiple lines using the method *JOptionPane.showMessageDialog*, insert the new line character '\n' into the string expression
  - *int totalFruit = 12;*
  - *JOptionPane.showMessageDialog(null,*
    "The number of apples\n" +
    "plus the number of oranges\n" +
    "is equal to " + totalFruit);



Assignments without Submission

- Run and manipulate these codes in eclipse:
  - Jan22.java
  - StringsAndChars.java
  - TypeCasting.java
- Finish this doc and bring it to class next Tuesday
  - Chapters 1 and 2 Review Worksheet
  - You can print it or type answer in computer
- You don't have to submit them, but these materials are close to exams

Lab 2

- Implement requirements, and show them to me
  - There is no specific deadline
  - But you will submit Lab 3 in 2 weeks, and Lab 3 is based on Lab 2
- Today you can work on Lab 1, Prog 1 or Lab 2
  - You are not allowed to discuss details in Prog 1 and Lab 2