



CMSC 131

Object-Oriented Programming I

String Comparison, Scanner

**Dept of Computer Science
University of Maryland College Park**

This material is based on material provided by Ben Bederson, Bonnie Dorr,
Fawzi Emad, David Mount, Jan Plane

Overview

- String Comparison
- Scanner
- Programming errors
- Debugging
- Conditionals
- Logical Operators

To Import Class Examples

1. Do not extract the zip file
2. In Eclipse select File→Import...→General(Expand by clicking on triangle)→Existing Projects into Workspace→Next
NOTICE that "Archive File" is NOT the option you want
3. Select the option "Select archive file" and browse for the zip file
4. Select Open and Finish

String Comparison

- **String Comparison:** Strings should not be compared using the above operators (`==`, `<=`, `<`, etc). Let **s** and **t** be strings.
 - s.equals(t)** → returns true if s equals t
 - s.length()** → returns length
 - s.compareTo(t)** → compares strings **lexicographically** (dictionary order)
 - result < 0 if s is less than t
 - result == 0 if s is equal to t
 - result > 0 if s is greater than t
- **Example:** StringComparison1.java
- **Important:**
 - In some cases you may get the expected results when comparing strings using `==`
 - Do not use `==` to compare strings. Use `equals` or `compareTo`
- **Example:** StringComparison2.java

User Input in Java

- We've done output (System.out); what about input?
- Java includes the **Scanner class** feature
 - Can use Scanner to create “scanner objects”
 - Scanner objects convert user input into data
- To use Scanner need to *import* a library:
 - `import java.util.Scanner;`

Scanner Class Details

- To create a scanner object:
- **new** Scanner(*input_source*);
 - Input source can be keyboard (System.in), files, etc.
 - Object must be assigned to a variable (e.g. sc)
- Operations
 - nextBoolean()
 - nextByte()
 - nextDouble()
 - nextFloat()
 - nextInt()
 - nextLong()
 - nextShort()

Returns value of indicated type
(reports error if type mismatch)
- next() Returns sequence of characters up to next whitespace
(space, carriage return, tab, etc.)
- nextLine() Returns sequence of characters up to next carriage return
- **Example:** Scanner1.java, Scanner2.java

Objects

- About Scanner
 - `Scanner` is a class defined in `java.util.Scanner`
 - `System.in` is a predefined *object* for keyboard input
 - **`sc = new Scanner(System.in)`** creates a new *object* in the `Scanner` class and assigns it to `sc`
- Object?
 - A bundle of data (*instance variables*) and operations (*methods*)
 - A class defines both instance variables and methods for objects
 - A class is also a type for objects
 - **`new`** creates new objects in the given class
- We will learn (much) more about objects later

Programming Errors

- Syntax error
 - Violates language's grammar
 - Compiler will catch them
 - Eclipse makes red squiggles under your code
- Run-time errors (exceptions in Java)
 - Something unexpected happens during program execution (e.g., dividing by 0, file not found)
- Semantical/logical errors
 - Program does not generate expected results
 - Can be challenging to uncover

Debugging

- Process of finding and fixing problems
- Important rule while writing programs and to avoid debugging:
Incremental Code Development
- Incremental code development (write a little bit, test thoroughly and continue)
- Suggestions about writing computer programs can be found at:
<http://www.cs.umd.edu/~nelson/documents/SuggestionsForWritingComputerPrograms.htm>