Standard Java Libraries
'static' Keyword

22 Jan 2010
CMPT166
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Quiz 1: 10min, 20pts

- Contrast the JRE with the JDK. [4]
- Name the 8 primitive types built-in to Java, and contrast them (what's the difference?) [8]
- Write a Java function numHashes() that takes an integer parameter and prints that many hash marks ("#") to the screen: [8]
  - **numHashes(5); // outputs: "#####"**
  - Include a docstring and pre/post-conditions.

Quiz 1: answers #1-2

- Contrast the JRE with the JDK. [4]
  - JRE: Java Runtime Environment: enough to run precompiled Java applications (java)
  - JDK: Java Development Kit: JRE + enough to write+compile your own (javac+java)
- Name the 8 primitive types built-in to Java, and contrast them (what's the difference?) [8]
  - boolean (true/false), char (Unicode)
  - byte (1), short (2), int (4), long (8)
  - float (4), double (8)

Quiz 1: answers #3

- Write a Java function numHashes() that takes an integer parameter and prints that many hash marks ("#") to the screen: **[8]**

```
/** Print given number of hashes.
  * @param num Number of hashes to print.
  * @return Doesn't return anything (prints). */
public void numHashes( int num ) {
        for (int I=0; i++; i<num) {
            System.out.print("#");
        }
        System.out.println();
}
```

Some handy Math methods

- Class methods in Math module
  - sqrt(x)
  - abs(x)
  - max(x, y), min(x, y)
  - ceil(x), floor(x)
  - cos(x), sin(x), etc.
  - exp(x), log(x) (natural log)
  - pow(x, y) (y can be a float)
  - random() (double in range [0, 1))

Some handy standard packages

- java.lang: automatically imported
- java.io: files and streams
- java.net: networking
- java.text: manipulate strings, dates, i8n
- java.util: miscellaneous utilities: strings, etc.

- java.applet: or javax.swing.JApplet for Swing
- java.awt: or javax.swing
- java.awt.event: or javax.swing.event

static keyword

- public static void main(String args[]) {
- static keyword: class attribute
  - Shared by all instances of this class
  - vs. instance attribute: separate for each object
  - Exists before class is instantiated
  - Invoke class methods as: ClassName.method()
- Running a class vs. instantiating a class:
  - Run a class from JRE: java MyClass
  - No instances made, just MyClass.main() invoked
  - Instantiating: new MyClass()
  - Constructor is run, main() is not run

static import

- import static java.lang.Math.*;
- Import all static members of a class
- Brings static variables/methods into current namespace:
  - sqrt(36.0); instead of Math.sqrt(36.0);
  - log(E); instead of Math.log(Math.E);
- Can also bring in one particular member:
  - import static java.lang.Math.sqrt;

Scope and duration

- The duration (lifetime) of an identifier is the runtime period when it exists in memory
  - Automatic duration
    - Local variables disappear when block finishes
  - Static duration
    - As long as the object/module/program exists
- The scope of an identifier is the lexical extent where it can be referenced
  - Block scope
  - Class scope

Scope example

```
public class ScopeExample {
    int numApples = 0;          // class scope
    public void listApples() {
        int counter = 0;        // block scope
    }
}
```

- numApples is an instance variable with class scope: accessible to all methods of this class
- counter is a local variable with block scope: not accessible outside the listApples() method