Object-Oriented Programming II
Java Fundamentals

Mats Swan

mats.swan@sheridancollege.ca

Winter 2014

Sheridan

About the class…

- Class plan
- Calendar
- Expectations
- Contact info
- Conduct

This week

- Getting to know each other
- Understanding the journey ahead of us this term
  - What is the course about?
  - How are you going to be evaluated?
  - What materials and study methods do we use?
  - How can I be successful in this class?
- Java fundamentals
- Professional software development tools

What is this course about?

- Course outline
- What would you like this course to be about?
  - What can I do with Java?
- Starting off…
  - OOP 1
  - Thinking in objects!!
- In the meantime…
  - IDEs, professional development, collections, GUI development, I/O programming, etc.

Text book

- Required
  - "Introduction to Java Programming" Y. Daniel Liang, 7th edition
- Recommended
  - "Head First Java" by Kathy Sierra and Bert Bates, 2nd edition
- 24 x 7 books
  - A better source than Google!

Slate2 content and usage

- Slate2 content
  - Course content
  - Links
  - Contact info
  - Announcements
  - Grades
  - Dropbox
- Watch for announcements and news
- Grades will be available by next class*

*no promises

What to expect

- Theory and hands-on
  - Slides and discussions to go over theory
  - Exercises, assignments, labs to practice
- Exercises (important!)
  - Lots of lab exercises you may need to finish at home and deliver the following week
- Labs and assignments (*very* important!)
  - Building blocks for The Project
- The Project (*super* important!)
  - Due at the end of term
  - Applies everything you learned in class… with GUI!

Exams

- Approximately 2.5 hours
- Knowledge component
  - Closed book
  - Multiple choice, fill-in-the-blanks, paragraph style
- Practical component
  - Open book
  - Develop, debug, test, and produce a working program
- Midterm: Week 7 (week of Feb 17)
- Final: Week 14 (week of April 14)

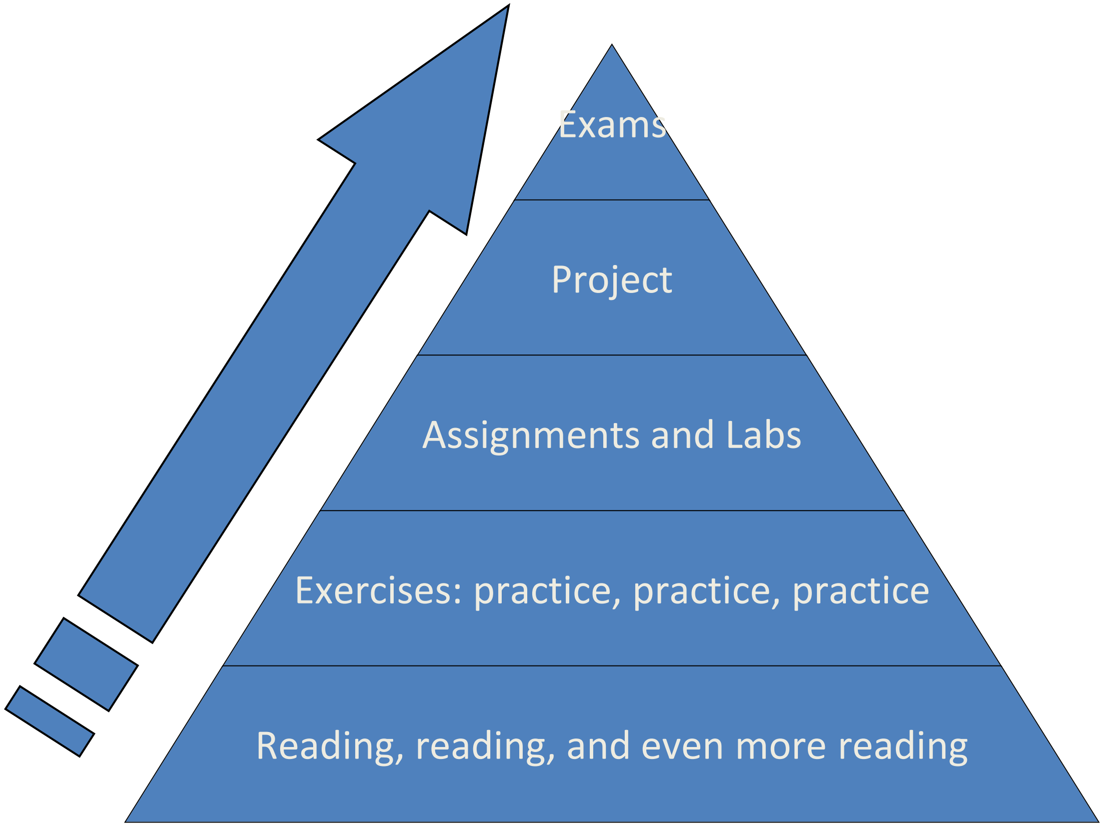
Evaluation

- Weighting

– Quizzes and Exercises	10%
– Assignments and Labs	15%
– Final Project	10%
– Midterm	30%
– Final Exam	35%

- To pass, you must:
  - Average 50% or more on the entire course, AND
  - Average 50% or more on the written exams (midterm and final)

Studying for this course



Exams

Project

Assignments and Labs

Exercises: practice, practice, practice

Reading, reading, and even more reading

While in class, please…

- Come prepared by reading the required material
- Be on time; if you start behind, you'll be behind. And you may miss time on a quiz.
- Take notes. If it's on the whiteboard, chances are, it won't be on Slate2
- Complete the exercises. If not, then at home and drop them in the Dropbox
- Pay attention
- Ask questions DURING class; everyone benefits

…and please do not

- Spend the class playing games, IMing, e-mailing, texting, tweeting, updating your status, chatting, or browsing the Internet.
- Work on tasks outside the scope of this class, such as other assignments due in other classes
- Disrupt your fellow students.

NetBeans

Introduction, installation, and overview

Let’s go!

1. Do you have JDK 7 installed yet?
   - No? Then what are you waiting for?
   - http://www.oracle.com/technetwork/java/javase/downloads/index.html
   - Yes? Congratulations, move on to Step 2
2. Download and install NetBeans
   - http://netbeans.org
   - Install the FULL 204MB version of NetBeans!

- Did you know?
  - You can install JDK 7 and NetBeans together at oracle.com

The IDE

- What is an IDE?
- How important is an IDE?

Some Java Platform IDEs

- NetBeans
  - Developed by a private company, later bought by Sun Microsystems
  - Open source by Sun Microsystems
  - Currently an open-source IDE
- Eclipse
  - Open source project originally developed by IBM
  - Very extensible, giving it the "platform" flavour.
- Oracle JDeveloper
  - Proprietary, but free
- Lots and lots of others

NetBeans (v7.4)

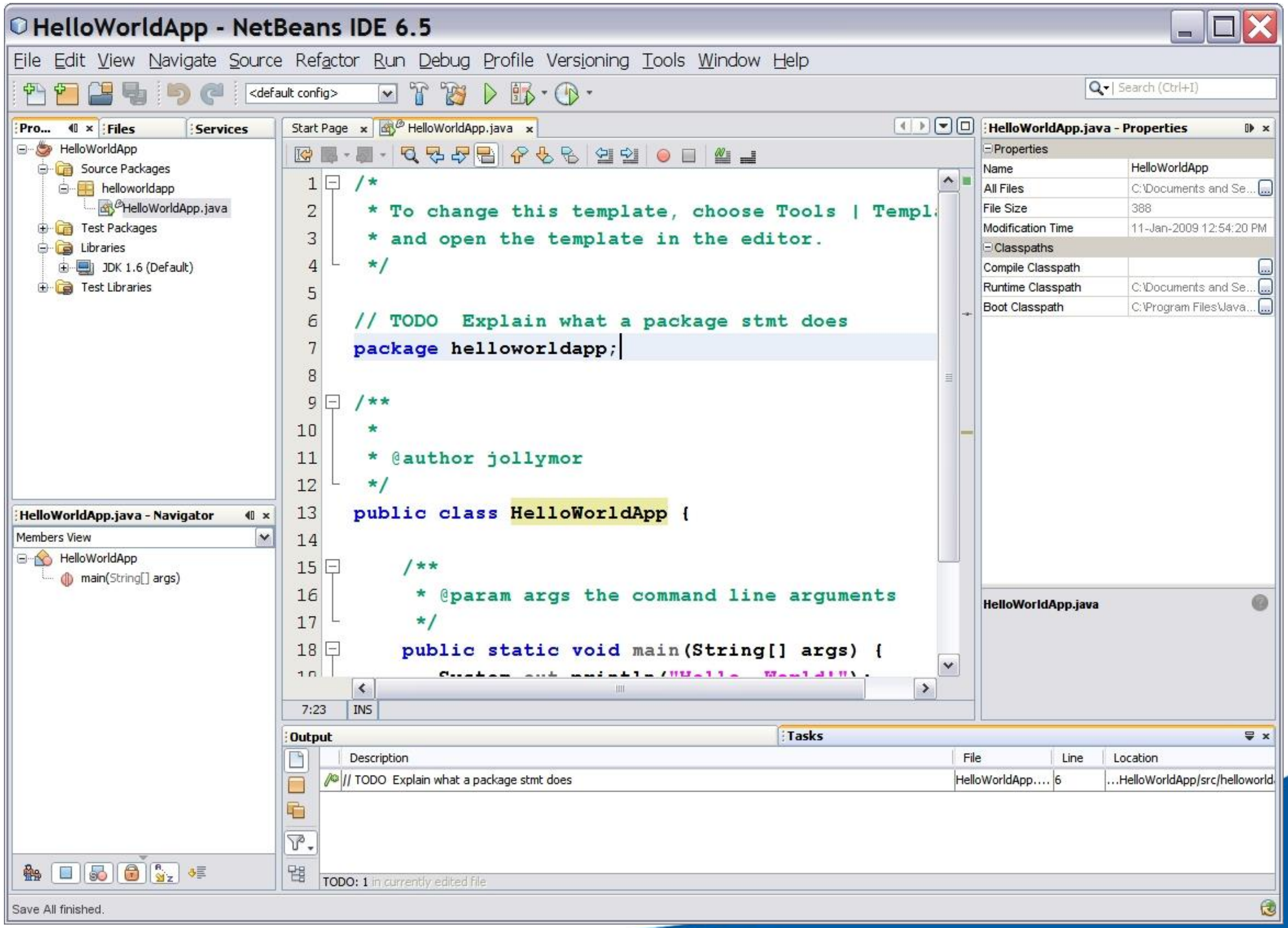
- Integrated Development Environment (IDE)
  - Runs on JDK 7
- Organizing large applications
  - Management of Java projects with project templates
  - Code assistance improves productivity
- Things the IDE does for you:
  - Refactor, Intellisense (“Content assist”)
  - GUI programming
  - Testing and Debugging
  - Code management
  - Deployment, HTML5, PHP, C++, Groovy, etc.
  - Apache Tomcat and Glassfish web servers

How is Eclipse different?

- Eclipse is a platform, designed for building more IDEs
  - Eclipse Software Development Kit (Eclipse SDK)
  - Eclipse SDK + Java Development Tools (JDT) -> Java IDE
  - Eclipse SDK + C++ Development Tools (CDT) -> C++ IDE
- Not just about using programming languages
  - Banking
  - Automotive
  - Medical
  - Space exploration
- Provides integration points ("plug-ins") for developers and users

NetBeans IDE

- Projects
- Files
- Services
- Navigator
- Source editor
- Output
  - ToDos
- Properties



```
/*
 * To change this template, choose Tools | Templ
 * and open the template in the editor.
 */

// TODO  Explain what a package stmt does
package helloworldapp;

/**
 *
 * @author jollymor
 */
public class HelloWorldApp {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        System.out.println("Hello, World!");
```

HelloWorldApp.java - Properties

Properties	
Name	HelloWorldApp
All Files	C:\Documents and Se...
File Size	388
Modification Time	11-Jan-2009 12:54:20 PM
Classpaths	
Compile Classpath	
Runtime Classpath	C:\Documents and Se...
Boot Classpath	C:\Program Files\Java...

HelloWorldApp.java

Description	File	Line	Location
// TODO Explain what a package stmt does	HelloWorldApp....	6	...HelloWorldApp/src/helloworld

TODO: 1 in currently edited file

Exercises

1. Build a “Hello World” application with NetBeans
   - Have a user enter her/his name. E.g. “Hello, Mats”
2. Build a program that spells a phrase backwards
   - E.g., “Mary had a little lamb” -> “bmal elttil a dah yraM”

Making NetBeans your own

- Customization
  - Code Folding (Editor > General)
  - Tabs (Editor > Formatting)
  - Java code formatting (Editor > Formatting)
  - Code completion (Editor > Code completion)
  - Code templates (Editor > Code templates)
  - Syntax colours (Fonts and Colours > Syntax)

Other cool things in NetBeans

- Real estate: location, location, location
- Code formatting: look like a pro… even if you aren’t
- Refactoring: look like a *real* pro
- Templates: developers really are lazy
- In-code navigation: at your fingertips
- Debugging
  - Identify errors
  - Set breakpoints
  - Watch variables

Packages

What are they and why do I need them?

What is a package?

- Organizes classes
- Physical and logical organization
- How packages relate to directory structure

Defining a package

- Simple!

```
package com.mes.package;

public class SampleClass {
   ...
```

What you need to know

- Packages are always the first noncomment, nonblank statement in a program
- Packages (logical) correspond to directories (physical)
- Packages contain both compiled and uncompiled code (.class and .java)
- Packages are hierarchical

Why do I need packages?

- To locate classes
- To avoid naming conflicts
- To distribute software conveniently
- To protect classes
  - remember ‘protected’?

Using packages

- The **import** statement
- Import a single class from a package

```
import com.wsj.SampleClass;
```

- Import all the classes from a package

```
import com.wsj.*;
```

- Secret: you’ve always been importing the java.lang.* package, it’s automatically done!