Introduction

Object Orientated Programming in Java

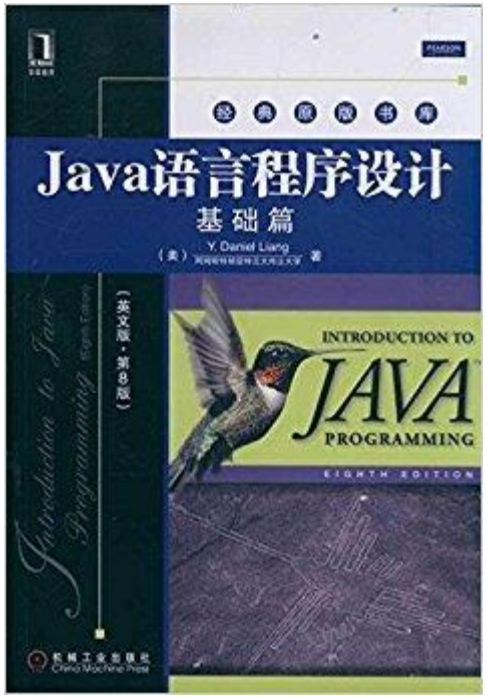
Benjamin Kenwright

Outline

- What do we mean by Object Orientated Programming?
- Why Java?
- Structure of the Course
- Assessment/Marking
- Today's Practical
- Review/Discussion

Recommended Reading

- Introduction to Java Programming by Daniel Liang (Pearson Publishing)
  - ▷Ebook Available
  - ▷https://zjnu2017.github.io/OOP



- Chapter 1 & 2
  - ▷Read This Week

Why Java?

Why Java?

- Incredible toolset
  - Large Community
- Java is an Object Oriented language
  - internally embraces best practices
  - promotes correct usage
- Ease of learning
  - after learning the basics you'll find that you can easily grab onto more advanced concepts
- Android adoption
- Scalable
  - Easier to Maintain, Cross-Platform, Optimized Performance
- Lots of available jobs
- …

Why use Object Orientated Programming Techniques?

Why use Object Orientated Programming Techniques?

- Sooner rather than later, you'll have to work with object-oriented code
- Modularity
- Scalability
- Frameworks
- Contributing to open source software
- Gives you various ways to think and solve problems
- More easily translate your programming skills into other Object-Oriented languages
- Become a more valuable developer

Question

- A programming paradigm is a style, or “way”, of approaching a problem to come up with a solution.
- **What are the two main programming paradigms?**

Answer

- Procedural and Object-Oriented

Assessment

- 1. Attendance and Participation:10%
- 2. Experiments: 40%;
- 3. Examination: 50%.

Lessons

Topic and Teaching Aims	Lecture/Self-Study	Class Hours
01 Intro to Java (History/Facts/Features)	L	2 hrs
02 Java Basics (Grammar/Syntax)	L	4 hrs
03 Classed and Objects (Principles of Object-Oriented Programming)	L	4 hrs
04 Inheritance and Interface (Inheritance,multi-state,interface)	L	4 hrs
05 Internal classes and Exception (Internal classes/Exception handling)	L	2 hrs
06 GUI(Common components, Applications and Events Handling)	L/S	6 hrs
07 Common classes (Attributes and Operations of the Common Classes)	L/S	2 hrs
08 Database operation (Access and Operate Database)	L/S	4 hrs
09 Generic and collections (Concepts and Common Casses)	L/S	2 hrs
10 Java Multi-Threads (Multi-threading and Applications)	L/S	2 hrs
11 Network Programming (Sockets)	S	-
		Total 34

Practicals

Experiment project schedule

No.	Name	Hours	Action (Verifying/Synthetic/Designing)
1	IDE	2	Verifying
2	Java basics	4	Designing
3	Classes and objects	4	Designing
4	Inheritance and interface	4	Designing
5	Exception handling	2	Designing
6	Programming based on common classes	2	Designing
7	Programming on GUI	6	Designing
8	IO stream	2	Designing
9	Database	4	Designing
10	Multi-threads	2	Designing

Hands-On

- Hands-On Course
- Exciting & Challenging
- Practice/Work through Examples
- Experiment/Trial-and-Error
- Don’t be afraid to make mistakes
- Learn by `DOING’ (not just theory)

Contact

- Questions/Issues

Benjamin Kenwright
email: bkenwright@ieee.org

- Open Door Policy
  - ▷Problems/Help
  - ▷Within Reason

Question

- Is Javascript the same as Java?

A. Yes
B. No

Answer

(B) No

Java is not to be confused with JavaScript

Today

- Getting started with Java
- Writing/debugging simple programs
  - “Hello World”
- IDE (Integrated Development Environment)

Steps

- 1. Work through Chapter 1
- 2. Setup Java
- 3. Compile & Run Simple Java Program
  - e.g., Page 12 Introduction to OOP by Daniel Liang
  - Command Prompt
- 4. IDE
  - e.g., Eclipse (www.eclipse.org)

.java files

- .java files are txt files
  - Edit in any text editor program
- Compile .java files to intermediate binary files for that the Java Virtual Machine can execute
  - .java -> .class files
- Move to IDE to make it easier to manage your Java projects
  - Intel sense, spell-checking, …

Basics

- Class per Java file
- Class name must match Java file name
  - E.g., class Test (Test.java)

Example Question

- What is the output of this program fragment? Read it carefully!

```
String greet = "Hi";
String name = "Smedley";
String nickName = name.substring(0,4);
if (nickName == name.substring(0,4));
System.out.println("has real nickname");
else if (greet + name == greet + nickName)
System.out.println("no real nickname");
else
System.out.println("hmmm...changed names?");
```

- A. has real nickname
- B. no real nickname
- C. hmmm...changed names?
- D. it's one of the three lines given in A, B, and C above, we can't tell which one without running the program
- E. none, because there is at least one compile-time error

Answer

- E. none, because there is at least one compile-time error

```
test.java:22: error: 'else' without 'if'
        else if (greet + name == greet + nickName)
        ^
1 error
```

Summary

- Overview of the Course/Plan
- Hands-On/Practical
- Assessment (Breakdown of Marks)
- Self Study (Can’t learn from just attending)
- Today is about `Getting Started’

Questions/Discussion