

Programming in Java

1. INTRODUCTION

Hamid Fadishei, Assistant Professor

CE Department, University of Bojnord

fadishei@yahoo.com, <http://www.fadishei.ir>

About the course

In this course, you will learn advanced programming concepts and techniques such as:

- Object oriented programming
- Exception handling
- Concurrency
- Top-down software design
- Generic programming
- Software test

You'll practice to apply such concepts to the real-world problems

Course prerequisite

You are expected to be familiar with the basic programming constructs of a programming language, such as:

- Variables
- Expressions
- Loops
- Conditional statements
- Subroutines

Administrative details

You are expected to...

- Attend classes regularly
- Put away your cellphones and other electronic devices
- Not cheat on your exams, projects and homework
- Not randomly walk in and out during the class

Administrative details

Maximum allowed entrances during class are 0 and exits are 1!

Any kind of / any size of cheating will be strongly punished by **failing all involved** students

DELIVER YOUR OWN WORK ONLY!

Hit the threshold of 3/16 **absences** and you'll be **banned from final exam** and **receive a zero**

***STUDY THE EDUCATIONAL REGULATIONS
CAREFULLY!***

The penalty of cheating and excessive absences will be applied with no prior warnings and no compromises

Administrative details

The course will be presented by slides

- We'll often test-drive the sample on my laptop
- You'll get a mini-break of a few seconds at every 10 slides or so watching a clip or photo
- It is recommended to bring the hard copy of slides to the class. Print them multiple pages per sheet to save papers

The course calls for intensive effort and You'll face...

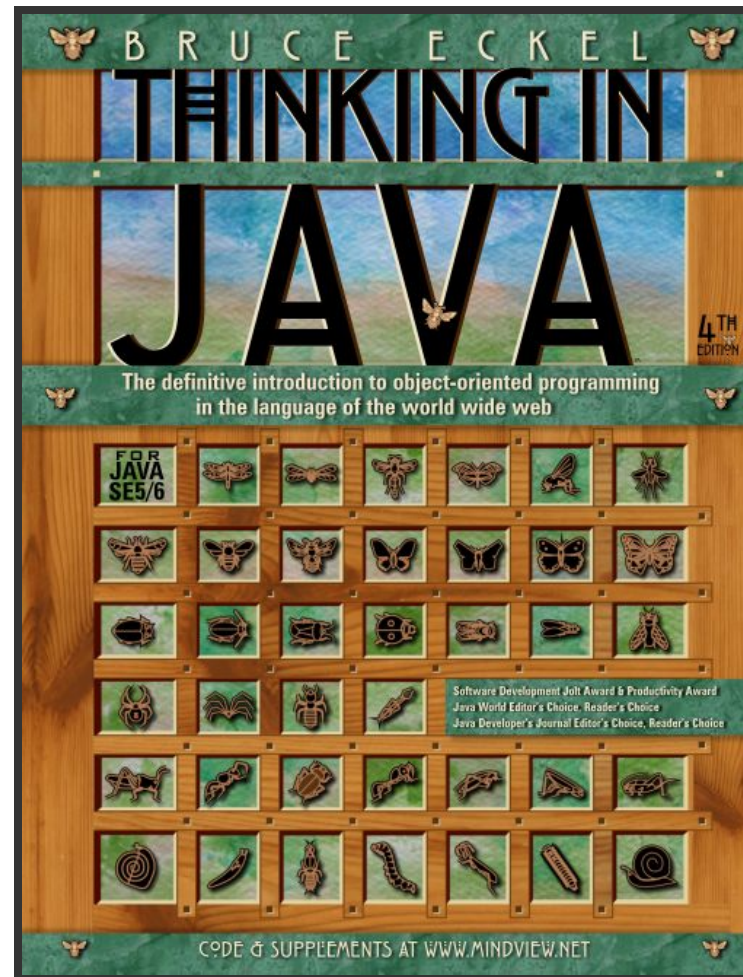
- A challenging final exam
- A challenging programming project
- Lots of homework

Course syllabus

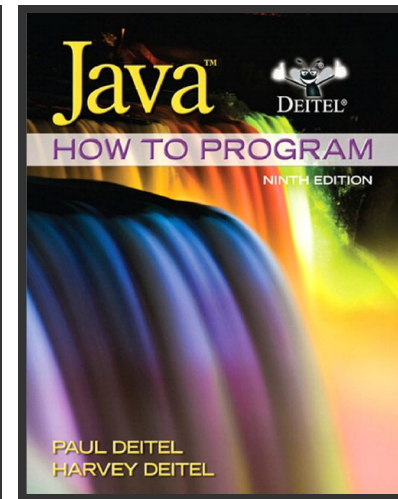
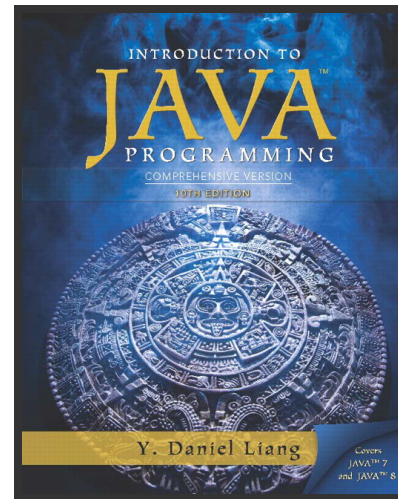
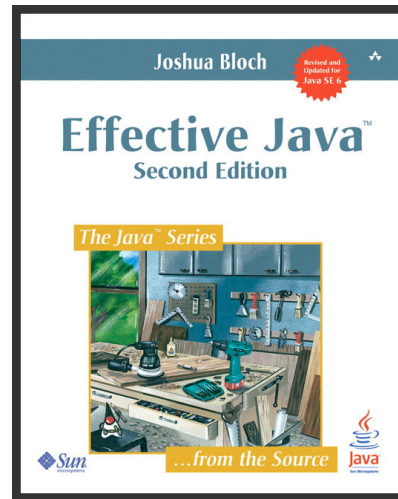
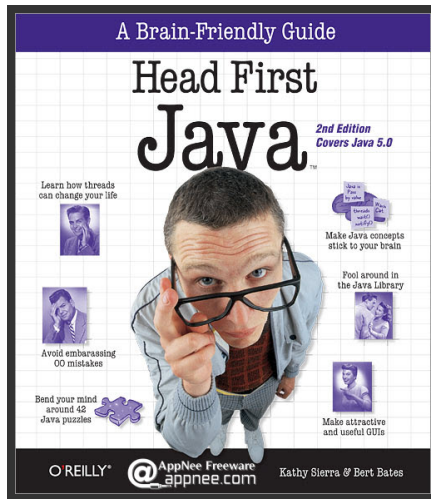
This course covers...

- Fundamentals of Java programming
- Object-oriented programming concepts
 - Objects, classes, interfaces, inheritance, polymorphism, etc
- Some professional programming skills
 - Top-down design approach, Software test, UI design, Concurrent programming, etc

Textbook



Some other recommended books



Our first Java program

```
public class Hello
{
    public static void main(String args[])
    {
        System.out.println("hello, world!");
    }
}
```

Let's run it!

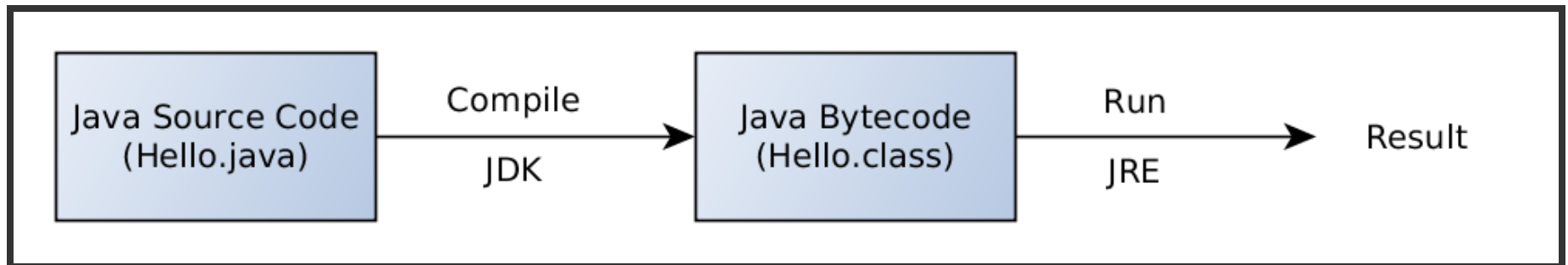
In order to run the Hello.java program

- We will need a **Java interpreter**
- Known as **Java Virtual Machine (JVM)**
- You need to install the **Java Runtime Environment (JRE)**
- Then you can use "**java**" command to run an instance of JVM

But JVM can not run the Java code directly

- It needs to be compiled to a binary form JVM understands
- Known as **Java Bytecode**
- You need to install the **Java Development Kit (JDK)**
- Then you can use "**javac**" command to compile Hello.java to Hello.class

The flow (simplified)



Homework #1

Download and install Oracle JDK 8

- Google for it
- Google for correctly setting PATH and JAVA_HOME environment variables after installing JDK

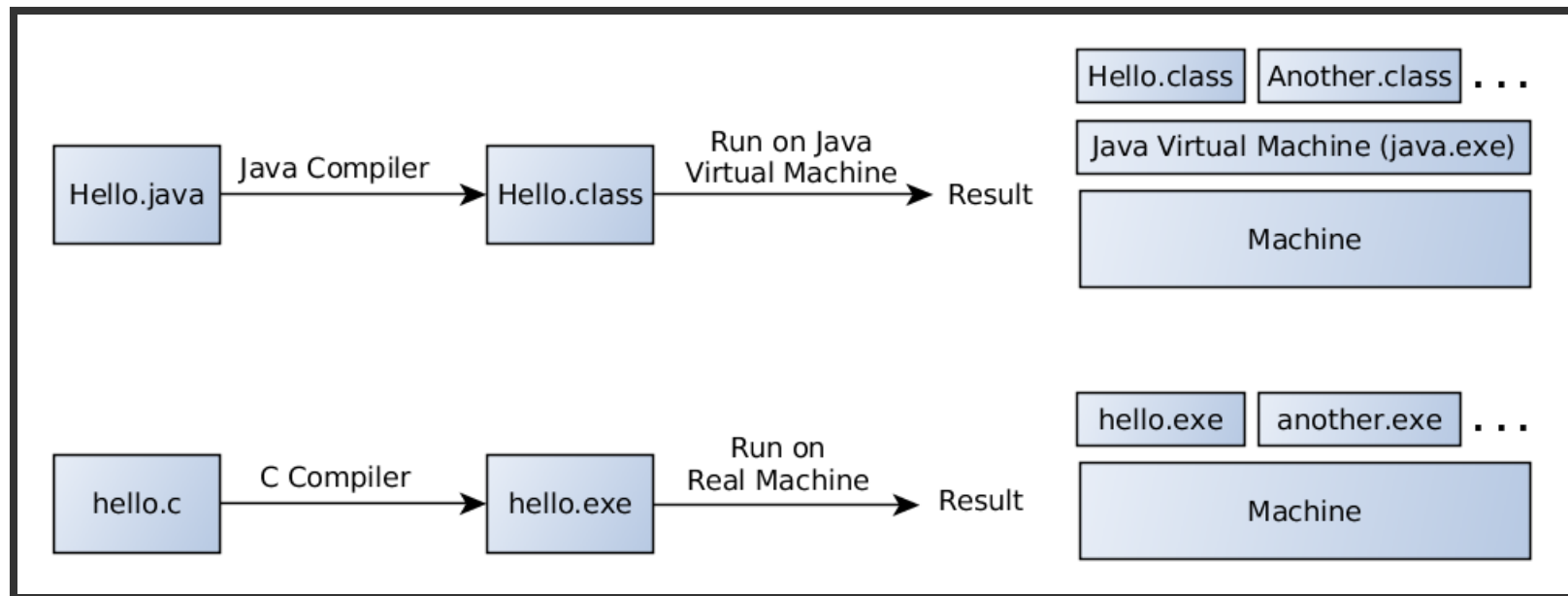
Note 1: Don't worry about JRE. Installing JDK also installs JRE.

Note 2: The teacher uses a Linux operating system. You are recommended to do the same thing, but you are free to use Microsoft Windows for this course provided you are responsible for all things specific to Windows.

Java vs C/C++

Java bytecode is not directly run by the computer. It is interpreted by JVM at runtime.

The binary compiled from a C/C++ code is directly (natively) run on the target computer.



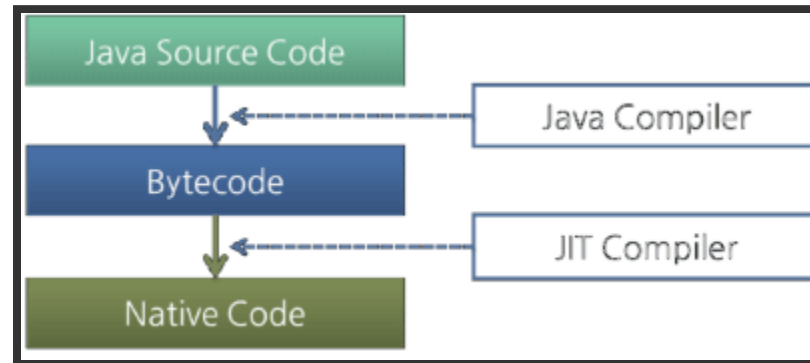
- Portability: Java wins
- Performance: C/C++ wins

Mini-break #1

Just-in-time (JIT) compiler

Helps JVM overcome performance issues imposed by the interpretation overhead

- A combination of compilation and interpretation
- Some portions of the code which are referred frequently (Hotspots) are compiled to native machine language as the bytecode is being executed



Integrated Development Environment (IDE)

As you remember from the first example, Java development flow involves several different tasks:

- Editing a source code: using a text editor like notepad or vim
- Compiling the source code to bytecode: using javac
- Running the bytecode: using java

an IDE is a software in which you can perform all these tasks easily in one place

Popular Java IDEs

The three most popular JAVA IDEs:

- Eclipse
- IntelliJ IDEA
- Netbeans

Choice of an IDE is a personal preference. But in this course you will use the Eclipse IDE

Homework #2

Download and run the latest version of Eclipse IDE for Java development (Version 4.5 AKA Mars)

Running the example inside Eclipse IDE

Let's see it in practice!...