Lecture 4: Introduction to Java and your Development Environment

COMP 250 Winter 2018

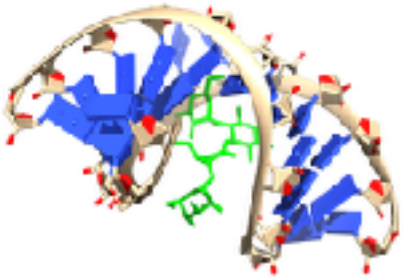
Carlos G. Oliver, Jérôme Waldispühl

Slides adapted from Mathieu Blanchette


Alaska Robotics
Programming
If I press these keys in the right order,
I can do anything!

About me

- 2nd Year PhD Student in Computer Science
- B.Sc. CS & Bio, MSc Biology
- Interested in:
  - Solving biological problems with computers and algorithms —> e.g. drug discovery
  - Cryptocurrencies and blockchain technology for scientific development



- Website: www.cgoliver.com
- Email: carlos.gonzalezoliver@mail.mcgill.ca
- Office: Trottier 3140

Housekeeping

- Office Hour Schedule is on the webpage as a Google Calendar: http://www.cs.mcgill.ca/~jeromew/comp250.html
- Office hours will be held in TR 3090
- Assignment 1 should be out by January 29th.
- **IMPORTANT:** **all** correspondence regarding the course should be sent to: **cs250@cs.mcgill.ca**
  - Otherwise, your email will likely not get read.

Java vs other programming languages

- Basic syntax very similar to C, C++
- Fully object-oriented: all code and data is within a class
- Java handles memory management: no need to allocate or free memory!
- No pointers, no segmentation faults!!

- Easy to learn and use
- Execution time slightly slower than C or C++

Installing Java

- Install the latest **Java Development Kit (JDK)** for your operating system.
  - http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
- **Recommendation:** attend OH before A1 is out to make sure you are ready to work with Java

[Downey Ch 2]

Hello World!

```
/* This programs prints a welcoming statement */
public class Welcome {
    // Every executable class have to contain a method called main like below
    // When the class is executed, main is the first method to be called
   public static void main(String args[]) {
        System.out.println("Welcome to Java!");
        System.out.println("This is easy!");
   }
}
```

File: Welcome.java

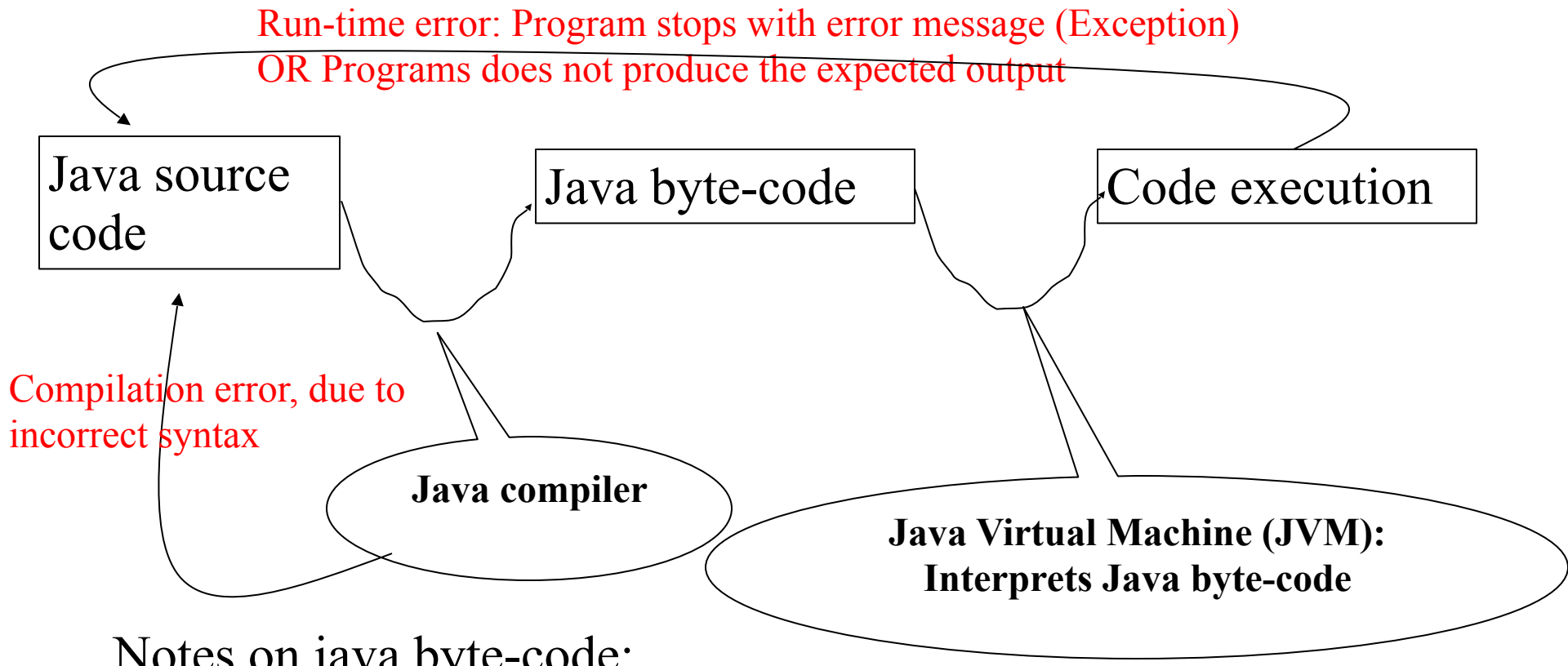
1) Compile Welcome.java
2) Run the program

Output: Welcome to Java!

This is easy!

Programming cycle in Java

[Downey Ch 1]



Notes on java byte-code:

Advantages: - Byte-code is platform independent
- Very important for internet!

Disadvantage: - Byte-code has to be interpreted by the JVM so it runs slightly slower

How do we execute these instructions?

- Two major options
  - Text editor + Command Line/Terminal (my preferred method)
  - Integrated Development Environment (IDE)

Text editor + Command Line

- Step 1:
  - Write a class in a **.java** file using a PLAINTEXT text editor (vim, sublime, etc).
- Step 2
  - Compile the source **.java** file using **javac**
  - `javac HelloWorld.java`
- Step 3
  - Execute the Byte code (.class)
  - `java HelloWorld`

Files of Interest

- Source files contain the Java code you write.
  - Always have suffix **.java**
  - Have the **same name** as the class in th
  - **These are the ONLY files we want when submitting assignments (unless otherwise stated)**
- Class files contain Java byte code
  - Always have suffix **.class**
  - Generated by the compiler

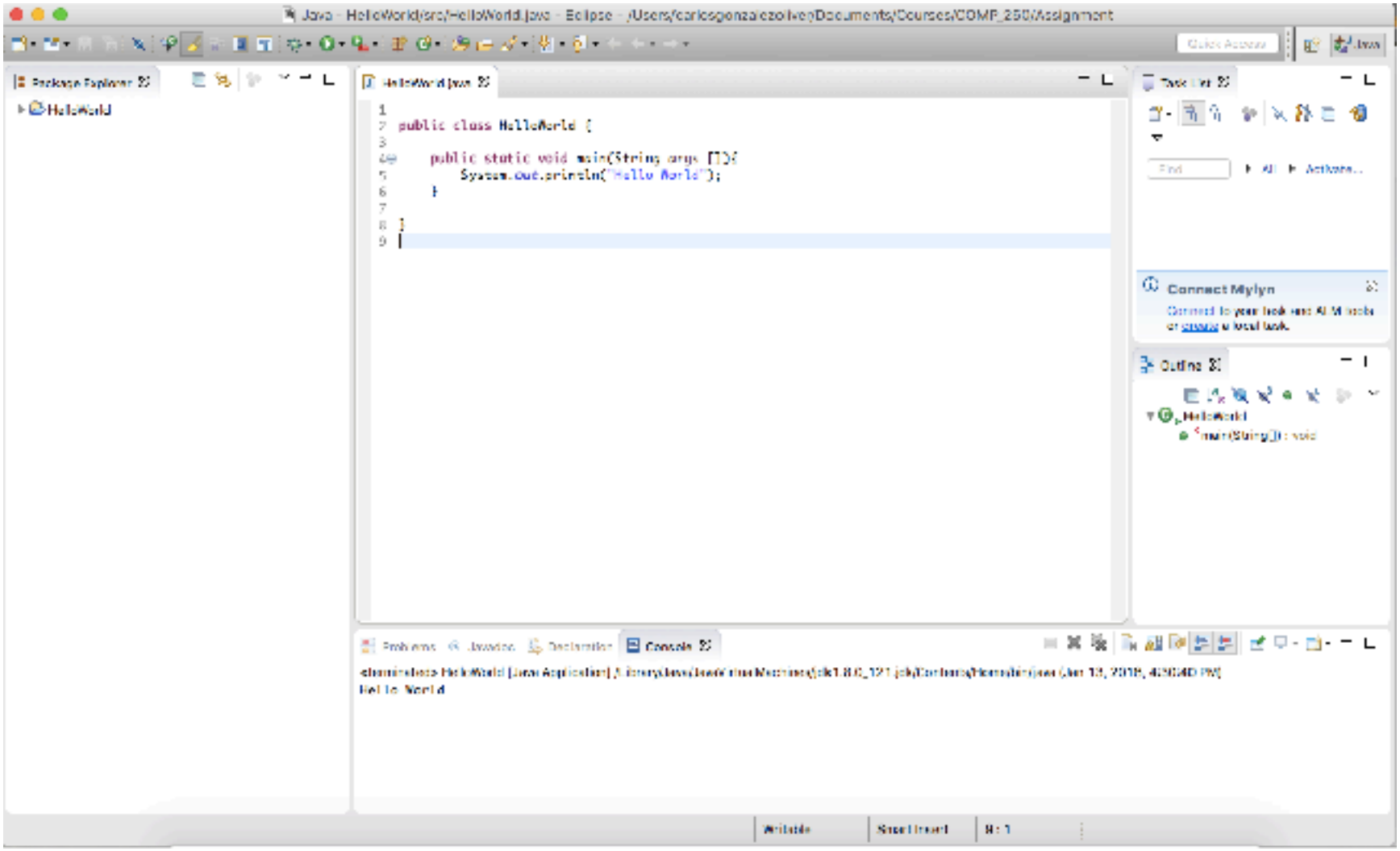
Integrated Development Environment

- IDE: Program that facilitates writing code, compiling it, running it, and debugging it.
- Recommended IDE: Eclipse
  - Freely available at http://www.eclipse.org/downloads/index.php
  - You also need to install a Java Runtime Environment (JRE), from the same URL.
  - Runs on all OS: Windows, Mac, Linux, etc.
  - Installed on all machines in Trottier
- Guide by Caitrin Armstrong from last semester on using Eclipse: http://www.cim.mcgill.ca/~langer/250/eclipse_tutorial.pdf

How to install Eclipse

https://youtu.be/V04EjDVSIJg

Hello World in Eclipse



Eclipse Disclaimer

- Eclipse sometimes generates extra files and inserts extra statements in your code.
- E.g. package declarations
- We DO NOT accept these in assignment submissions (unless otherwise stated)

IDE vs Command Line: Pros and Cons

- IDE
  - Pros
    - Automates compilation, syntax checking
    - Debugging tools
  - Cons
    - Too automatic (unnecessary modifications)
    - Many more options than necessary for us

IDE vs Command Line: Pros and Cons

- Command Line
  - Pros
    - Lightweight and simple
    - No unnecessary features
    - Often no need to install
  - Cons
    - A bit more manual work

Trottier Workstations

- We recommend you use computers on Trottier 3rd floor to test your code.
- Step 1 - get an account
  - Create a SOCS (School of Computer Science) account
  - Option 1
    - If you are on campus you can do it on this link: http://newuser.cs.mcgill.ca/newuser/
  - Option 2
    - Directly on any computer in Trottier 3rd floor. Follow instructions on the screen

Trottier Workstations

- Step 2 - log in to your account
  - Option 1: run your code from Trottier
    - Log in normally and test
  - Option 2: remotely (from home or anywhere around the globe) a.k.a ssh
    - https://www.cs.mcgill.ca/docs/accounts
    - In your terminal (MacOS or Linux):
      - `ssh username@mimi.cs.mcgill.ca`
    - Windows, install PuTTy http://www.putty.org/