CSCI 209: Software Development

Sara Sprenkle
sprenkles@wlu.edu
September 11, 2009

Sep 11, 2009   Sprenkle - CS209   1

What is Programming?

"If you don't think carefully, you might think that programming is just typing statements in a programming language."
--Ward Cunningham

Sep 11, 2009   Sprenkle - CS209   2

Discussion: What Is *Good* Software?

- What are its outcomes?

- What are the characteristics of the software?

- How can we write good software?

Sep 11, 2009   Sprenkle - CS209   3

Characteristics of *Good* Software?

- Free of bugs
  - Robust, reliability
- Code is easy to read, extend, maintain
  - Readability, extensibility, maintainability
- Application is easy to use
  - Usability
- Efficiency
- Scalability

➔   Referred to as the *ilities

Sep 11, 2009   Sprenkle - CS209   4

Ward Cunningham

"If you don't think carefully, you might think that programming is just typing statements in a programming language."
--Ward Cunningham

- American computer programmer
- Developed the first **Wiki** (1994)
  - Pioneer in **design patterns** and **Extreme Programming**

    Will mean more to you later this semester

Sep 11, 2009   Sprenkle - CS209   5

Course Content

- Software Design Principles
- Java
  - Statically typed language
- Software development tools
  - Eclipse
  - Version Control Systems
  - Some UNIX

Sep 11, 2009   Sprenkle - CS209   6

What to Expect from this Class

- Programming intensive
  - ➢ Interesting assignments and projects
  - ➢ More freedom in design, *ilities
    - • Larger portion of your grade
    - • Correctness is **NOT** enough
  - ➢ Building on large library of classes
  - ➢ Building larger applications
- Compare/Contrast with Python
- Learning on your own
  - ➢ Online resources

Class Details

- Course Web Site
  - ➢ Example code, lecture slides, readings, resources
  - ➢ Course wiki/blog
- Optional Textbooks
  - ➢ Use plentiful online resources instead!
- Participation
  - ➢ Class discussions

Class Details

- Programming Assignments
  - ➢ Various sizes
- Reading assignments
  - ➢ Write up short summaries on Sakai
  - ➢ Class discussions
- Several Projects
- 2 Exams

Course Dynamics

- Professor's Responsibilities:
  - ➢ Be prepared for class
  - ➢ Provide non-judgmental feedback to students
  - ➢ Treat students with respect
  - ➢ Challenge and encourage students
  - ➢ Make material as clear as possible
- Student's Responsibilities
  - ➢ Be prepared for class (do readings and homework)
  - ➢ Give attention and effort in class to learning
  - ➢ Ask questions (**during class** and via email)
  - ➢ Use professor's office hours
  - ➢ Let professor know if something is going wrong
  - ➢ Treat other students and professor with respect

My Bio

- From Dallastown, PA
- B.S., Gettysburg College
- M.S., Duke University
- Ph.D., University of Delaware
- For fun: ultimate, pop culture, ACC basketball

My Research Interests

- General: Software engineering
- Automated testing of web applications
  - ➢ Develop algorithms
  - ➢ Implement in tools

Your Bios

- Where you're from
- What activities you're involved in
- What you do in your free time
- Any experience with Java or Python?

Sep 11, 2009   Sprenkle - CS209   13

What is Java?
... and, why should I learn it?

- From Sun Microsystems
  - ➢ 1995, James Gosling and Patrick Naughton
  - ➢ Specifications
- Object-oriented
- Rich and **large** library
- Develop cross-platform applications
  - ➢ Web, desktop, embedded
  - ➢ Frameworks
- Widely used

Sep 11, 2009   Sprenkle - CS209   14

What is Java?

- Java Programming Language
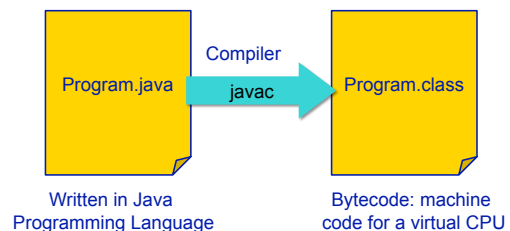- Java Virtual Machine
- Java Class Libraries

Sep 11, 2009   Sprenkle - CS209   15

Java Programming Language

- Similar to Python
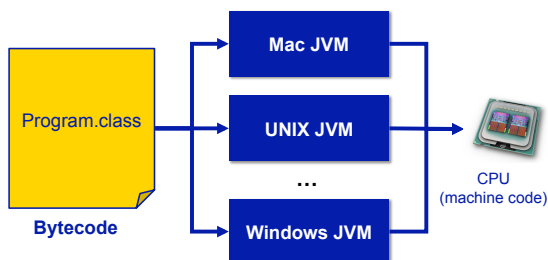- But *entirely* object-oriented*

Label the Python equivalents

Program.java → Compiler javac → Program.class

Written in Java Programming Language

Bytecode: machine code for a virtual CPU

Sep 11, 2009   Sprenkle - CS209   16

Java Virtual Machine (JVM)

Program.class (Bytecode) → Mac JVM / UNIX JVM / ... / Windows JVM → CPU (machine code)

- Same **bytecode** executes on each platform

Sep 11, 2009   Sprenkle - CS209   17

Java Virtual Machine (JVM)

- Emulates the CPU, usually specified in software
- Executes the program's *bytecode*
  - ➢ Bytecode: virtual machine code
- Different versions for each platform Java supports
  - ➢ Enables program portability
- Sun's Hotspot VM
  - ➢ Code dynamically compiled to machine code
- Garbage Collection

Sep 11, 2009   Sprenkle - CS209   18

3

Traditional Way

```
Program  →  Platform-specific   Platform-specific
            Compiler             Executable
```

- Executable is not portable

Discussion:
How does Java's approach ease distribution of software?

Java Editions

- *Edition*: Version of Java
- 3 different editions
  - **SE**: Standard Edition
  - **EE**: Enterprise Edition
    - Server-side applications
    - Web applications, Communication (mail)
  - **ME**: Micro Edition
    - For PDAs, mobile devices, etc.

Java Class Libraries

- Pre-defined classes
  - Included with Java 2 Software Development Kit (SDK) and Java 2 Runtime Environment (JRE)
  - View the available classes online:
    http://java.sun.com/javase/6/docs/api/
- Similar in purpose to modules included with Python

Summary of Java Platform SE 6.0

Image from Sun's site

Benefits of Java

- Rapid development of programs
  - Large library of classes, including GUIs, Enterprise-level applications, Web applications
- Portability
  - Run program on multiple platforms without recompiling
- Statically-typed language
  - Compiling - find some errors before execution
  - Can give performance boost

Aside: JavaScript vs Java

- JavaScript is **not** Java
  - JavaScript is a *scripting* language, primarily embedded in HTML, executed by Web browsers

Web Browser
JavaScript

```
<script type="text/javascript">
function myFunction() {
    return ("Hello, have a nice day!")
}
</script>
</head>
<body>
<script type="text/javascript">
    document.write(myFunction())
</script>
```

Which 'Java' is this class about?

- Java programming language
- Java class libraries

- Use the JVM but won't learn about how it works
  - For more information:
    http://java.sun.com/docs/books/vmspec/

Java Development Kit (J2SDK)

- J2SDK: Java 2 Software Development Kit
- Free from Sun
- Contains
  - `javac`: Java compiler
  - `java`: Java Virtual Machine
  - Java class libraries

Java Development Kit (J2SDK)

- Installed on Linux machines
  - Java 1.6 should be reachable using default path
  - To see which executable you're executing use
    - `which java`
    - Should be `/usr/bin/java`
- Run `java -version` to determine which version you're running
- You can download the JDK for your machine from http://java.sun.com/javase/downloads/index.jsp
  - JRE is for *running* Java applications
    - Does not include the compiler

Using the J2SDK

- Copy `First.java` from `/home/courses/cs209/handouts/`
- Compile and run `First.java`
  - `javac First.java`
    - Compiles the program into `First.class`
  - `java First`
    - Runs the JVM, which executes the bytecode
- View `First.java` in jEdit

Summary

- Today
  - Overview of Java programming language
  - Example program
- Next time:
  - Data types
  - Control structures
- Your To Do
  - Review UNIX commands
  - Check out course web site