

Lecture 1

08-671

Java Programming for App Developers

September 1, 2015

Jeffrey L. Eppinger

Professor of the Practice
School of Computer Science

How many are in the class?

- Still trying to get in?
 - Stop by and see me after class

Our Teaching Assistants

1. Renzo Bautisa
2. Anny Ni
3. Hao Fu
4. Akhil Prakash
5. Hao Tang
6. Yifu Wang
7. Xinyue Wu
8. Tianyue Xiao

See Blackboard/Piazza for office hours & additional contact info

Outline

→ Why We Are Here

Introductions

HelloWorld.java

Java Overview

HelloWorld Revisited

Homework & Readings

Why We Are Here

- Java Programming for masters students
- Goal: fast pace, practical focus
- Covering things that undergrads don't cover
 - Internet access
 - Database
 - Object-Relational Mapping Tools
 - General perspective

The screenshot shows a Mac OS X desktop with a browser window open. There are five tabs visible in the top bar:

- New Tab
- Home Page
- Carnegie Mellon fulfilled, ...
- How to take a screenshot ...
- We're Sorry, Applicants. W...

The main content area of the browser displays the header "The New York Times" and the article title "We're Sorry, Applicants. We Accepted You in Error." Below the title is the byline "By RICHARD PÉREZ-PÉNA FEB. 18, 2015". To the left of the main content are social sharing icons for Email, Share, Tweet, Pin, Save, and More.

EDUCATION

We're Sorry, Applicants. We Accepted You in Error.

By RICHARD PÉREZ-PÉNA FEB. 18, 2015



Email



Share



Tweet



Pin



Save



More

Acceptance to a world-class graduate school, in a highly competitive field, offers a path to credentials that open doors throughout a career, a stamp of validation to last a lifetime. Or maybe a few hours.

Carnegie Mellon University this week emailed about 800 applicants to a graduate computer science program word that they were accepted, only to email them again later the same day to say, in effect: Oops, not really. It became the latest in a string of big-name colleges to make similar mistakes. But this one had a particularly cringeworthy twist: A university renowned for its computer science offerings had fulfilled and then dashed applicants' hopes with a computer foul-up.

That Carnegie Mellon, in Pittsburgh, had ample, exalted company in an admissions gaffe worthy of Emily Litella (the "Saturday Night Live" character who always ended her sketches with the words, "Never mind") — including the Massachusetts Institute of Technology and Johns Hopkins University in the last year — did nothing to cushion the blow.

(Source: New York Times)

Bad Press was Everywhere

qz.com/345752/carnegie-mellon-fulfilled-then-horribly-crushed-the-dreams-of-800-budding-computer-scientists/  Search   

OUR PICKS	LATEST	POPULAR	QUARTZ	OBSESSIONS
-----------	--------	---------	--------	------------

OOPS

Carnegie Mellon fulfilled, then horribly crushed, the dreams of 800 budding computer scientists



How Did it Happen?

- It's was a coding bug
 - Followed by several human errors
- Some details are provided in Frank Pfenning's Google+ post of 4-Mar-2015 ↗



Frank Pfenning

Shared publicly - Mar 4, 2015

As has been widely reported by now, the Computer Science Department recently sent incorrect admission emails to about 800 applicants to the MS in CS program at CMU. We had to retract a few hours later, which caused a great deal of understandable distress among the applicants. Here, I would like to explain how this came about as an unfortunate confluence of a single program error, followed by a chain of several human mistakes.

In the 2013-14 cycle, a bug was introduced into the graduate application and review system. This bug escaped testing, and in retrospect we do not think our testing framework was adequate. The bug was not exercised at all during the 2013-14 cycle. It arose when a database field was changed from a number value to a string, and one comparison was not appropriately updated. Ironically, for a programming language researcher like myself who has dedicated most of his career to developing expressive disciplines of static typing, this should have been flagged as a static type error but the code was written in PHP.

In the 2014-15 cycle, a change in procedure led to this bug corrupting the database by setting the "admit" bit for many applicants when their applications were marked as "complete".

Alert administrators discovered this before PhD admission letters went out in early February. The bug was fixed, and all affected degree programs except one were manually corrected. The MS in CS program was overlooked.

All program administrators were notified of the problem and were asked to double-check their admit lists.

Java Type Checking

- Java is strongly typed
 - You have to declare the types of your variables
- Java has static type checking
 - It detects many problems at compile time
- In PHP, what is the meaning of: if (\$admit) ...
 - 0 is false 1 is true
 - "0" is false "defer" is true
 - "" is false "reject" is true

Java is More “Work” to Write

- It can be quicker to write the code in Python
 - Dynamically typed, interpreted
 - Less to write
- But Java will detect more problems earlier
- Programmers may be able to more easily understand Java programs when they are read
- What is the cost of an undetected bug?

Outline

✓ Why We Are Here

→ Introductions

HelloWorld.java

Java Overview

HelloWorld Revisited

Homework & Readings

About Me

Jeff Eppinger (eppinger@cmu.edu, Wean 5124)

Ph.D. Computer Science (CMU)

Asst Professor of Computer Science (Stanford)

Co-founder of Transarc Corp. (Bought by IBM)

- Transaction Processing Software
- Distributed File Systems Software

IBM Faculty Loan to CMU eCommerce Inst. (99-00)

Now an the faculty of the Institute Software Research

Most Significant Qualification to Teach Java:

- James Gosling was my officemate in grad school

My Co-Instructor

Terry Lee (terrylee@cs.cmu.edu, Wean 4126)

Researcher, Korea Info Security Agency (2002-2007)

- MSIT eBusiness Technology (CMU, 2007)

Software Engineer, Vivisimo (2007-2010)

Program Faculty in eBusiness Program (2010-)

Assistant Teaching Professor (2014-)

Our Teaching Assistants

1. Renzo Bautisa
2. Anny Ni
3. Hao Fu
4. Akhil Prakash
5. Hao Tang
6. Yifu Wang
7. Xinyue Wu
8. Tianyue Xiao

See Blackboard/Piazza for office hours & additional contact info

You

- MSIT eBiz – 75
- Mech E 2
- INI 7
- MISM –
- BME
- ECE 4
- MSE 2

Course Meetings

- Location: Baker Hall, Room A51
- Two Lectures per Week
 - Tuesday & Thursday 12:00pm – 1:20pm
 - We'll cover new material here
 - Lectures will be video-recorded & available via the Internet
- One Recitation per Week
 - Fridays, 1:30pm – 2:50pm, in DH A302
 - At this time, attendance will be required
 - A chance for Q&A in a more focused setting
 - A chance to critique student programs
 - Quizzes

Textbooks

- You can use any textbook you like
 - When we assign readings, we'll tell you the topics
 - Read these topics in your book or on the web
- In the past, students have liked:
 - SAMS Teach Yourself Java in 21 Days
- Now popular is:
 - Head First Java (O'Reilly)
- Nevertheless, we recommend specific textbook
 - Head First Java (O'Reilly)
- Readings will be from the Head First book



08-671 Lecture Topics

(subject to change – but only a little bit)

#1 Intro

#2 Primitive Types

#3 Java Classes

#4 Reference Types

#5 Loops & Arrays

#6 Lists & Sorting

#7 Maps

#8 File & Network I/O

#9 Swing Interfaces

#10 Swing Actions

#11 Threads

#12 Exceptions

#13 Functional Programming

#14 In-class Written Exam

* Final Exam – this will be a 3-hour programming problem

The Course Website

Up to date info (will be) available on the “blackboard”

<http://www.cmu.edu/blackboard>

- Syllabus
- Copies of lecture slides
- Links to lecture videos
- Java code for examples shown in class
- Homework assignments

Discussion Board

- We will set up a Course Piazza
 - This will just be for Q&A

Homework Assignments

- Seven Homework Assignments are Planned
- Most will be graded using AutoLab
 - You will be able to submit multiple times
 - So you can correct any errors to get all the points
- The last (two) assignments will be graded by TAs
 - But still turned in via AutoLab

Homework Late Policy

Homework #1: 5% penalty per day
(or fraction thereof)

Homework #2: 10% penalty per day

Homework #3: 15% penalty per day

Homework #4: 20% penalty per day

Homework #5: 25% penalty per day

Homework #6: 30% penalty per day

Homework #7: 35% penalty per day

Course Exams

- One in-class “written” exam
 - Last class meeting, Thurs, Oct 15th
 - Location may be different
- One 3-hour Final Exam
 - Week of Oct 19th
 - Registrar will announce date & time
 - Programming problem
 - On your laptop
 - Similar to a homework assignment

Docs on the Web

Java docs are on the web:

<http://docs.oracle.com/javase/7/docs/api/>

<http://docs.oracle.com/javase/8/docs/api/>

Docs can also be downloaded into your machine
from <http://www.oracle.com> -> Downloads

Hardware

- You are expected to provide your own hardware for this course
- Who doesn't have their own hardware?
- Who doesn't have a laptop to use for this class?

Course Grade Computation

30% – 40% Homework

- Easier (earlier) homework weighted less

0% – 5% Quizzes

20% – 25% Written Exam

30% – 40% Final Exam

Collaboration Policy

- Everyone should read and abide by the University Policy:
<http://www.cmu.edu/policies/documents/Academic Integrity.htm>
- Here is some additional information for this course:
 - You are allowed to talk with/work with other students on homework assignments
 - You can share ideas and can examine/critique each others' solutions
 - You must turn in your own work
 - Your homework solution should be different than others
 - You may electronically copy any examples given in class to use as a basis for your homework solutions
 - You may not electronically copy files, or portions thereof from any other source when completing your homework (unless you have permission from course staff)
 - You may not let someone else prepare your homework for you
 - You may not let another student electronically copy your file(s) or portions thereof
 - See the Collaboration Policy on the Course Blackboard for more details
 - Minimum penalty for copying files:
 - Zero for that assignment (50% penalty for the source)
 - You'll probably get reported to the university administration, too

When Do People Cheat?

- Usually, when they are late on homework

If You're About to Cheat...

- Stop and come talk to me and see if we can work something out
 - Remember – this goes for the copier and the source!

MOSS

- There is software which we use to compare your solutions
 - It's called “MOSS”
- Studies show...

My Goals

What should you expect to get from this class?

Intuition about writing software

- What's possible
- What's hard
- How long it takes
- Whom to hire
- Who is giving you the run around

Ability to write moderately complex Java Programs

- Ability to find what you need in Java
- Ability to do programming for other classes/tasks

What Are Your Goals?

- Eat
- Do Homework
- Sleep
- Get A+

08-671 Lecture Topics

(subject to change – but only a little bit)

#1 Intro

#2 Primitive Types

#3 Java Classes

#4 Reference Types

#5 Loops & Arrays

#6 Lists & Sorting

#7 Maps

#8 File & Network I/O

#9 Swing Interfaces

#10 Swing Actions

#11 Threads

#12 Exceptions

#13 Functional Programming

#14 In-class Written Exam

* Final Exam – this will be a 3-hour programming problem

Outline

✓ Why We Are Here

✓ Introductions

→ HelloWorld.java

Java Overview

HelloWorld Revisited

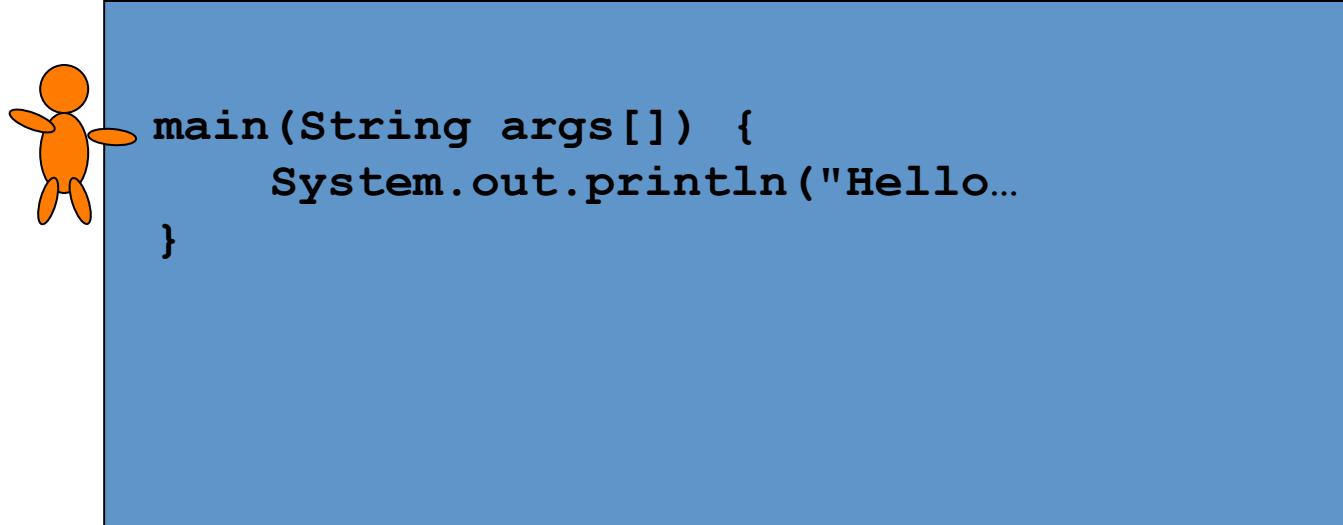
Homework & Readings

HelloWorld.java

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello World");  
    }  
}
```

How does a program work?

- When you run a program, Java starts a little man running in the main method. The little man speaks only Java. He diligently does whatever the Java program says to do.



Outline

✓ Why We Are Here

✓ Introductions

✓ HelloWorld.java

→ Java Overview

HelloWorld Revisited

Homework & Readings

Java

- Java is the coolest technology that you'll learn while at Carnegie Mellon
- Here's why:
 - Programming Language with the “Right Stuff”
 - Runs “everywhere”

Right Stuff?

- Object-oriented
- Network-oriented (Security-oriented)
- Multithreaded
- High-performance
- Easy-to-use

Object-oriented

- The Data
 - Numbers, strings, etc, that represent the object
- The Methods
 - Operations you can perform on the data
- Concept
 - You don't need to understand the data or the implementation of the methods
- In Java these are called classes
- Examples of classes:
 - [String](#) (or more formally [java.lang.String](#))
 - StockQuote

Example Object: StockQuote

- Constructor:

```
StockQuote(String ticker)
```

- Instance Methods:

```
String currentQuote()
```

```
float getPrice()
```

```
String getName()
```

```
String getChange()
```

Let's Try It: StockQuoteTest.java

```
public class StockQuoteTest {  
    public static void main(String args[]) {  
        StockQuote sq = new StockQuote(args[0]);  
        System.out.println(" " +  
                           sq.getName() + " " +  
                           sq.currentQuote());  
    }  
}
```

Network/Security Oriented

- Class Libraries to access the network
- Portable: “write once, run anywhere”
 - Interpreted byte codes
- Sandbox
 - Allows you to run untrusted code
 - Downloaded applets have restricted access
- Libraries to access security features
 - ACLs, Certificates, Encryption
- Example
 - [java.net.URL](#)

Selected Guts of StockQuote.java

```
public class StockQuote ... {  
    ...  
    private URL url;  
    ...  
    private final String urlHeader =  
        "http://quote.yahoo.com/d/quotes.csv?s=";  
    ...  
    String urlString = urlHeader+ticker+urlTrailer;  
    ...  
    url = new java.net.URL(urlString);  
    ...  
    InputStream is = url.openStream();  
    ...  
}
```

Large Class Libraries

- Need a class that does something?
 - Chances are it's already been written!
- The Java Foundation Classes
 - Packaged with Java
 - Provides everything from data structures to GUIs
- Examples
 - ArrayLists (implemented by [java.util.ArrayList](#))
 - Swing GUIs ([javax.swing](#))
 - Applets (e.g., [java.swing.JApplet](#))

Example: java.util.ArrayList

(for those of you with programming background...)

- Implements an array of arbitrary size
- Incredibly handy

Selected Constructor:

ArrayList()

Selected Instance Methods:

add(obj)

add(obj, pos)

Example: ToDoSwingGUI.java

Code excerpt from: `ToDoSwingGUI.java`

```
public class ToDoSwingGUI extends JFrame ... {
    JButton topButton, bottomButton;
    JTextField textField;
    JTextArea textArea;
    ArrayList toDoList;

    public ToDoSwingGUI() {
        ...
    }

    ...

    public static void main(String[] args) {
        new ToDoSwingGUI();
    }
}
```

Multithreaded

- Lets programs do multiple things at a time
- Your PC is “multithreaded”
 - Browser can run multiple windows
- Typical uses:
 - Separate threads to perform I/O and computing
- Example:
 - StockQuoteMTTest.java

Example: StockQuoteSwingGUI

Code excerpt from `StockQuoteSwingGUI.java`

```
public class StockQuoteSwingGUI extends JFrame ... {  
    private JButton button;  
    private JTextField textField;  
    private JTextArea textArea;  
  
    private java.util.ArrayList quoteHistory;  
  
    public StockQuoteSwingGUI() {  
        ...  
    }  
  
    ...  
  
    }  
}  
public static void main(String[] args) {  
    new StockQuoteSwingGUI();  
}
```



High Performance

- High being a relative term:
 - Faster than most scripting languages (JavaScript, Python, PERL, etc)
 - (Slower than C or C++)
- Threading can improve throughput
- Many people working on optimization tools
 - JIT: Just-in-time compilation
- Your machine is fast, you can afford to run Java

Easy-to-use

- (Easy being a relative-term)
- Easier than other programming languages
 - No pointers
 - Automatic Garbage Collection
- Less cryptic than some scripting languages

Right Stuff?

- Object-oriented
- Network-oriented (Security-oriented)
- Multithreaded
- Large class libraries
- High-performance
- Easy-to-use

Runs Everywhere

- You can run this stuff anywhere
 - On Windows, Mac, Linux, Unix, Mainframe, ...
 - From MS-DOS Prompt
 - In Web Browsers
 - Using Applets!
 - In Web Servers
 - Using Servlets and EJBs
 - In Smart Cards
 - In Integrated Development Environments

Integrated Development Environment

- Source-code, syntax-directed editor
- Incremental compiler
- Repository-based environment for code
- Project-based development
- Integrated debugging
- Support for team development
- Tools to facilitate specific programming models
 - Applets, Servlets, EJBs, Stored Procedures, XML, ...

Outline

- ✓ Why We Are Here
- ✓ Introductions
- ✓ HelloWorld.java
- ✓ Java Overview
- HelloWorld Revisited

Homework & Readings

So Let's Write a Java Program

- Traditional first program called HelloWorld
- Why...all is does is output “Hello, World!”
- Think of it as cookbook...

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

Download Java 8

- The Java Standard Edition Developers Kit
 - Abbreviated: J2SE JDK
 - Downloads Page: <http://java.oracle.com>
- Download current version, if not already on laptop
 - Download the Standard Edition (SE) JDK (Java Dev't Kit)
 - I recommend not using NetBeans
 - We're going to use the Eclipse IDE soon

Windows: Enter the Program in WordPad

- Start -> Programs -> Accessories -> WordPad
 - File -> New ... Text Document
 - Enter the program

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World!");  
    }  
}
```
- Save the program as `c:\08-600\HelloWorld.java`
 - Capitalization counts for these names
- In Windows Explorer...if necessary, right-click to rename:
`HelloWorld.java.txt` to `HelloWorld.java`

MAC/Linux: Enter the Program in WordPad

- Use a text editor to enter the program
- On the MAC:TextEdit use “Make Plain Text” under Format

Compile and Run

- Compile **(This generates HelloWorld.class)**

```
javac HelloWorld.java
```

- Run(This runs HelloWorld.class)

```
java HelloWorld
```

Compile

- Looks in other (class) files you reference
- Checks for errors
- Translates `.java` files into `.class` files
 - The `.class` file is a hardware independent byte-code representation of the `.java` file
 - (Compare this with other languages that compile down to hardware dependent instructions.)

Windows: You Want to Set the Default Path?

- (So you won't have to do it every time you start the Command Prompt Window...)
- Start -> Settings -> Control Panel -> System
 - Click on “Advanced” tab
 - Click on “Environment Variables” button
 - Select “PATH” under System Variables
 - Click “Edit” button
 - Append “;`c:\jdk1.x.x_xx\bin`”
 - Click “OK” button
 - Click “OK” button

FYI

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

- Things we will talk more about:
 - Each `.java` file defines a class (what's class?)
 - The file name and the class name must match
 - Java programs start in a method called `main()`
 - The `main()` method is always `public static void` with an array of `Strings` as the arguments

Outline

- ✓ Why We Are Here
- ✓ Introductions
- ✓ HelloWorld.java
- ✓ Java Overview
- ✓ HelloWorld Revisited
- Homework & Readings

Homework #1

- Write a variation of the HelloWorld program.
 - When run, your program should output the following:
 - Your Andrew ID & Name
 - Your favorite restaurant and its best dish
 - Compile & test your program
 - Fix the bugs and repeat
 - Submit via Autolab
 - Fix the bugs and repeat
 - See additional details on the Blackboard (under Assignments)
 - Due Friday, 9/4 at midnight

Readings

- Find your book on Java
 - Start reading the introductory chapters:
 - How Java works
 - Intro to program structure
 - Intro to objects
 - In Head First Java: Chapters 1 & 2
 - In Java in 21 Days: Days 1, 2 & 3

Outline

- ✓ Why We Are Here
- ✓ Introductions
- ✓ HelloWorld.java
- ✓ Java Overview
- ✓ HelloWorld Revisited
- ✓ Homework & Readings