CIS 110: Introduction to Computer Programming

Lecture 1

An introduction of an introduction

(§ 1.1 – 1.3)*

*(no, you shouldn't have read those yet =P)

Outline

1. What is computer science and computer programming?
2. Introductions and logistics
3. The anatomy of a Java program

What is computer programming?

What is computer science?

(By demonstration!)

What is Computer Science?

"Computer programming is not computer science"

- Programming languages
- Software engineering
- Artificial intelligence/machine learning
- Graphics
- Computer architecture
- Compilers
- Embedded and real-time systems
- Formal methods
- Theory of computation
- Computer security and privacy
- Databases and data management
- Operating systems and networking
- Mobile, distributed, and ubiquitous systems
- Computational biology
- Algorithms and complexity analysis
- Human-computer interaction

What unifies all these crazy, different things?

Algorithmic Thinking

- *Algorithm*: a step-by-step procedure to solve a problem
- ***Algorithmic thinking****:* a structured approach to problem solving

It represents a universally applicable attitude and skill set everyone, not just computer scientists, would be eager to learn and use.

- Jeannette M. Wing (CMU professor)

The Three Skills of Algorithmic Thinking

- *Precision*
  - "Accurately and completely describe how to solve a problem"
- *Decomposition*
  - "Break up a big problem into smaller ones."
- *Abstraction*
  - "Recognize that several problems are the same."

How are CS and Programming Related?

It has often been said that a person does not really understand something until after teaching it to someone else. Actually a person does not *really* understand something until after teaching it to a *computer*, i.e., expressing it as an algorithm.

- Donald Knuth

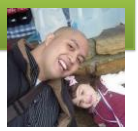
What is Computer Programming?

- Two things for our purposes:
  - *A way to practice algorithmic thinking skills in a concrete way*
  - *A practical skill you can use in your own job*

While computer programming is not computer science, programming allows us to exercise the core skills that all computer scientists possess. Also, sharpening our algorithmic thinking makes us better programmers!

Logistics

About me



- My name: Peter-Michael Osera.
  - Call me whatever you want, e.g.,

Peter	Mike	Pete-Moss
Pete	Mikey	(Lots more, many inappropriate)
Michael	PM	

- I am a
  - 4th year Ph.D. student (*not* a professor).
  - Programming languages researcher.
  - Former program manager @ Microsoft (VC++ compiler).
  - Die-hard supporter of e-sports and pro-gaming.

CIS 110: the Highlights

- 2 lecture offerings (11-12 and 1-2)
- *Required* lab section (10% of your grade)
- *Required* textbook: *Building Java Programs: A Back to Basics Approach*, Reges and Stepp
- Piazza message board system
- No curve, slide if necessary
- Late day policy (4 late days)
- 2 exams, 1 final

See the syllabus for more details

CIS 110 vs. CIS 120

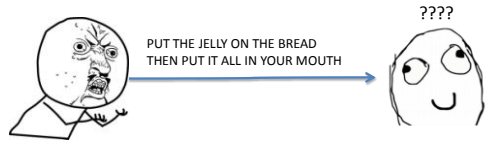
- CIS 110 and 120 = the CS intro sequence.
  - Both emphasize *algorithmic thinking* via programming.
- CIS 110:
  - Assumes no prior programming experience.
  - Focuses on *control* issues in programming.
  - Uses the Java programming language.
- CIS 120:
  - Assumes prior programming experience.
  - Focuses on *data representation* issues in programming.
  - Initially uses OCaml then goes back to Java.

To swap classes, please speak to your instructor.

Computer Programming

What is a Computer Program?

- Imagine that we need to tell our best friend to do something, but he doesn't speak our language.



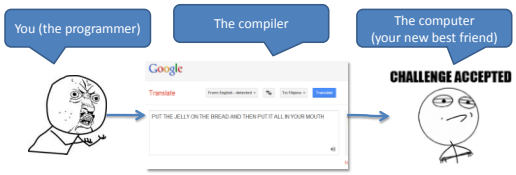
Translation

- We can *translate* our instructions so that our best friend understands them.

PUT THE JELLY ON THE BREAD AND THEN PUT IT ALL IN YOUR MOUTH

Ilagay ang halaya SA tinapay AT pagkatapos a y ilagay ito LAHAT SA INYONG bibig

The analogy revealed



PUT THE JELLY ON THE BREAD AND THEN PUT IT ALL IN YOUR MOUTH

Ilagay ang halaya SA tinapay AT pagkatapos a y ilagay ito LAHAT SA INYONG bibig

The computer program

Machine code

Programming Languages

- Natural languages (e.g., English) are
  - Ambiguous.
  - Overly general.
  - Difficult to translate (in fact, a big research field!).
- Solution: create specialized *programming languages* that are good at specifying instructions to computers.
- Examples: Ada, Algol, BASIC, C, C++, C#, CLOS, D, Eiffel, Fortran, F#, Haskell, and so many more!

The Java Programming Language

- Created in 1990 by Sun Microsystems
  - Alternative to C/C++
- Object-oriented language
  - "(Almost) Everything is an object"
- Platform independent
  - Java programs run on Windows, Mac, or Linux
- Most popular language out there
  - See the TIOBE programming index

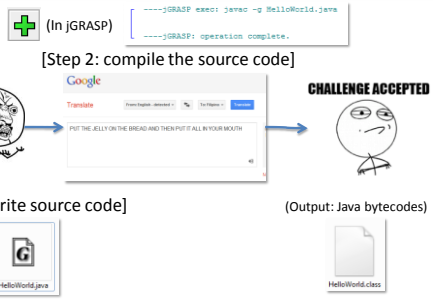
Our first computer program.

(jGRASP demonstration)

Our First Computer Program

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


The Compilation Pipeline



Running your compiled program

- Step 3: tell the computer to run your program

The structure of a Java Program

A *class* named HelloWorld. For now, classes are containers for our programs.

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

A *statement* that prints "Hello World" to the console. Statements are commands to be executed by your program.

A *method* named main. main is a special method because this is where the program begins execution.

Syntax, syntax, syntax

- *Syntax*: the rules to form legal programs

Class template

```
public class <name> {
  <method>
  <method>
  …
  <method>
}
```

Method template

```
public static void <name>(…) {
  <statement> ;
  …
  <statement> ;
}
```

Syntax errors

- *Syntax errors* occur when you violate a syntax rule.

```
1  public class HelloWorld {
2     public static void main(String[] args) {
3        System.out.println("Hello World!");
4        System.out.println()
5        System.out.println("I like pie");
6        System.out.println("I like pie a lot");
7     }
8  }
```

Line numbers. Not part of the program.

```
----jGRASP exec: javac -g HelloWorld.java
HelloWorld.java:4: ';' expected
                System.out.println()
                                    ^
1 error
----jGRASP wedge2: exit code for process is 1.
----jGRASP: operation complete.
```

Need to fix syntax errors before your program can be compiled!

Naming and Identifiers

- In Java, names have several restrictions.
  - *Must not* be a reserved *keyword* (e.g., `public`, `class`).
    - Many others, see page 20 of the text for the complete list.
  - *Must* start with a letter, _ (underscore), or $.
  - Otherwise can contain, letters, numbers, _, or $.
  - E.g., `HelloWorld42` is valid, `12HelloWorld` is not.
- Java is a *case-sensitive* language.
  - E.g., `main` and `Main` are different names.
- The name of a class must match its containing Java file
  - E.g., the `HelloWorld` class is found in *HelloWorld.java*

System.out.println

- Prints to the screen or *console*.
  - How to pronounce it: *print*-lin
- Two uses:
  - System.out.println("msg") prints msg to the screen along with a newline.
  - System.out.println() prints a newline.

Strings

- A *string* is a piece of data that represents text.
  - E.g., to be println'ed to the console.
- Syntax: "text goes here"
  - Text surrounded by quotes.
- Restrictions:
  - Cannot span over multiple lines
    "this is a
    syntax error!"
  - Cannot contain a quotation mark
    - "when does this string begin" and end?"

Escape sequences

- *Escape sequences* allow us to write special characters in strings.
  - Quotation mark: \"
  - Tab character: \t
  - Newline character: \n
  - Backslash: \\
- Ex: `System.out.println("\\\\\"\"\"\"\\");`
  - prints \\"""""\ to the console!