Chapter 1: Introduction

CS 121

Department of Computer Science
College of Engineering
Boise State University

August 22, 2016

Topics

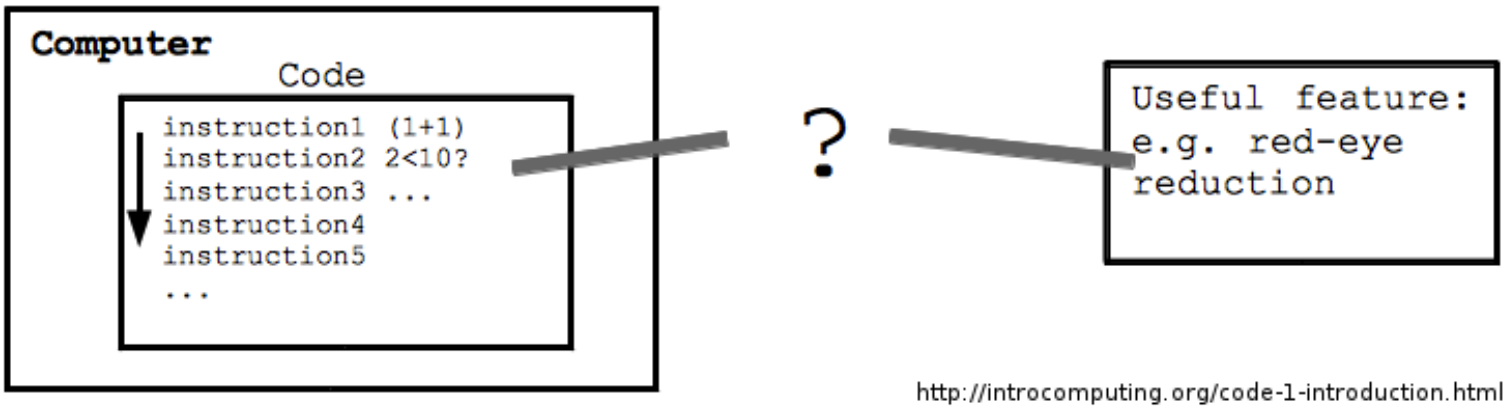
- What is Computer Science?
- Program development in Java
  - Writing and running our first program.
  - Breaking (and fixing) our first program.

What is a Computer?

> *Computers are incredibly fast, accurate, and stupid. Human beings are incredibly slow, inaccurate, and brilliant. Together they are powerful beyond imagination.*
>
> –Albert Einstein

- A computer is a mechanical tool that requires a human to tell it what to do.
- This is where Computer Science comes in...

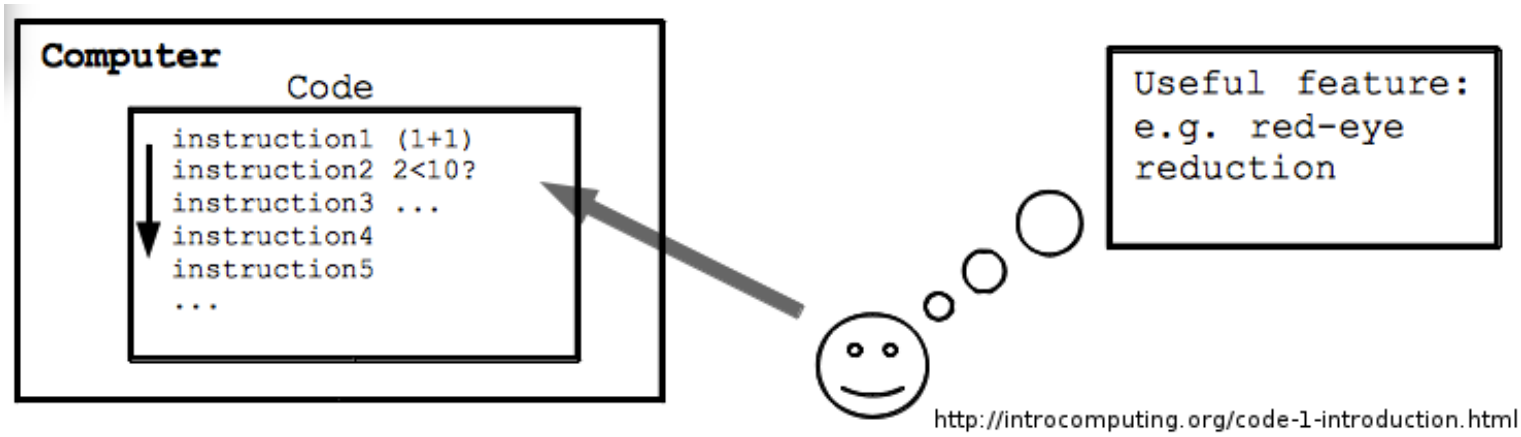
How does a computer work?



http://introcomputing.org/code-1-introduction.html

- Computer is driven by code.
- Code is made of simple, mechanical instructions.
- Computer runs series of instructions.
- Machine simply follows directions, so how does it do so much??

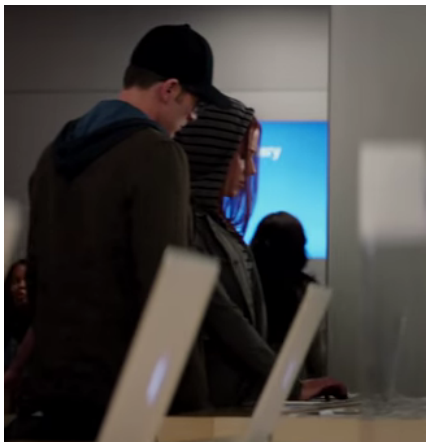
Humans to the rescue



- People come up with cool ideas (requirements),
- think through the solutions (design),
- break them down into simple instructions for the computer (algorithms),
- write code for the computer (implementation), and
- test them to make sure the computer is behaving properly (testing).

Computer Science - not just about writing code.

- Teaches people how to break down and solve problems algorithmically.
- Gives people the tools to solve problems in *all* human endeavors.
  - Farming, finance, entertainment, engineering, art, music etc.
- Perhaps someday, every company will become a software company!
- Even superheroes need computer science.



In-Class Exercise

- Write down an example of a "real-world" problem that could be broken down and solved algorithmically.
- Then write the algorithm as a short sequence of steps.
- Example: Make grilled cheese sandwich
  1. Obtain bread
  2. Obtain cheese
  3. Obtain butter
  4. Put butter on bread
  5. Obtain pan and heat it up on stove
  6. Put cheese between bread
  7. Cook on pan until golden brown

Program Development

- Program development (or implementation) can be broken into the following tasks.
  - Programming – writing the program.
  - Compilation – translating the program into a form the computer can execute (machine-language).
  - Execution – running the program.
  - Debugging – investigating and fixing various types of errors that occur.

Program Development

- Programming – writing the program.
- Compilation – translating the program into a form the computer can execute (machine-language).
- Execution – running the program.
- Debugging – investigating and fixing various types of errors that occur.

Programming Languages

- A program controls a computer. Programs are also called *applications* or *apps*.
- A programming language specifies the words and symbols we can use to write a program.
- A programming language employs a set of rules that dictate how the words and symbols can be put together to form valid program statements.

High-Level Programming Language

- We write programs in high-level languages.
  - Understandable by humans.
  - Machine independent (runs on different computer models).
  - Needs to be translated by a compiler.
- In this class, we will write programs in a high-level language called Java.

Program Development in Java

- There are many high-level computer programming languages.
  - Java
  - C/C++
  - C#
  - Python
  - JavaScript
- So, why Java?
  - Concepts you learn can be applied to other programming languages.
  - One of the world's most popular programming languages.
  - At the core of many applications you use every day.

Program Development in Java

- Java is freely available.
- The Java Development Kit (JDK) is for developing Java programs.
- The Java Runtime Environment (JRE) is for running Java programs.
- Latest major version is Java 1.8. You would want to install the JDK. It includes the JRE.

Basic Programming in Java

- Example: Lincoln.java
- Example: Lincoln2.java and Lincoln3.java

Let's break it down.

- Class header
- Class body
- Comments
- Method header
- Method body
- Identifiers
- White space

In-Class Exercise

What do you think the following code does?

```
public class HelloClass
{
    public static void main(String[] args)
    {
        // Let's say hello.
        System.out.println("Hello CS 121!");
    }
}
```

1. Prints "Hello CS 121!" to the screen and terminates.
2. Prints "Hello CS 121!" to the screen and keeps running.
3. Prints "Let's say hello. Hello CS 121!" to the screen and terminates.
4. Does nothing useful. There is an error.

In-Class Exercise

What do you think the following code does?

```
public class HelloClass
{
    public static void main(String[] args)
    {
        // Let's say hello.
        System.out.println("Hello CS 121!"));
    }
}
```

1. Prints "Hello CS 121!" to the screen and terminates.
2. Prints "Hello CS 121!" to the screen and keeps running.
3. Prints "Let's say hello. Hello CS 121!" to the screen and terminates.
4. Does nothing useful. There is an error.

Program Development

- Programming – writing the program.
- Compilation – translating the program into a form the computer can execute (machine-language).
- Execution – running the program.
- Debugging – investigating and fixing various types of errors that occur.

Machine Language

- Computer hardware can only execute a machine language program.
  - Tells hardware what to do.
  - Sequence of binary numbers (e.g. 0001 0010 0011 0100...)
  - Machine langauge is specific to hardware brand/model.
  - Difficult for most humans to compose, edit, and understand.
- We don't write programs in machine language, so how do we tell it what to do?

Compilation

- A program must be *translated* into machine language before it can be executed.
- A compiler translates a high-level language into another programming language.
- A compiler is just a computer program, a programmer's tool.

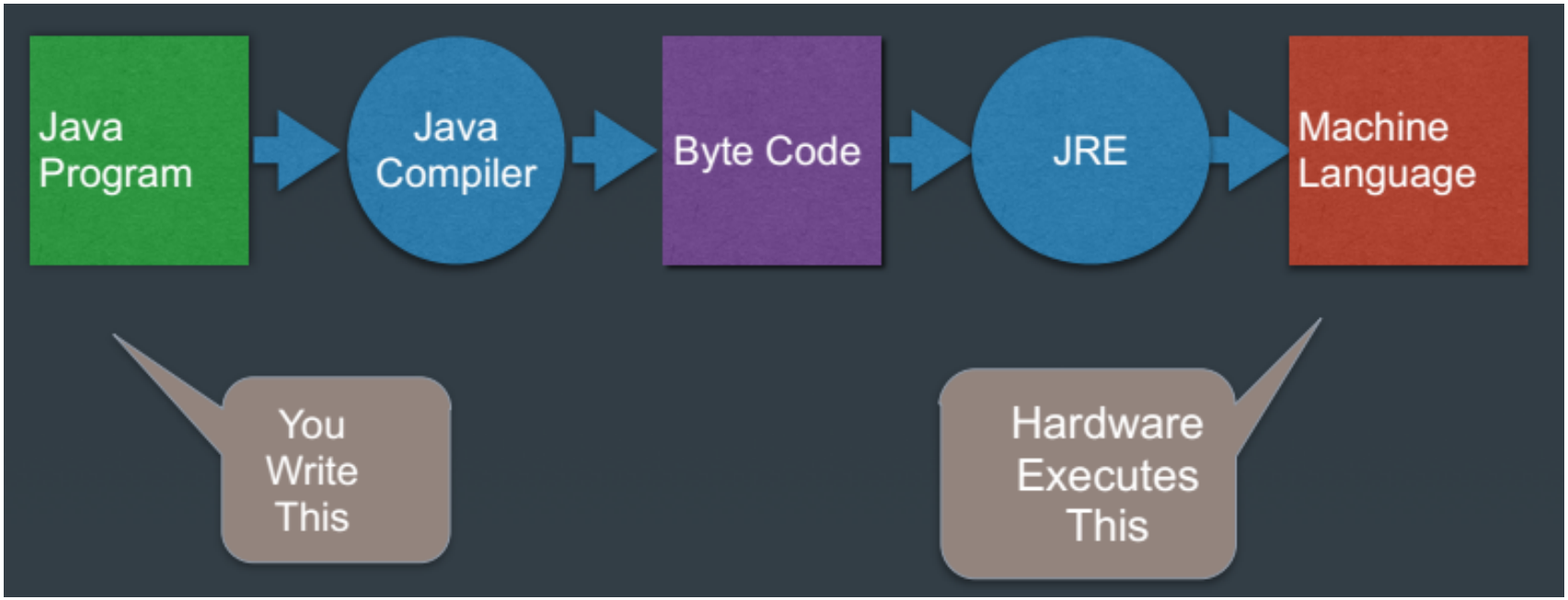
The Java Compiler

- The Java compiler is included with the Java Development Kit (JDK).

```
javac Lincoln.java
```

- The Java compiler translates Java code to byte-code.
  - Machine-independent intermediate language.
  - Difficult for humans to understand.
  - Java Runtime Environment (JRE) (an interpreter) translates byte-code to the hardware's machine language.
- Places byte-code in separate `.class` file.

The Big Picture



In-Class Exercise

What do you think the following code does?

```
public class HelloClass
{
    public static void main(String[] args)
    {
    }
}
```

1. Prints "HelloClass" to the screen and terminates.
2. Prints nothing to the screen and terminates.
3. Does not compile.
4. Compiles but does not run.

Program Development

- Programming – writing the program.
- Compilation – translating the program into a form the computer can execute (machine-language).
- Execution – running the program.
- Debugging – investigating and fixing various types of errors that occur.

Execution

- After a program is compiled, you may run it from the command-line.

```
java Lincoln
```

Program Development

- Programming – writing the program.
- Compilation – translating the program into a form the computer can execute (machine-language).
- Execution – running the program.
- Debugging – investigating and fixing various types of errors that occur.

Syntax and Semantics

- Every language has its own set of syntax rules and semantics.
- Syntax – Grammar. The rules for what kinds of words/symbols you can put together and in what order.
  - English: **"Horse monkey candy cane"** (invalid syntax)
  - Math: **=10+7>** (invalid syntax)
- Semantics – Meaning of the statement.
  - English: **"The puddle runs Greenland forcefully"** (valid syntax, invalid semantics)
  - Math: **1=10** (valid syntax, invalid semantics)

Debugging

> *Debugging is twice as hard as writing the code in the first place. Therefore, if you write the code as cleverly as possible, you are, by definition, not smart enough to debug it.*
>
> –Brian Kernighan

- Where you will probably spend most of your time.
- Three major types of errors.
  - compile-time: Incorrect syntax or other basic problems
  - run-time: A problem during program execution that causes it to terminate abnormally. (e.g. divide by zero, etc.)
  - logical: The program runs but produces incorrect results. (e.g. using the wrong formula, producing the wrong output, etc.)

In-class Exercise

Ex 1.6. Categorize each of the following as a compile-time error, run-time error, or logical error.

- multiplying two numbers when you meant to add them
- dividing by zero
- forgetting a semicolon at the end of a code statement
- spelling a word wrong in the output
- producing inaccurate results
- typing a { when you should have typed a (

Example

- Enough talk...
- Let's just break the program and see what happens.
  - compile-time: remove semi-colon, remove quote, misspell class.
  - run-time: change 'main' to 'Main', divide by zero
  - logical: want it to print "Hello World!", but it prints "Hello world"

Review

- What is Computer Science?
- Program development in Java
  - Programming
  - Compilation
  - Execution
  - Debugging

Exercises

- Access Blackboard.
- Activate your Piazza account.
- Read Chapter 1. Focus on sections 1.1, 1.3 and 1.4 and skim the rest.
- **Recommended Homework**
  - Exercises: EX 1.1-1.6.
  - Projects: PP 1.4, 1.7.
- Browse Chapter 2.