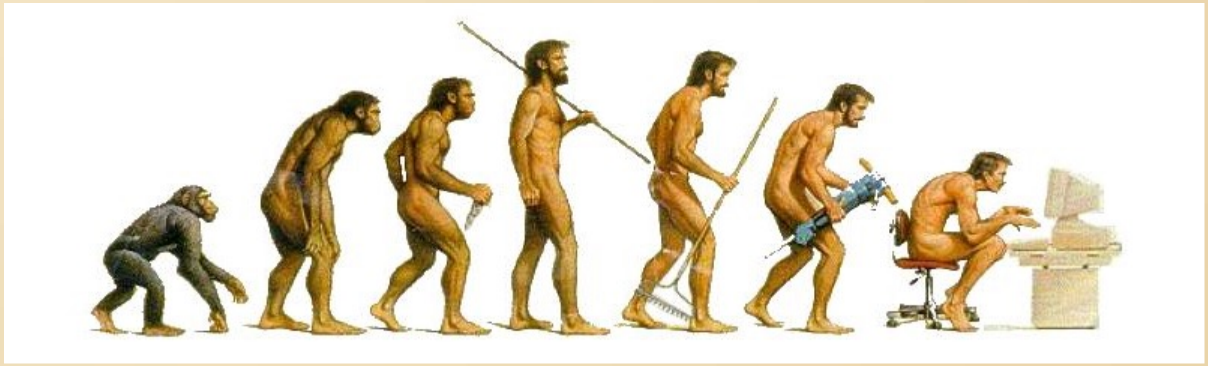
Lecture 1-1

Introduction to Computer Science:
Course Overview



Course objectives

The course will give you …

- Basic exposure to Computer Science
- Basic programming skills

In addition, you will …

- Sharpen your analytic skills
- Appreciate clarity and elegance
- Develop a taste for beauty in science and engineering
- Learn how to learn and develop.

Course requirements

Attend:

- Two weekly lectures (שיעור)
- One weekly recitation (תרגיל)
- One weekly workshop (סדנה)

Submit:

- A weekly homework assignment (שיעורי בית)

Course Website (“Moodle”)

Contents:

- Lecture slides + code
- HW documents + code
- Course announcements
- HW submission / grades

Important:

The only formal and abiding channel of communications between the course team and the students is **the course web site**

What is said by the course staff about HW, exams, requirements, grading, deadlines – is **tentative**. What is written in the web site is **final**.

Q&A Forums

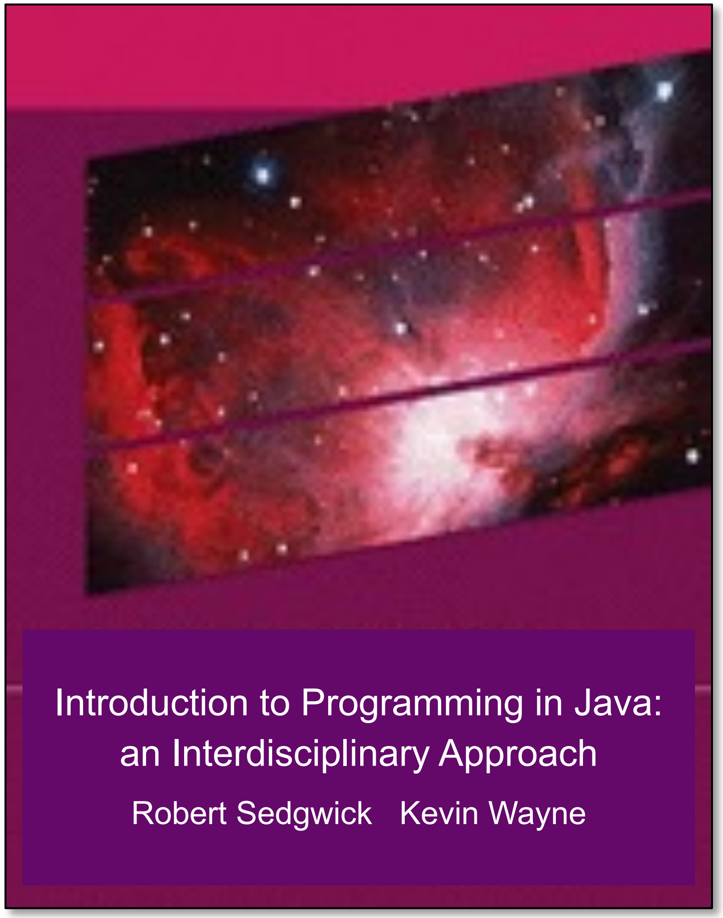
How to ask questions

- Find the relevant forum (by week)
- Read existing posts
- If there is no relevant answer, post a question
- You'll get an answer within a few hours, from a TA or from another student
- Avoid clutter, keep the channel clean
- Use English (whatever is your language level – let's practice!)

Individual work

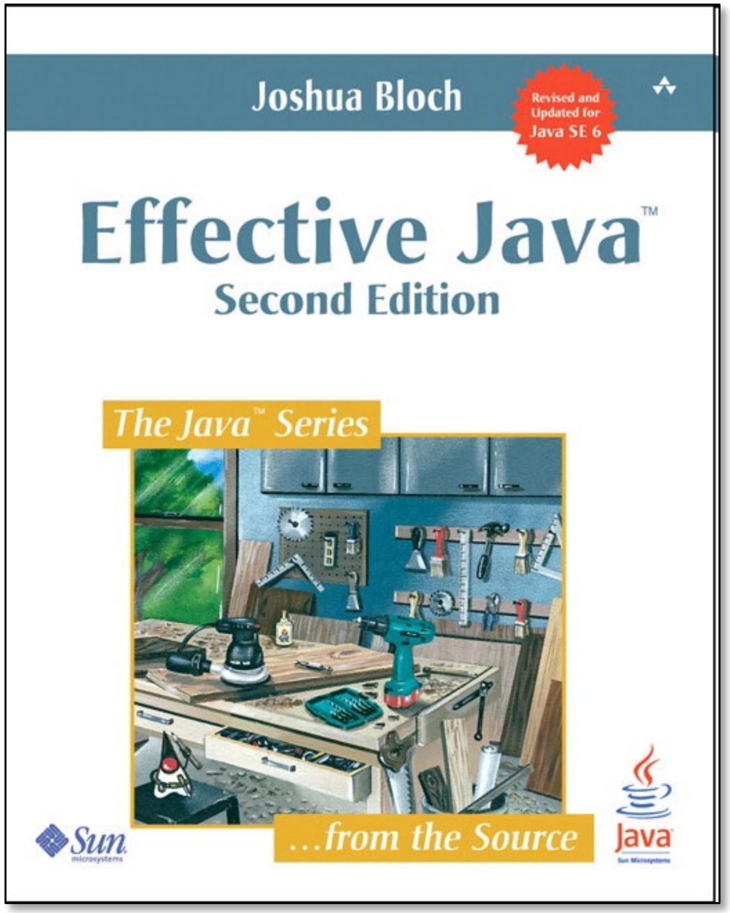
Read the course honor code

Textbooks


Introduction to Programming in Java:
an Interdisciplinary Approach
Robert Sedgwick Kevin Wayne

Recommended textbook

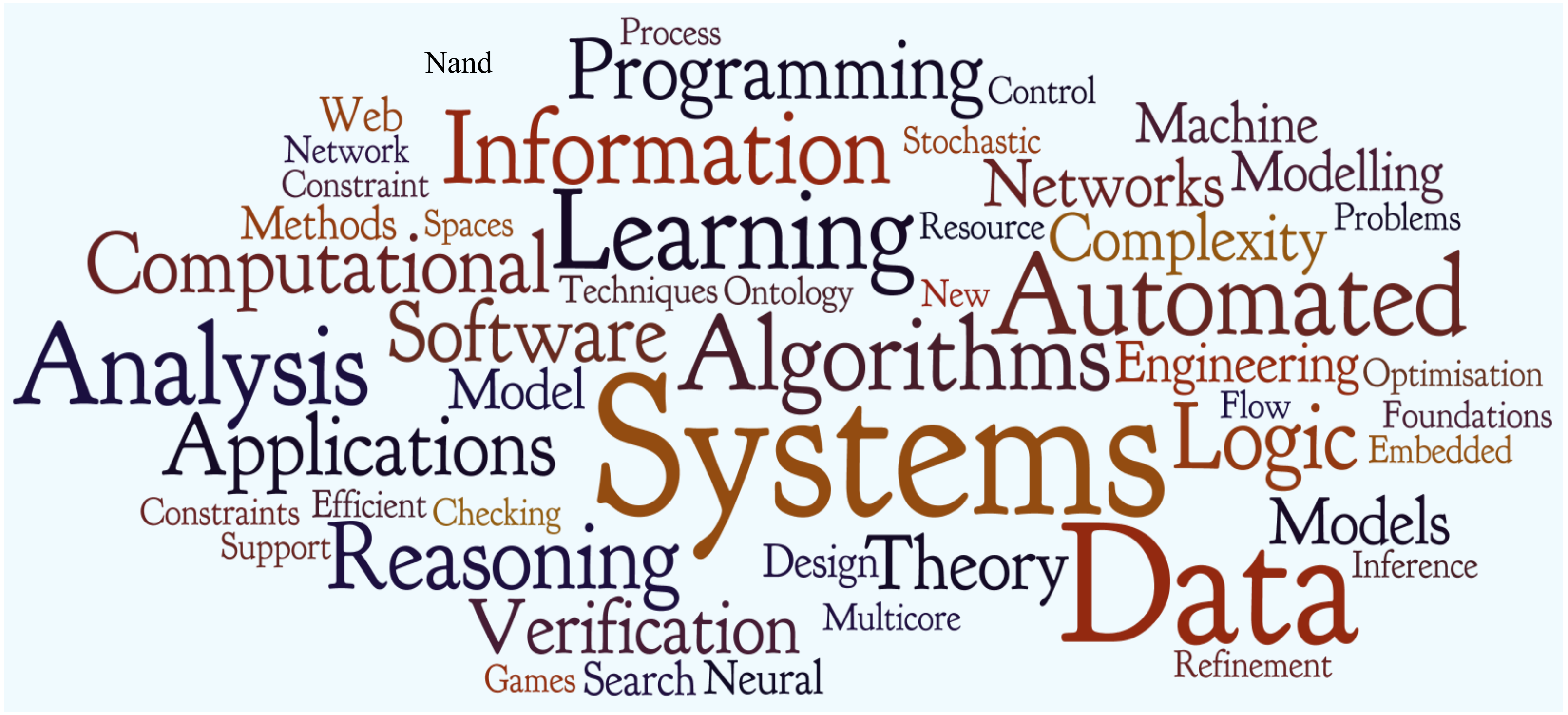
(any edition is fine)

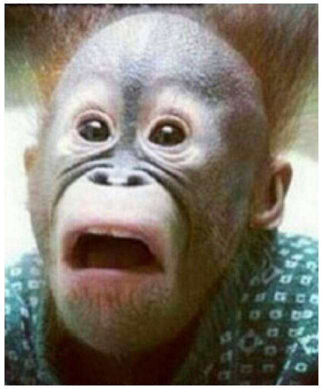


Students with Java experience

(a classic)

Computer science is about …





Computer science is about …

Theory

- Algorithms
- Data structures
- Complexity
- . . .

Systems

- Operating Systems
- Compilers
- Networks
- . . .

Applications

- Image processing
- Molecular biology
- Artificial intelligence

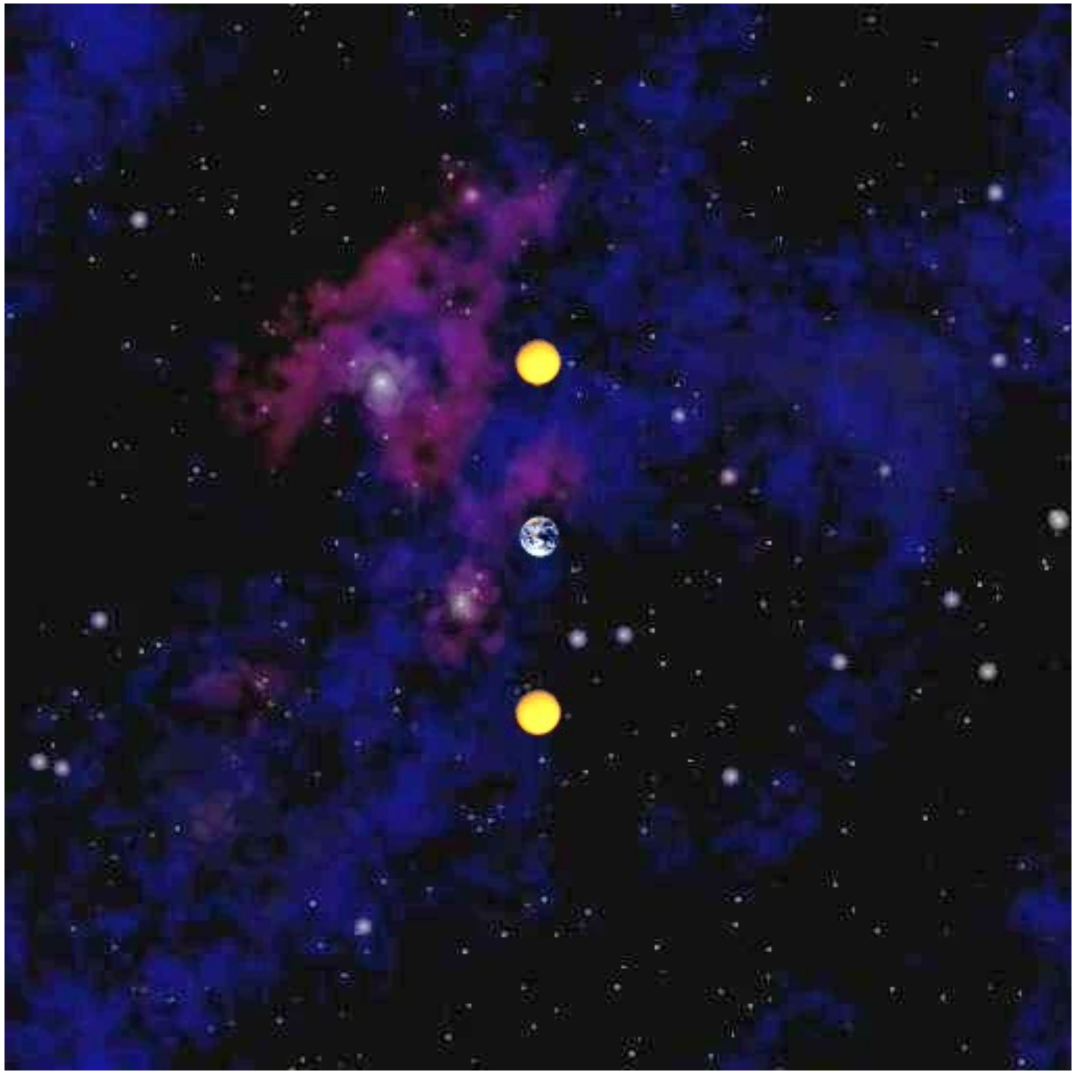
. . .

Common themes

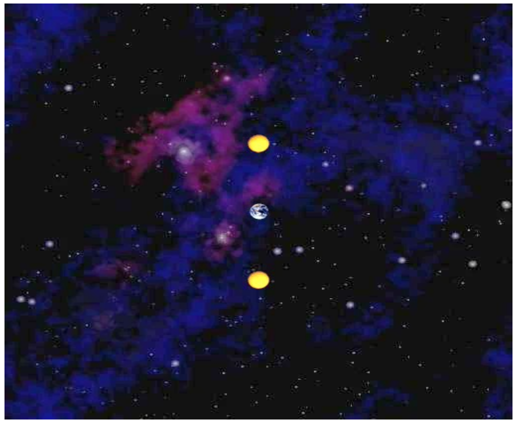
- Programming
- Efficiency
- Beauty (acquired taste)

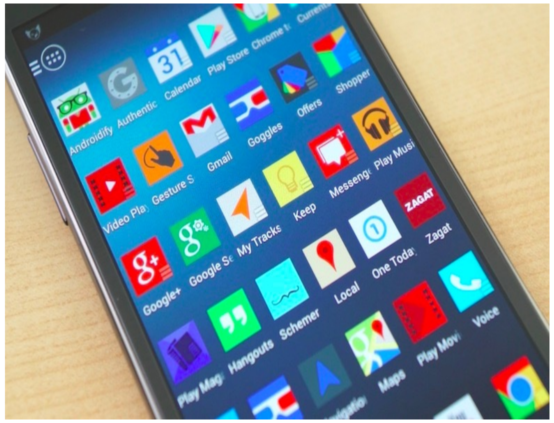
A taste of programming

Simulate the motion of *N* heavenly bodies,
subject to Newton's laws of motion and gravity



Programming is about . . .





<u>Problem solving</u>

Imaging, simulation, medical systems, e-commerce, social networks, chatbots, ...

<u>Packaged software</u>

Apps, games, tools, ...

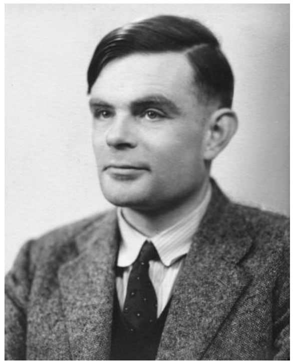
<u>System software</u>

Operating systems, compilers, networks, cloud, security, ...

Can programming help solve any possible problem / need?

- No, computers and programming have inherent limitations
- Stay tuned.

Ada Lovelace



Alan Turing

Programming is about . . .

Writing code that is:

- Correct
- Efficient

And is:

- Easy to test
- Easy to understand
- Easy to maintain
- Easy to extend
- Pleasure to work with

"Instead of imagining that our main task as programmers is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do." – Donald Knuth

Programming is about . . .

<u>Writing code that is:</u>

- Correct
- Efficient

<u>And is:</u>

- Easy to test
- Easy to understand
- Easy to maintain
- Easy to extend
- Pleasure to work with

"Instead of imagining that our main task as programmers is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do." – Donald Knuth

How to make a peanut butter sandwich...

Programming languages

Which language to learn?

- Java
- Python
- C
- Haskell

  …

Programming languages

Why Java?

- Widely used
- Widely available
- Powerful, elegant, multi-platform
- Addresses numerous needs
- Excellent software development tools
- Our School legacy

Java Applications

- Android
- Google docs
- Netflix
- Spotify
- LinkedIn
- Amazon

. . .

Programming languages

Facts of life

- There is no perfect programming language
- We need to choose *some* language

Our approach

- Teach a **subset** of Java
- Develop **general programming skills** applicable to any software development task
- Build a foundation that allows learning any other language quickly

It's not about the language!

"There are two kinds of programming languages: those that people always complain about, and those that nobody uses."

– Bjarne Stroustrup (father of C++)

Java program example

Task: Print the numbers 0 to 5

Algorithm

```
i = 0
while (i < 6)
    print i
    i = i + 1
```

Pseudocode

Java implementation

```java
public class PrintSomeNumbers {
    public static void main(String[] args) {
        // Declares an integer variable and sets it to 0
        int i = 0;
        while (i < 6) {
            // Prints i, and increments it
            System.out.println(i);
            i = i + 1;
        }
        System.out.println("Done");
    }
}
```

Java code

Java syntax elements (first approximation)

"Words":

- reserved words
- identifiers (user-defined)

Literals (constants):

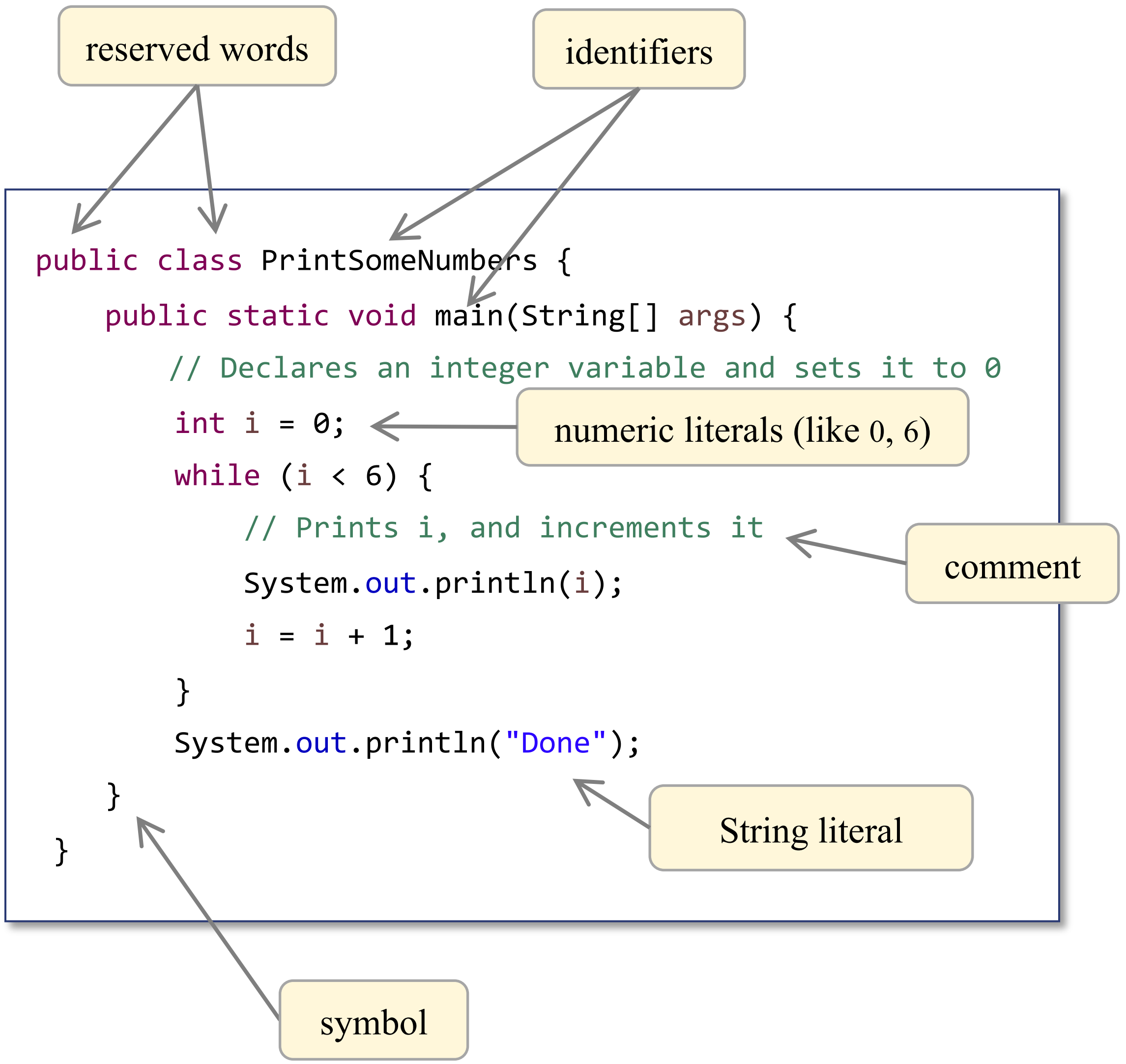
- numbers
- strings

Symbols:

( ) [ ] { } , . ; + - * / …

White space

- comments
- indentation
- colors.



```
public class PrintSomeNumbers {
    public static void main(String[] args) {
        // Declares an integer variable and sets it to 0
        int i = 0;
        while (i < 6) {
            // Prints i, and increments it
            System.out.println(i);
            i = i + 1;
        }
        System.out.println("Done");
    }
}
```

Java reserved words

`abstract`	`continue`	`for`	`new`	`switch`
`assert`	`default`	`goto`	`package`	`synchronized`
`boolean`	`do`	`if`	`private`	`this`
`break`	`double`	`implements`	`protected`	`throw`
`byte`	`else`	`import`	`public`	`throws`
`case`	`enum`	`instanceof`	`return`	`transient`
`catch`	`extends`	`int`	`short`	`try`
`char`	`final`	`interface`	`static`	`void`
`class`	`finally`	`long`	`strict`	`volatile`
`const`	`float`	`native`	`super`	`while`

Java program structure

```
public class PrintSomeNumbers {
    public static void main(String[] args) {
        // Declares an integer variable and sets it to 0
        int i = 0;
        while (i < 6) {
            // Prints i, and increments it
            System.out.println(i);
            i = i + 1;
        }
        System.out.println("Done");
    }
}
```

method, function

class

<u>Program</u> (loosely defined): a collection of one or more classes

<u>Class</u>: a collection of one or more methods / functions

<u>Method / function:</u> a sequence of one or more statements

<u>In this course:</u>

- We'll start writing programs that consist of one class and one method ("`main`")
- Later we will write classes that consist of several methods
- Later we will write programs that consist of several classes.

Java program structure

```
public class PrintSomeNumbers {
   public static void main(String[] args) {
       // Declares an integer variable and sets it to 0
       int i = 0;
       while (i < 6) {
           // Prints i, and increments it
           System.out.println(i);
           i = i + 1;
       }
       System.out.println("Done");
   }
}
```

Same functionality

```
public class PrintSomeNumbers {public static void main(String[] args){int i=0;while
(i<6){System.out.println(i);i=i+1;}System.out.println("Done");}}
```

White space

Comments, indentation, colors (ignored by the compiler)

Purpose

Used to make programs readable

Program readability and clarity are as important as program correctness (maybe more)!

"Any fool can write code that a computer understands.
Good programmers write code that humans understand" – Martin Fowler

Java program structure

```
public class PrintSomeNumbers {
   public static void main(String[] args) {
       // Declares an integer variable and sets it to 0
       int i = 0;
       while (i < 6) {
           // Prints i, and increments it
           System.out.println(i);
           i = i + 1;
       }
       System.out.println("Done");
   }
}
```

Aspects of any language

Semantics (what you want to say / do):
meaning / intention

Syntax (how to say it):
The rules of the language:
vocabulary and grammar

Style (how *well* you say it):
Critically important

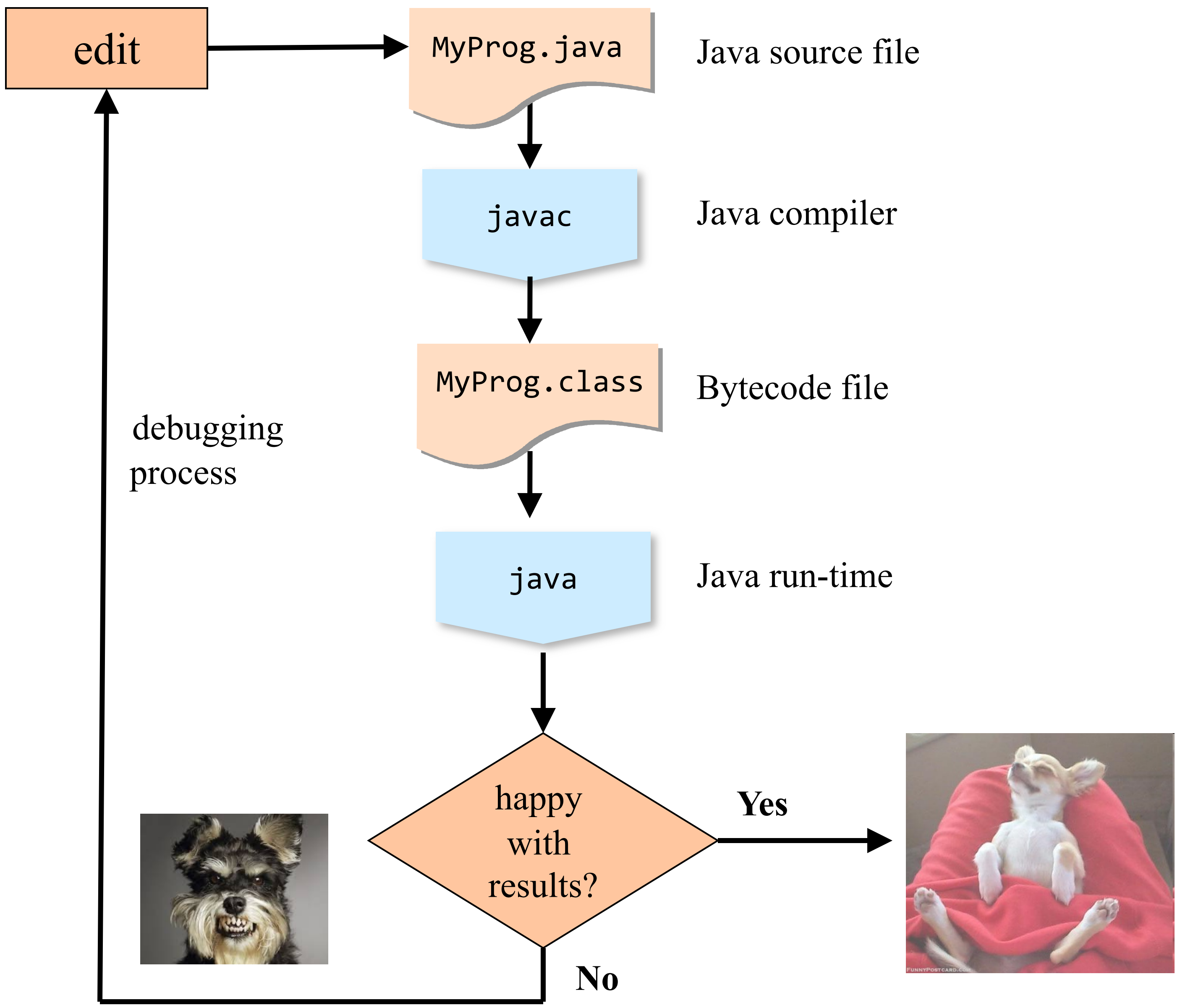
Natural languages (Hebrew, English, ...)

- Allow breaking syntax rules
- Occasionally there is more than one meaning to a sentence

Programming languages

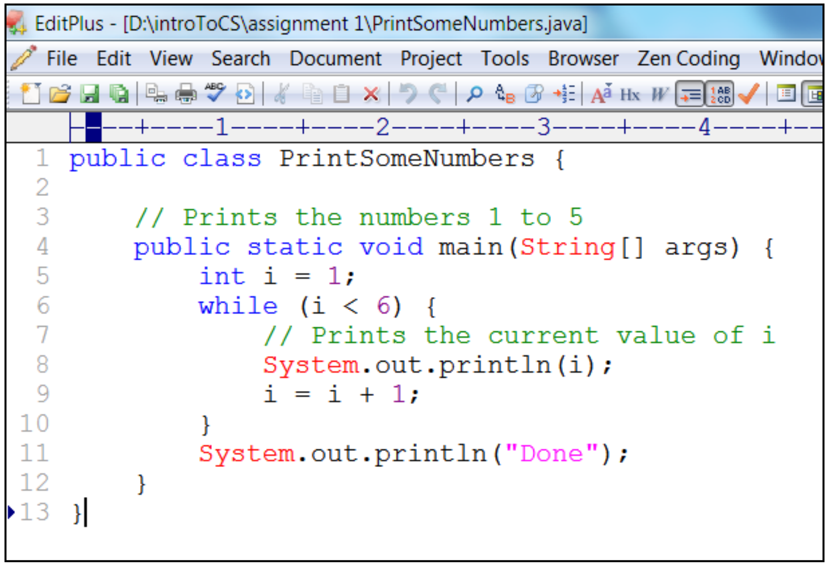
- Syntax is sacred
- Only one semantic interpretation: No ambiguity

Program development



Program development: Command / terminal level

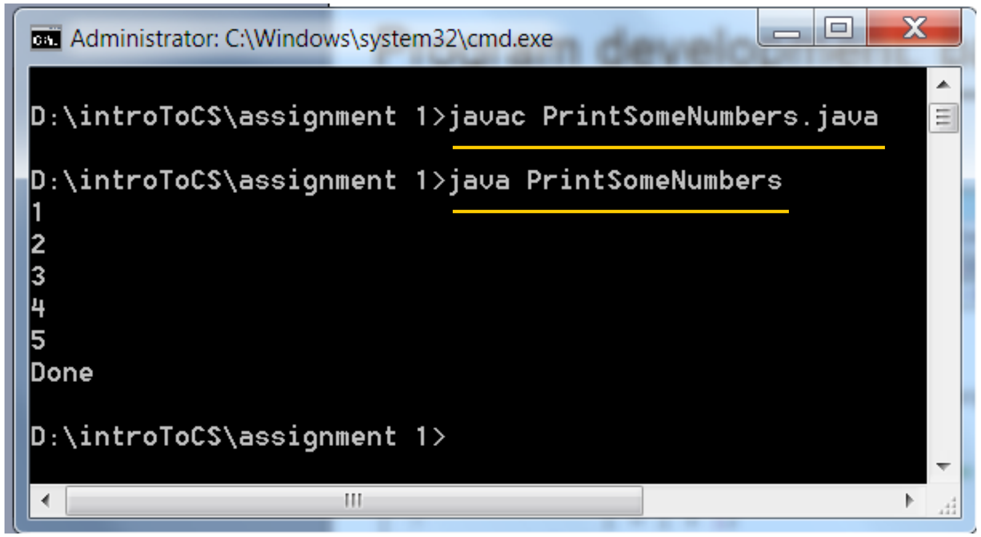
Edit (in this example: the file `PrintSomeNumbers.java`):


EditPlus - [D:\introToCS\assignment 1\PrintSomeNumbers.java]
File Edit View Search Document Project Tools Browser Zen Coding Windo

```java
public class PrintSomeNumbers {

    // Prints the numbers 1 to 5
    public static void main(String[] args) {
        int i = 1;
        while (i < 6) {
            // Prints the current value of i
            System.out.println(i);
            i = i + 1;
        }
        System.out.println("Done");
    }
}
```

Compile and execute:



```
D:\introToCS\assignment 1>javac PrintSomeNumbers.java

D:\introToCS\assignment 1>java PrintSomeNumbers
1
2
3
4
5
Done

D:\introToCS\assignment 1>
```

<u>Debugging</u>

0. Run / execute the program

1. Observe the program's execution

2. Figure out what's wrong

3. Use the editor to fix the code

4. Goto step 0.

Program development: Integrated Development Environments

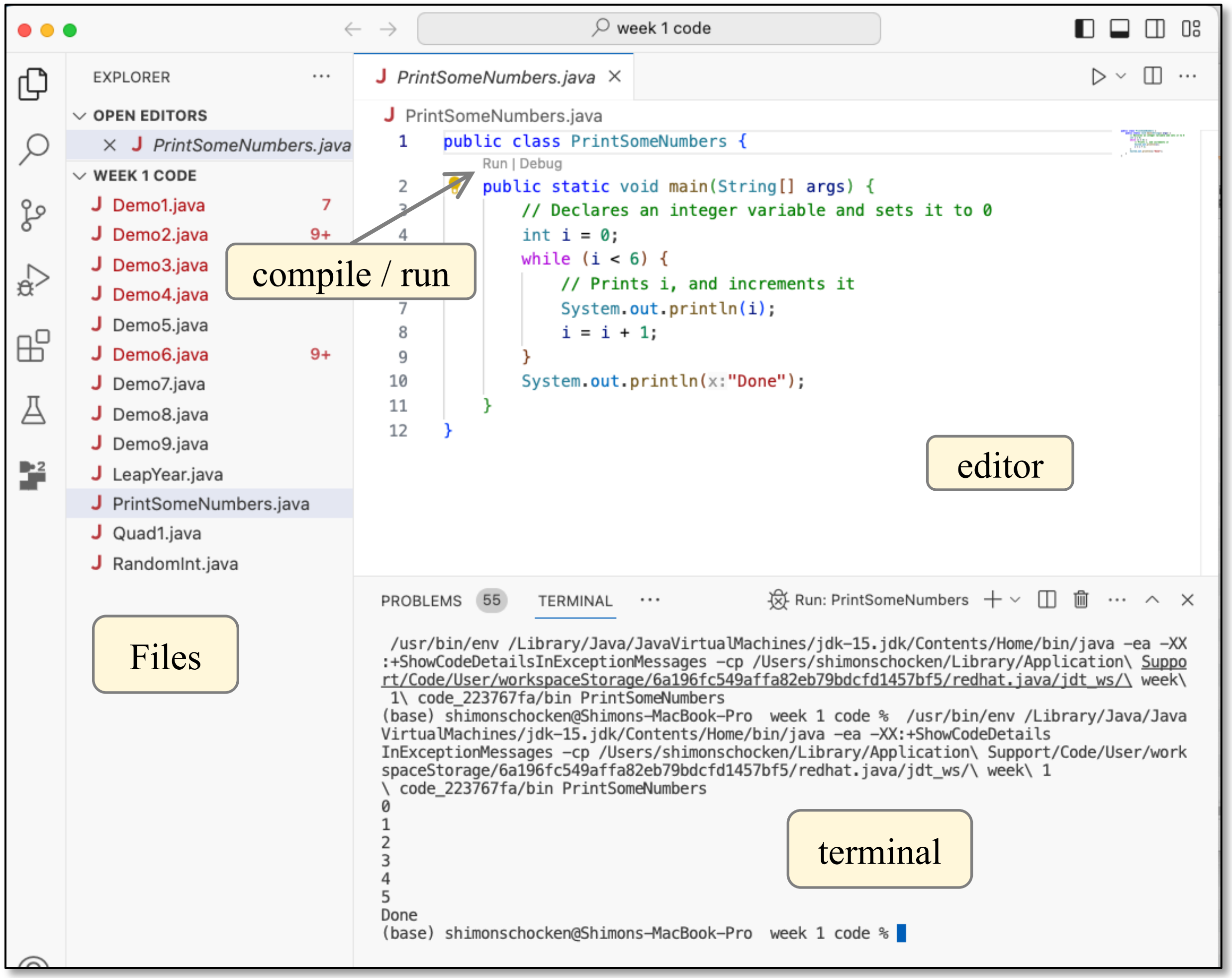
IDE: a software package featuring:

- editor (language-specific)
- compiler
- debugger
- . . .
- various dev goodies

Popular IDEs:

- VS Code
- InteliJ
- Eclipse
- NetBeans

. . .



```java
public class PrintSomeNumbers {
    public static void main(String[] args) {
        // Declares an integer variable and sets it to 0
        int i = 0;
        while (i < 6) {
            // Prints i, and increments it
            System.out.println(i);
            i = i + 1;
        }
        System.out.println(x:"Done");
    }
}
```

```
 /usr/bin/env /Library/Java/JavaVirtualMachines/jdk-15.jdk/Contents/Home/bin/java -ea -XX
:+ShowCodeDetailsInExceptionMessages -cp /Users/shimonschocken/Library/Application\ Suppo
rt/Code/User/workspaceStorage/6a196fc549affa82eb79bdcfd1457bf5/redhat.java/jdt_ws/\ week\
 1\ code_223767fa/bin PrintSomeNumbers
(base) shimonschocken@Shimons-MacBook-Pro  week 1 code %  /usr/bin/env /Library/Java/Java
VirtualMachines/jdk-15.jdk/Contents/Home/bin/java -ea -XX:+ShowCodeDetails
InExceptionMessages -cp /Users/shimonschocken/Library/Application\ Support/Code/User/work
spaceStorage/6a196fc549affa82eb79bdcfd1457bf5/redhat.java/jdt_ws/\ week\ 1
\ code_223767fa/bin PrintSomeNumbers
0
1
2
3
4
5
Done
(base) shimonschocken@Shimons-MacBook-Pro  week 1 code % 
```

Debugging

That's what you'll do most of the semester

Error types:

- **Compile-time errors:** mostly syntax violations; detected by the compiler
- **Run-time errors:** the program passes compilation, runs, but crashes
- **Logical errors:**
  - The program runs, doing something unexpected
  - The program runs, but should be improved

Mistakes are the portal of discovery

(James Joyce)

Anything that can possibly go wrong, will

(Murphy's Law)

Things to do

Getting started

- Visit the course website
- Read the honor code
- Install Java on your computer
  (*Safe Landing Tutorial*)

Homework 1

- Play with an existing Java program
- Experience debugging
- Write a few simple programs
- Further instructions: see the course website.