

→

ADMINISTRATION

- **What do you need?**
 - Computer (Windows, Mac, or Linux)
 - Books – see the syllabus
 - Prior programming experience is not required

ADMINISTRATION

- **Teaching Assistant - Assistants are available on-campus and/or online to answer quick questions.**
 - Class TA: Rishabh Sood – sood.r@northeastern.edu

ADMINISTRATION

- **Expectations**
 - Full attendance
 - Auditing is not allowed
 - Complete assignments – there will be several
 - What is the Final Project?

THE LECTURE

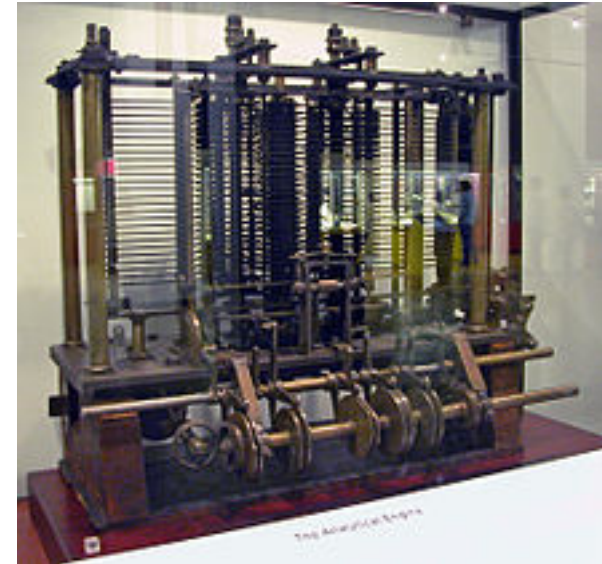
- **Why Java**
- **Installing Java / Other tools**
- **Coding**
 - Hello World
- **Variables**
- **Control Statements**
 - For-demo
- **UML**
 - Use Cases

WHY JAVA?

FIRST, A LOOK AT COMPUTERS

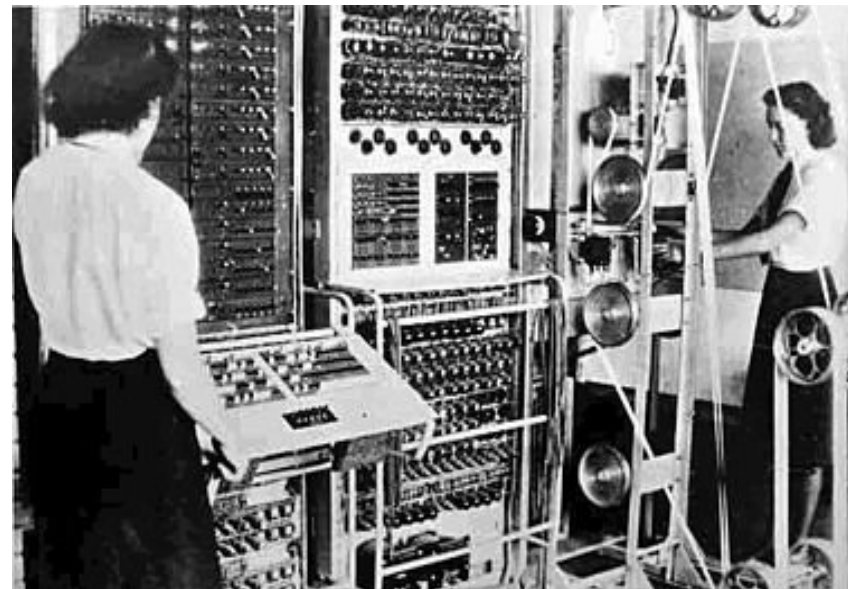
1837 - Charles Babbage Analytical Engine

https://en.wikipedia.org/wiki/File:AnalyticalMachine_Babbage_London.jpg

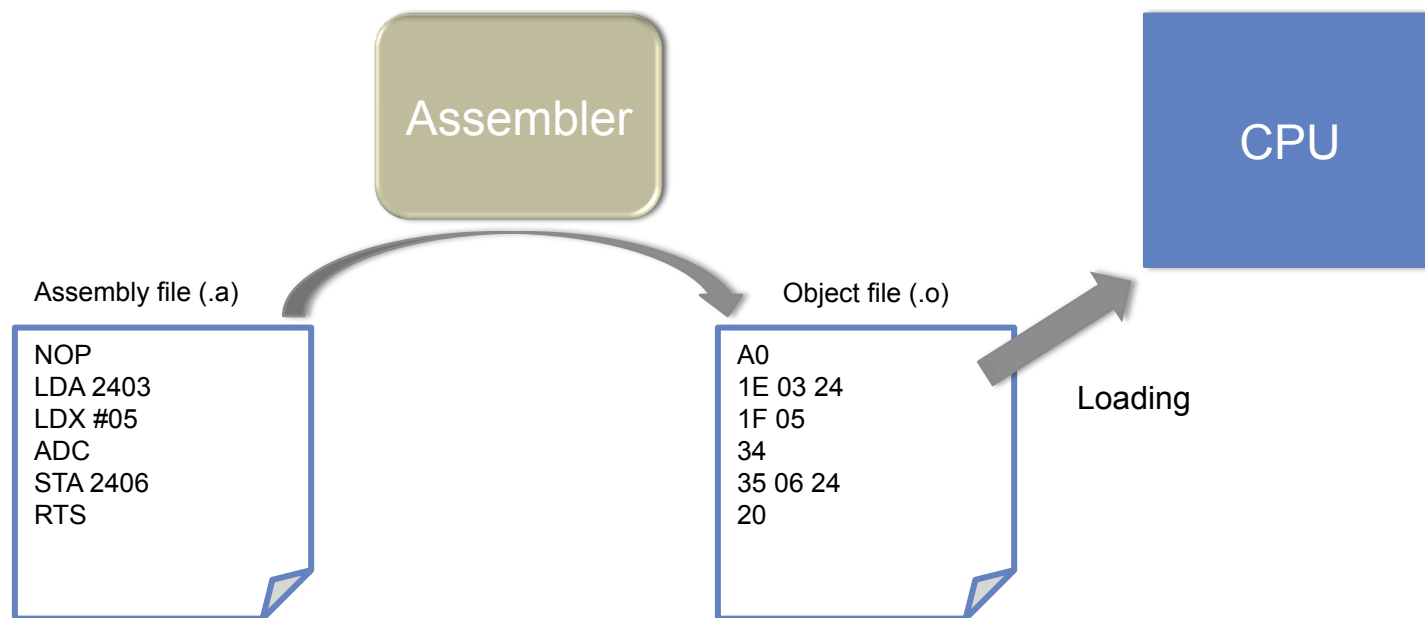


1944 – Colossus Mark 2

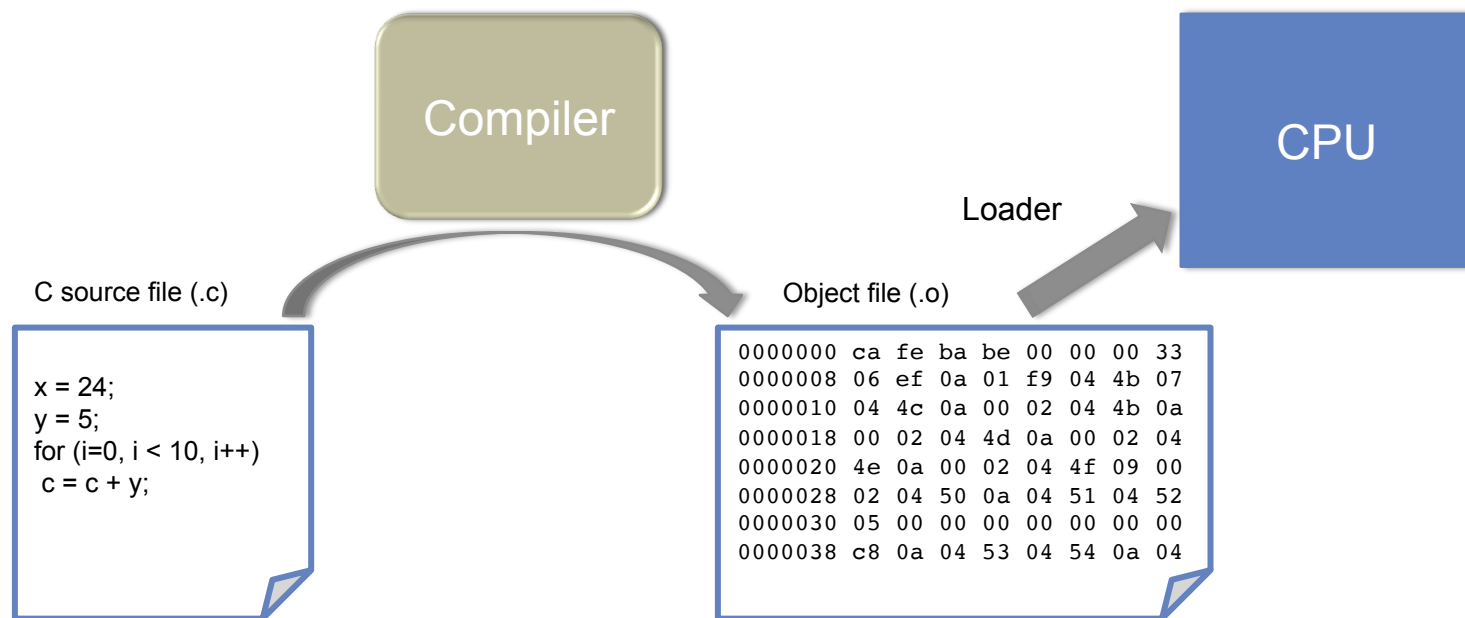
<https://en.wikipedia.org/wiki/File:Colossus.jpg>



EARLY PROGRAMMING MODEL



HIGH LEVEL LANGUAGE PROGRAMMING MODEL



RISE OF THE MAINFRAME

IBM System 370



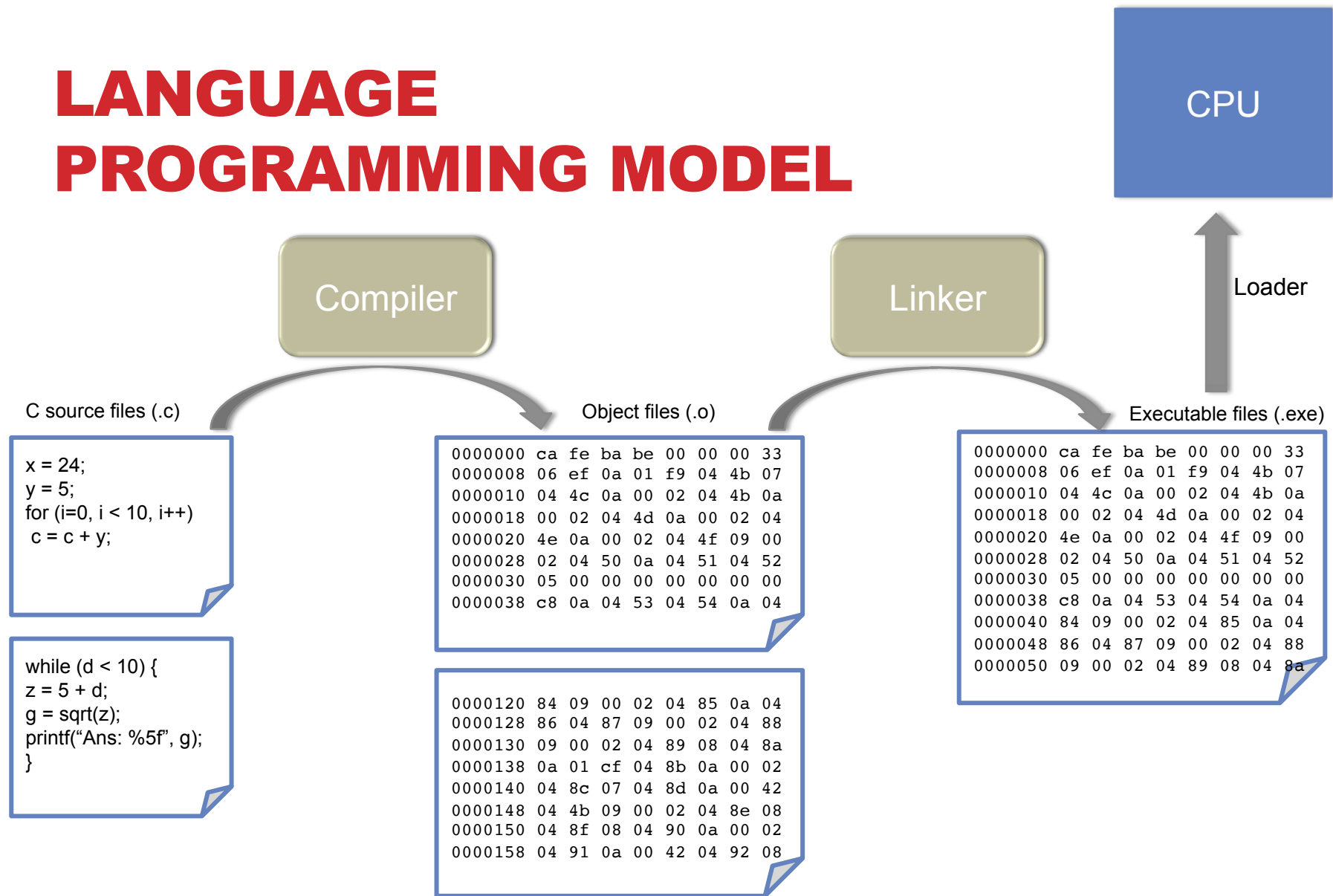
https://commons.wikimedia.org/wiki/File%3AIBM_System_370-145_und_Bandlaufwerke_2401.png
By Oliver.obi (Own work) [CC BY-SA 3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>)], via Wikimedia Commons



https://commons.wikimedia.org/wiki/File%3APanel_370-145.png
By Oliver.obi (Own work) via Wikimedia Commons

- **High level language made coding easier (COBOL, FORTRAN, SAS, CICS)**
- **Everyone shared the same computer via Time Slicing**

LANGUAGE PROGRAMMING MODEL

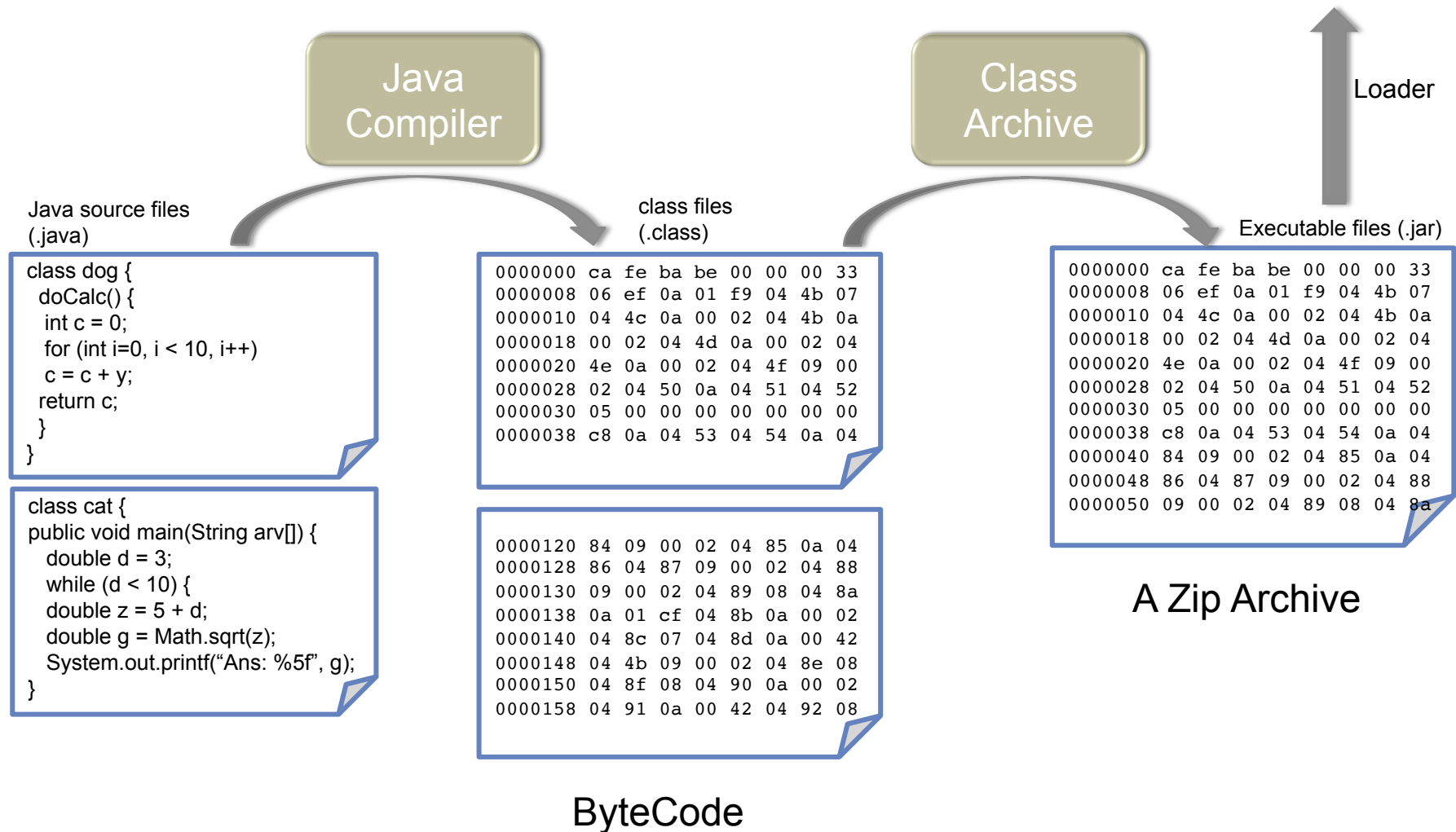
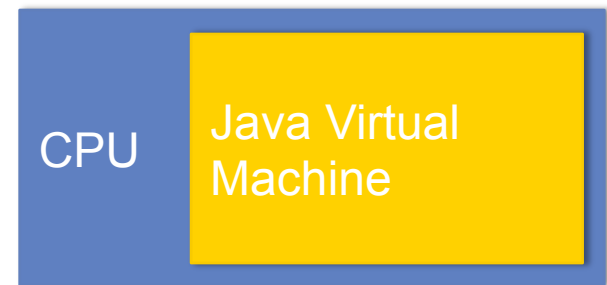


A MODEL PROBLEM

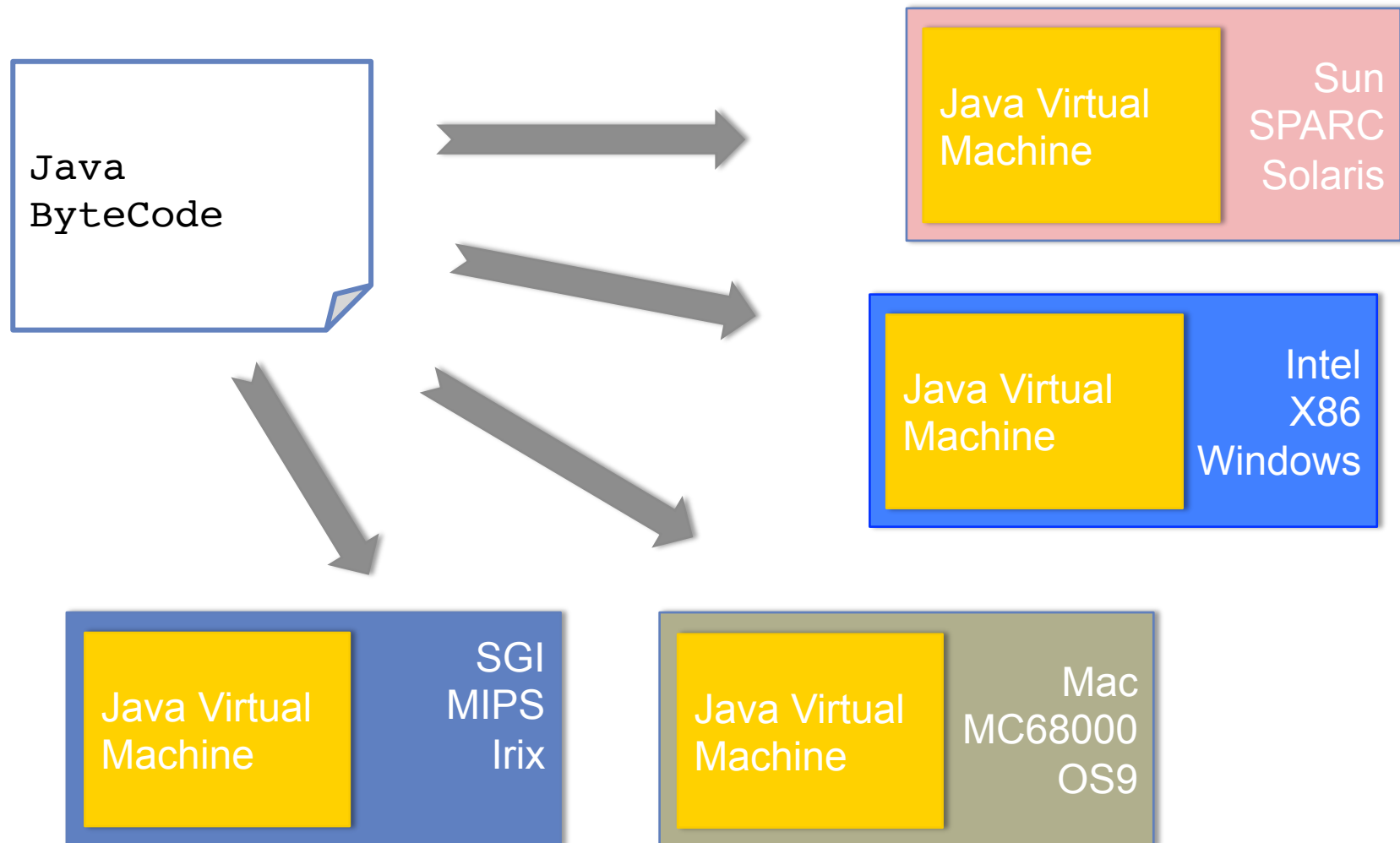
Problem: Too many processors, operating systems, and architectures – each platform requires customization – and more work to test, package and distribute

- **Chipsets**
 - Intel x86, Motorola 68K, WD 65C816
 - AMD Opteron, Sun SPARC, IBM PowerPC
 - ARM Atom
- **Operating Systems (Minicomputers and Personal Computers)**
 - Windows DOS/NT, Apple MacOS
 - IBM AIX, HP HP/UX, Sun Solaris, SGI Irix
 - Linux

JAVA LANGUAGE PROGRAMMING MODEL



JAVA PORTABILITY : A SINGLE FILE SOLUTION



THEN THE INTERNET 'HAPPENED'



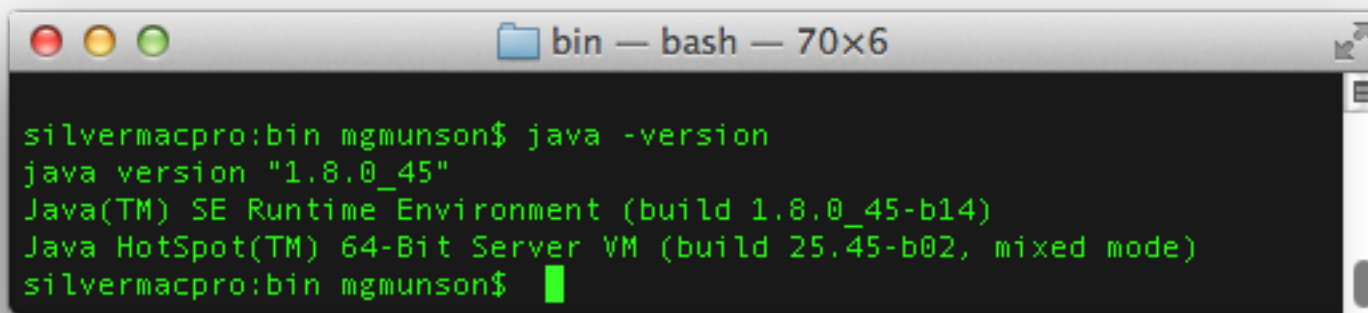
"Internet connections" by Wyn.junior - With help of Mark Zuckerberg Previously published: No. Licensed under CC BY-SA 3.0 via Wikipedia – https://en.wikipedia.org/wiki/File:Internet_connections.jpg#/media/File:Internet_connections.jpg

- **Java code was small, compressed, and fit into a single file**
- **Via 'applets', Java could run inside your Internet browser**

INSTALLING JAVA

INSTALLING JAVA

- Go to the Sun Java Developer site to install the latest 64-bit version of the Java Software Developer's Kit (JDK)
- OpenJDK: <https://jdk.java.net/14/> or Oracle: <https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>
- Run the installer, and use the default options
- Check your installation by running the 'java' command from a terminal window

A screenshot of a macOS terminal window titled 'bin — bash — 70x6'. The terminal shows the command 'java -version' being executed, with the following output: 'silvermacpro:bin mgmunson\$ java -version', 'java version "1.8.0_45"', 'Java(TM) SE Runtime Environment (build 1.8.0_45-b14)', 'Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)', and 'silvermacpro:bin mgmunson\$'.

```
silvermacpro:bin mgmunson$ java -version
java version "1.8.0_45"
Java(TM) SE Runtime Environment (build 1.8.0_45-b14)
Java HotSpot(TM) 64-Bit Server VM (build 25.45-b02, mixed mode)
silvermacpro:bin mgmunson$
```

INSTALLING JAVA (CONT.)

The screenshot shows the Oracle website's 'Java SE Development Kit 12 - Downloads' page. It includes a navigation menu, a search bar, and a sidebar with links to various Java releases. The main content area features a 'Java SE Development Kit 12 Downloads' section with a thank-you message and a link to the JDK 12.0.2 checksum. Below this is an 'Important Oracle JDK License Update' section, followed by a table of download links for Linux and macOS.

Product / File Description	File Size	Download
Linux	155.14 MB	jdk-12.0.2_linux-x64_bin.deb
Linux	162.79 MB	jdk-12.0.2_linux-x64_bin.rpm

Oracle JDK 14

The screenshot shows the jdk.java.net website's 'JDK 12.0.2 General-Availability Release' page. It includes a navigation menu, a search bar, and a sidebar with links to various Java releases. The main content area features a 'JDK 12.0.2 General-Availability Release' section with a thank-you message and a link to the JDK 12.0.2 checksum. Below this is an 'Important Oracle JDK License Update' section, followed by a table of download links for Linux and macOS.

Product / File Description	File Size	Download
Linux	155.14 MB	jdk-12.0.2_linux-x64_bin.deb
Linux	162.79 MB	jdk-12.0.2_linux-x64_bin.rpm

OpenJDK 14

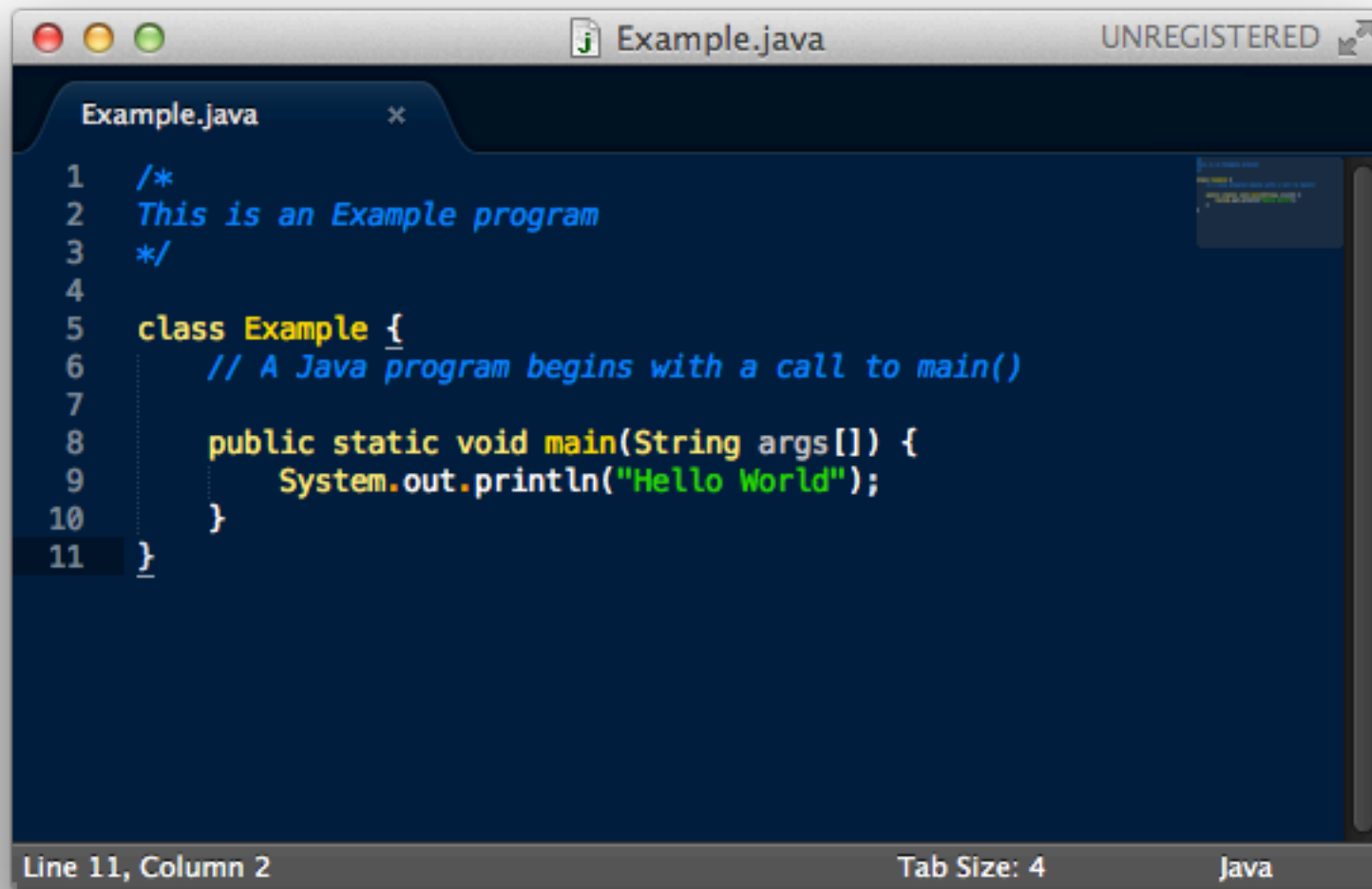
OTHER TOOLS

- **Windows users:**
 - Install Cygwin to emulate a Unix-style console
- **Version control:**
 - Install a git client
 - On Windows: TortiseGit allows git commands to be used directly from the GUI

A SIMPLE JAVA PROGRAM

LET'S CODE

HELLO WORLD



The image shows a screenshot of the Sublime Text 2 IDE. The window title is 'Example.java' and it is marked as 'UNREGISTERED'. The code is written in Java and is as follows:

```
1  /*
2  This is an Example program
3  */
4
5  class Example {
6      // A Java program begins with a call to main()
7
8      public static void main(String args[]) {
9          System.out.println("Hello World");
10     }
11 }
```

The status bar at the bottom indicates 'Line 11, Column 2', 'Tab Size: 4', and 'Java'.

SOME CLASS RULES

- **Java is Object Oriented, so procedures must live within a 'class'**
- **Class names start with an upper case letter (i.e. Example)**
- **Each class is stored in a file with the same name**
 - Example.java contains the Example class definition
- **Each class may contain a start point called 'main'**

BUILD IT

Use 'dir' command with
Windows DOS
Use 'ls' with OSX and
Linux

```
> ls
```

```
Example.java
```

```
> java -version
```

```
java 11.0.1 2018-10-16 LTS
```

```
Java(TM) SE Runtime Environment 18.9(build 11.0.1+13-LTS)
```

```
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed  
mode)
```

```
> javac Example.java
```

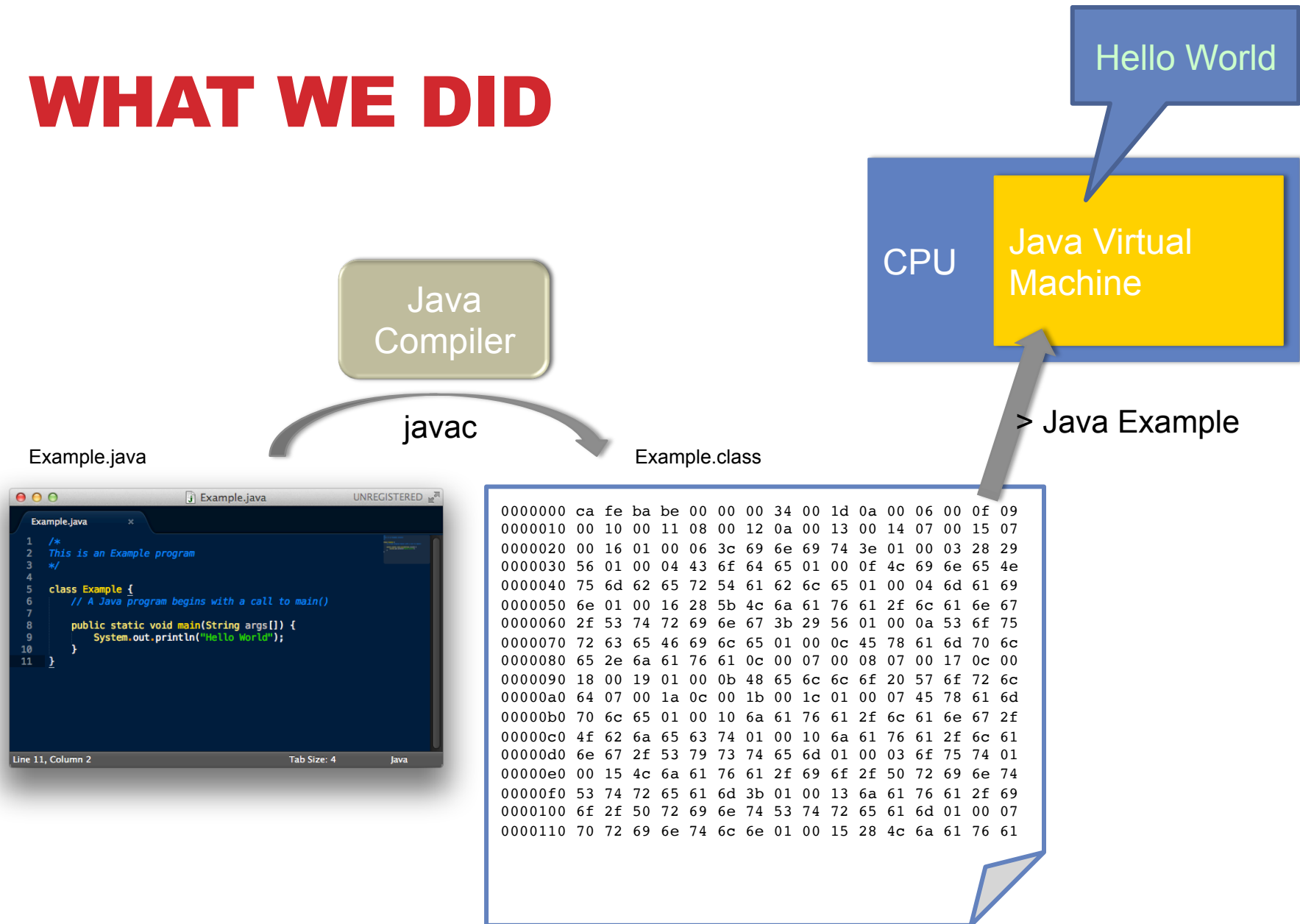
```
> ls
```

```
Example.class  Example.java
```

```
> java Example
```

```
Hello World
```

WHAT WE DID



VARIABLES

- **A variable is a named location where a program stores values**
 - Each class defines areas to store information
- **Each variable has three pieces of information**
 - *Name*: What this variable is called
 - *Type*: The kind of data that is stored (integers, floats, characters, etc.)
 - *Value*: The value at any period in time

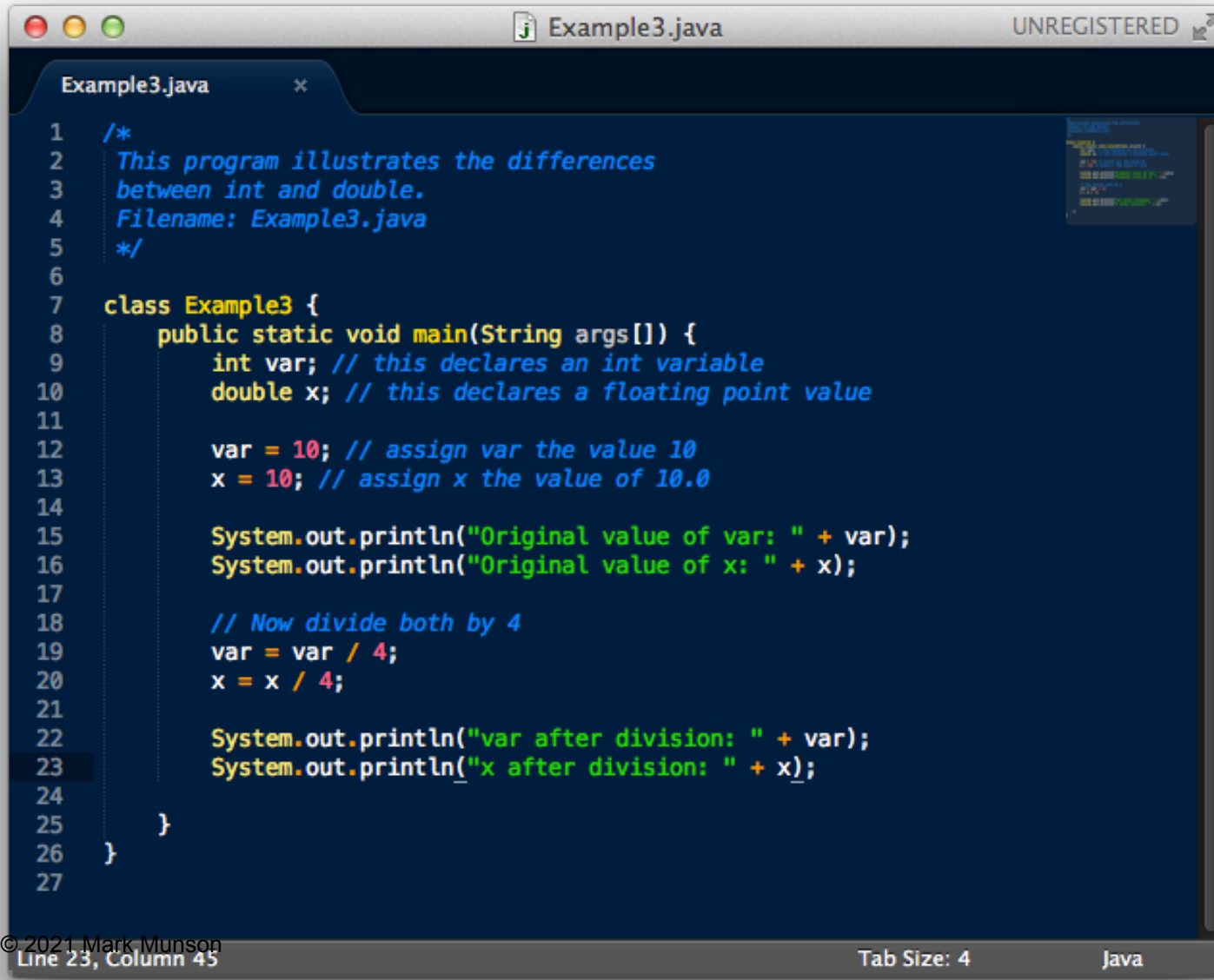
int
score 8

String
firstName "Mark"

VARIABLE NAMING

- May start with any letter of the alphabet, an underscore, or a dollar sign
 - valid: **boxCount3** invalid: **23box**
- Other characters may be a number, letter, an underscore or a dollar sign
 - Valid: **a\$** or **_counter** or **box_count_3**
- Upper and lower case characters are different
 - MyVal is different than myVal
- None of the fifty keywords (defined in Table 1-1) may be used in a variable name
- Programming style guides may be used to promote code consistency among developers.
 - For example, most programmers start variables with a lowercase letter and capitalize each word:
capitalizeEachWordButTheFirst

EXAMPLE 3



```
1  /*
2   This program illustrates the differences
3   between int and double.
4   Filename: Example3.java
5   */
6
7  class Example3 {
8      public static void main(String args[]) {
9          int var; // this declares an int variable
10         double x; // this declares a floating point value
11
12         var = 10; // assign var the value 10
13         x = 10; // assign x the value of 10.0
14
15         System.out.println("Original value of var: " + var);
16         System.out.println("Original value of x: " + x);
17
18         // Now divide both by 4
19         var = var / 4;
20         x = x / 4;
21
22         System.out.println("var after division: " + var);
23         System.out.println("x after division: " + x);
24     }
25 }
26
27
```

Line 23, Column 45

Tab Size: 4

Java

CONTROL STATEMENTS

- The **if** Statement
 - Allows code to selectively execute parts of a program

```
if (condition) statement;  
or  
if (condition) { statements }
```

Examples

```
if (c < 10) System.out.println("c is less than ten");  
if (c == 5) {  
    System.out.println("c is equal to five");  
    c = c+1;  
    System.out.println("c plus one is : " + c);  
}
```

CONTROL STATEMENTS (CONT.)

- The **for** Loop

- Repeatedly execute a sequence of code

for (initialization; condition; iteration) statement;

or

for (initialization; condition; iteration) {statements}

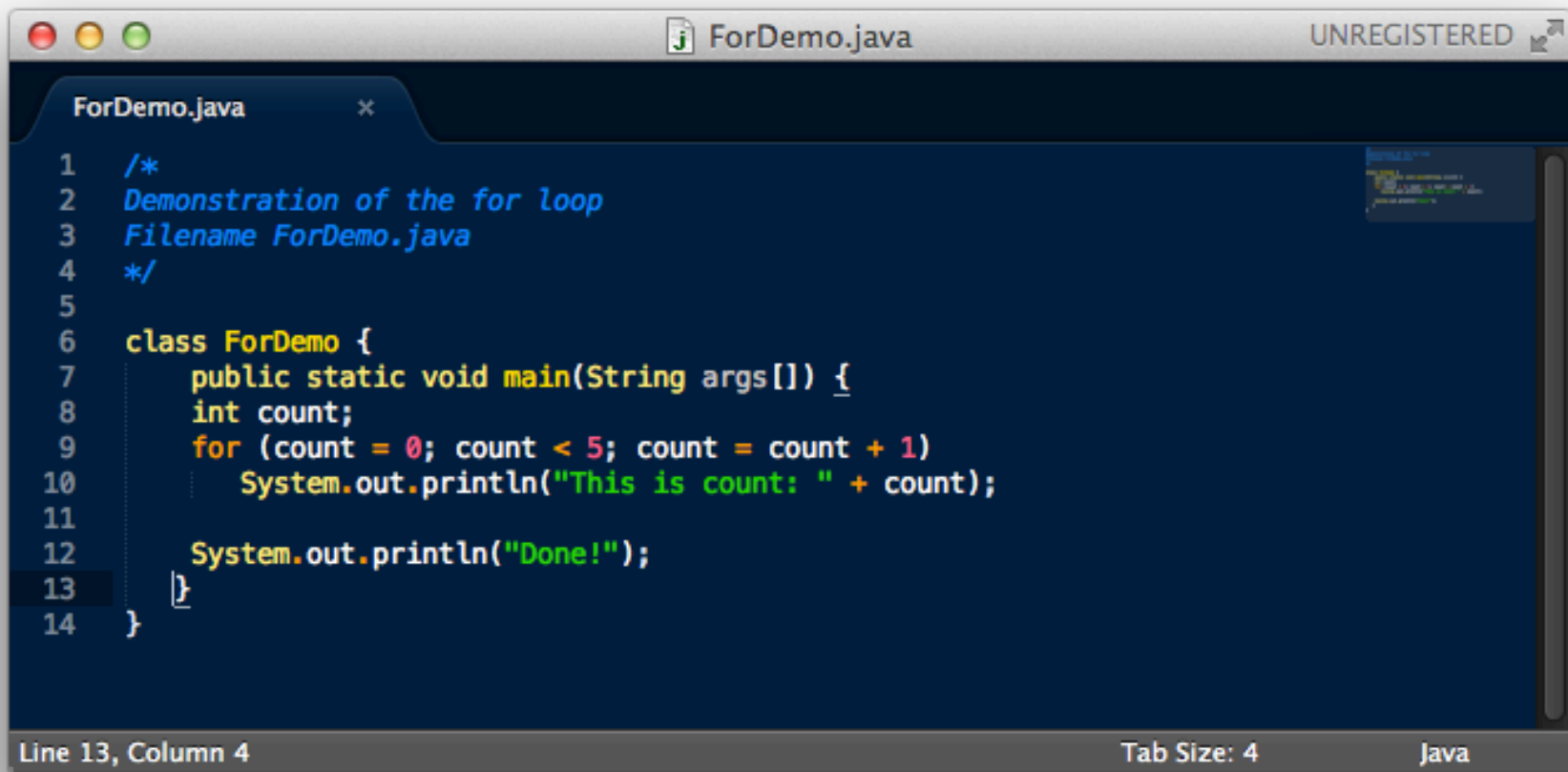
- Example

```
int count;
```

```
for (count = 0; count < 5; count = count + 1)
```

```
    System.out.println("This is count " + count);
```

EXAMPLE FORDEMO



The screenshot shows a Java IDE window with the title 'ForDemo.java' and 'UNREGISTERED' in the top right corner. The code is as follows:

```
1  /*
2  Demonstration of the for loop
3  Filename ForDemo.java
4  */
5
6  class ForDemo {
7      public static void main(String args[]) {
8          int count;
9          for (count = 0; count < 5; count = count + 1)
10             System.out.println("This is count: " + count);
11
12         System.out.println("Done!");
13     }
14 }
```

The status bar at the bottom indicates 'Line 13, Column 4', 'Tab Size: 4', and 'Java'.

NEXT SESSION

- **JABG: Read Ch. 2 and 3 (and 1 if you haven't already)**
- **Bring your laptop to class with installed software**
 - Java installed
 - Eclipse (or Netbeans) installed
 - Git client installed
 - TortiseGit (optional), or other Git client
 - Windows users
 - Cygwin (optional)
 - Simple Code editor – sublimetext.com (optional)
- **Helpful guides are available on the Course Materials section**