Introduction

Luis Tarrataca
luis.tarrataca@gmail.com

CEFET-RJ

1. First things first

2. First things first

First things first

First things first

By now most of you have had classes with me:

- If not, then brace yourself...

We will follow my preferred class organization:

What is my preferred class organization? Remember?

First things first

What is my preferred class organization? Remember?

- 2 / 3 exams per semester
  - Total evaluation weight: 50%
- Laboratory work (50%)
  - Total evaluation weight: 50%
  - Laboratories on every **Friday**;
- Lots of embarrassing and silly questions ;)

Bibliography

- There are lots of good books on OOP and JAVA
- We will mostly be focused on two:

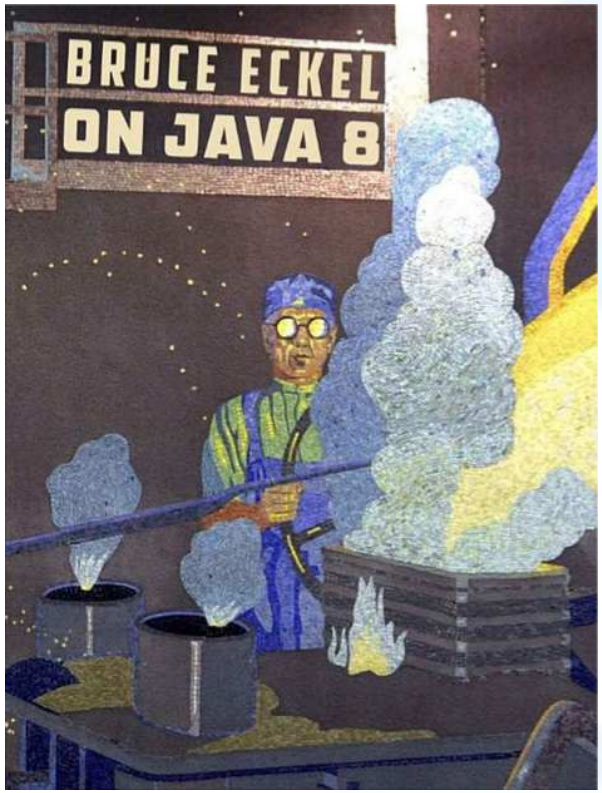

Figure: Source: (Eckel, 2017)

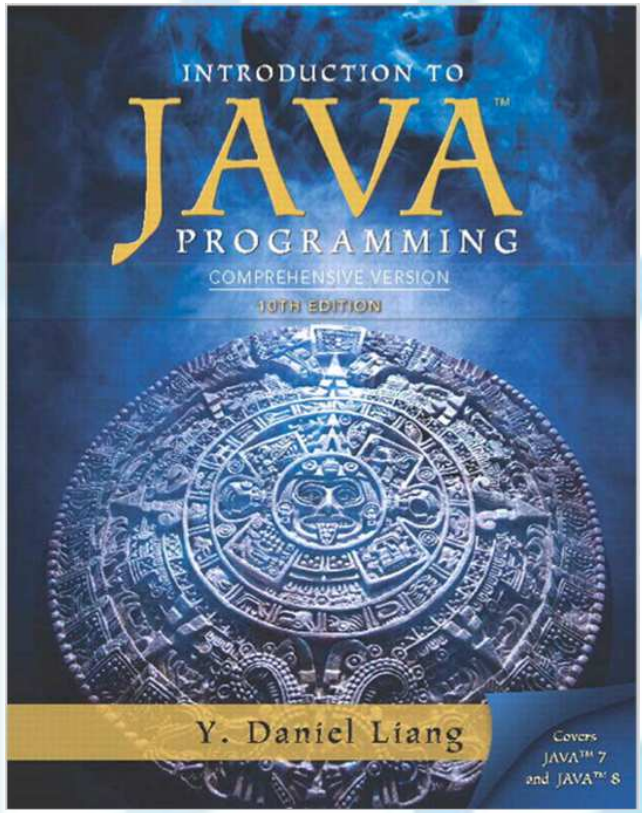

Figure: Source: (Liang, 2014)

Brucke Eckel's book:

- Considered a JAVA guru;
- I used one of his books as a computer engineering student;
  - ``Thinking in JAVA 3^{rd} edition''
- Lacks some organization:
  - Might be make student's life harder;
- Code-heavy;
- More oriented towards tech formation;

Liang's book:

- Appears to more focused on graduation students;
- Better organized;
- However:
  - I have never used it before;
  - Risky...
    - for the students...
    - not the professor ;)


SO WHAT?
I DON'T CARE.

Lets have a quick look at Professor Liang book organization...

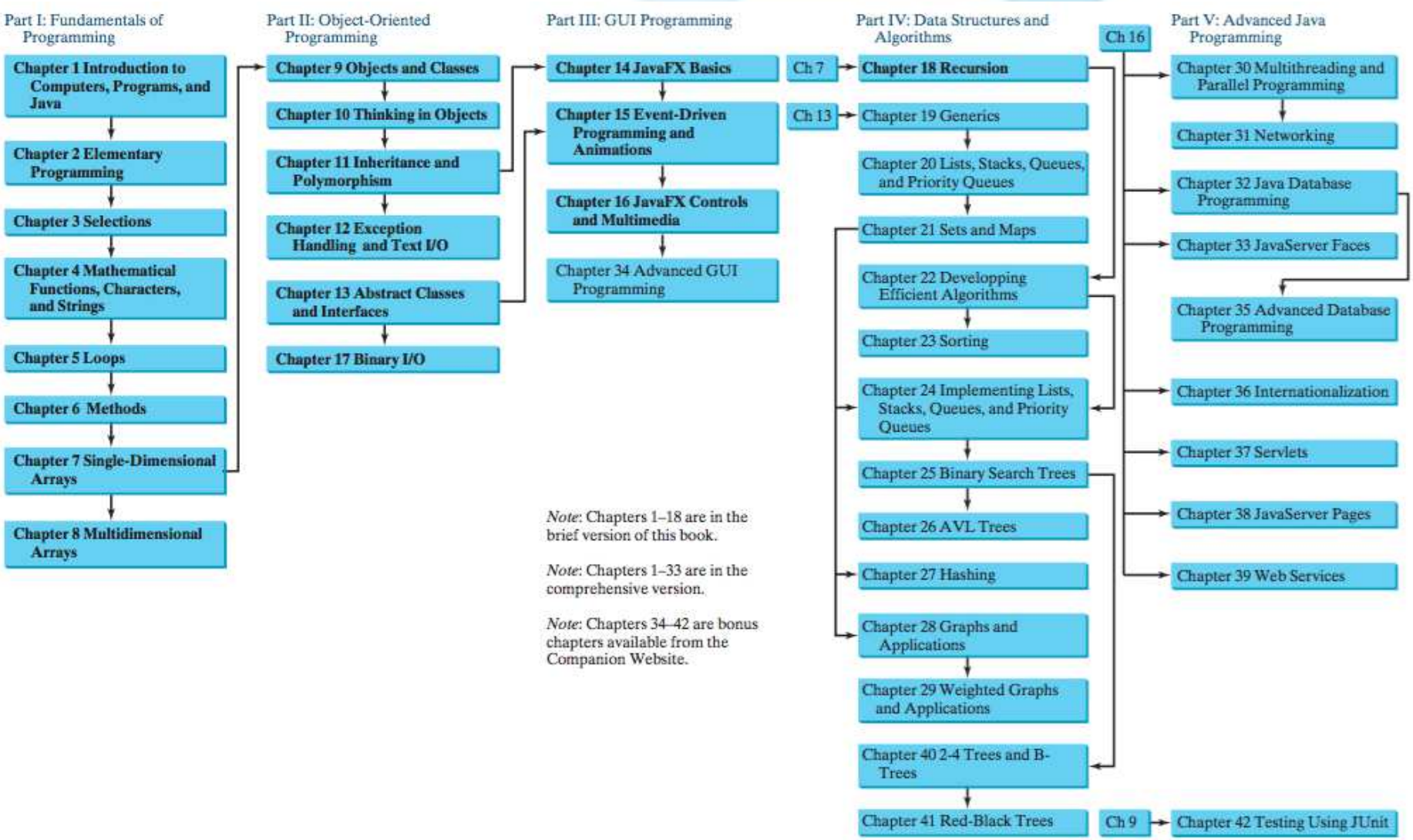

Figure: Source: (Liang, 2014)

Can you see anything interesting in this organization? Any ideas?

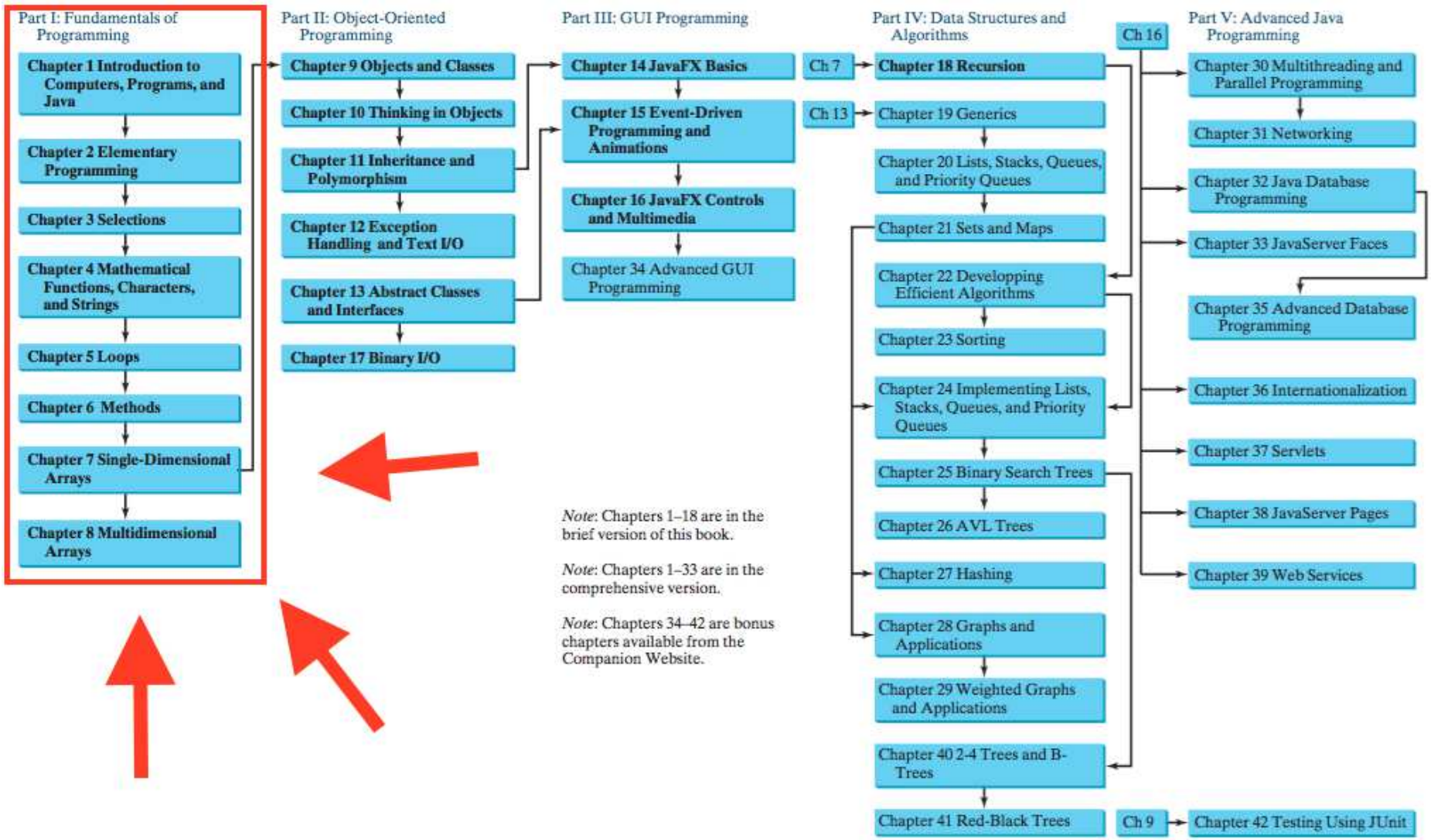

Figure: Source: (Liang, 2014)

Can you see anything interesting in this organization? Any ideas?

Part I: Fundamentals of Programming:

- Already been covered in other courses:
- First **320** pages of the book;
- We will not spend any time covering these subjects;
- Assumption: students **dominate** such matters;
- You can find these chapters on my webpage ;)


SO WHAT?
I DON'T CARE.

However:

In practice it is important to introduce some concepts =) Lets start?

However:

In practice it is important to introduce some concepts =) Lets start?

What is the very first basic example that we should start by seeing in JAVA?

A Simple Java program

File: Welcome.java

```
1: public class Welcome{
2: public static void main(String[] args) {
3:     // Display the basic hello world message
4:     System.out.println("Hello World!");
5: }
6:}
```

Some important points:

- In Java everything is a class;
- **main** method is where the program begins execution.

Creating, Compiling, and Executing a Java Program

But how can we execute this program? Any ideas?

Creating, Compiling, and Executing a Java Program

But how can we execute this program? Any ideas?

1. Java compiler compiles source file into a Java bytecode:

```
$javac Welcome.java
```

2. Compiler generates a **bytecode** Welcome.class file;
3. We are now able to execute the bytecode:

```
$java Welcome
   Hello World!
```

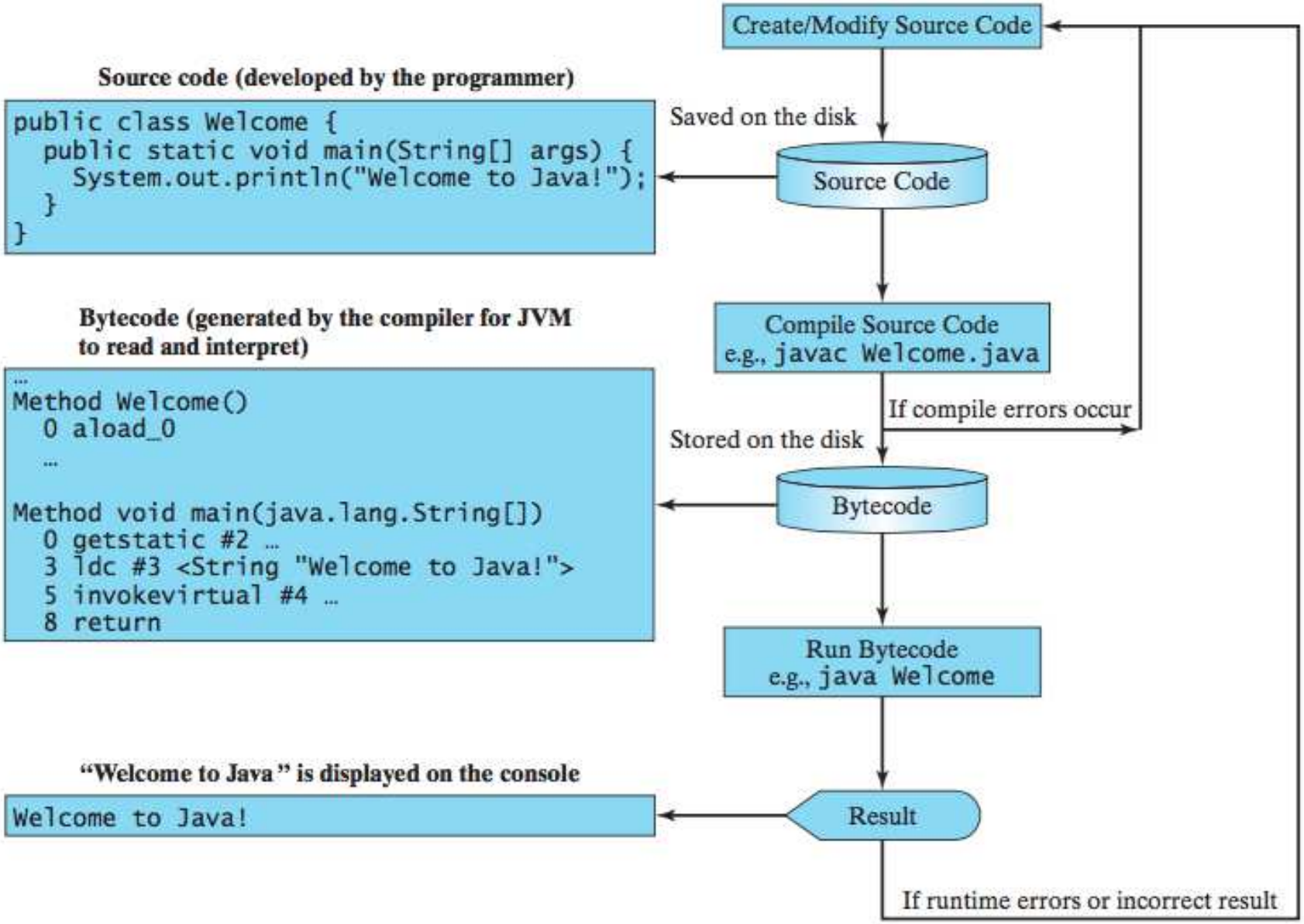

Figure: Java program-development process(Source: (Liang, 2014))

Is this clear? Any questions?

Is this clear? Any questions?

Well, I have a question:

What is a bytecode? Why does Java use bytecodes? Any ideas?

Java language is a **high-level** language:

- Java bytecode is a **low-level** language:
  - Similar to machine instructions (*i.e.*, assembly);
  - Architecture **neutral**;
  - Can run on any platform that has a Java Virtual Machine (JVM);

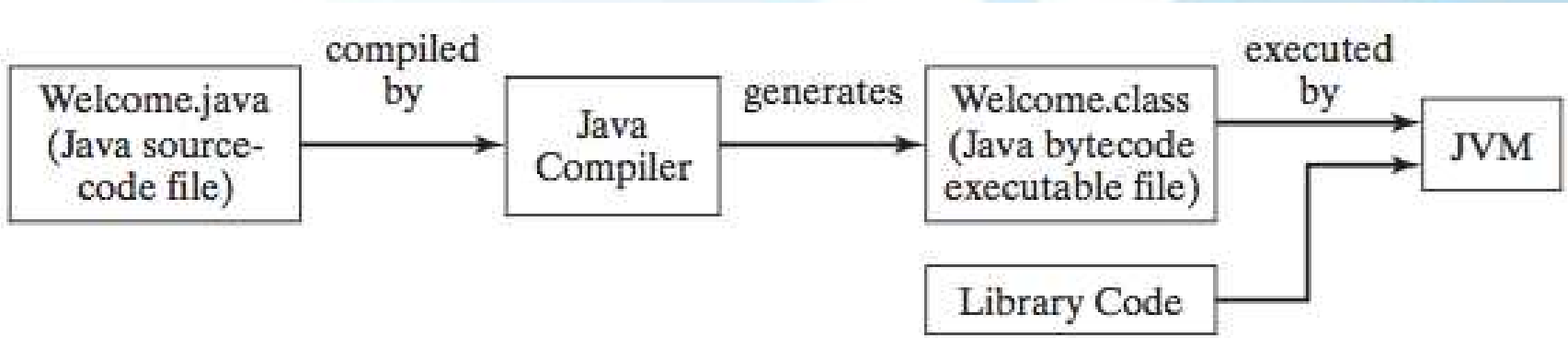

Figure: Java source code is translated into bytecode. (Source: (Liang, 2014))

But what is a JVM? Any ideas?

But what is a JVM? Any ideas?

- JVM is a program whose purpose is to execute other programs;
- Has two primary functions:
  - Allow Java programs to run on any device or operating system:
    - "Write once, run anywhere" principle
  - Manage and optimize program memory.
- Maintained by both corporate and open source developers;

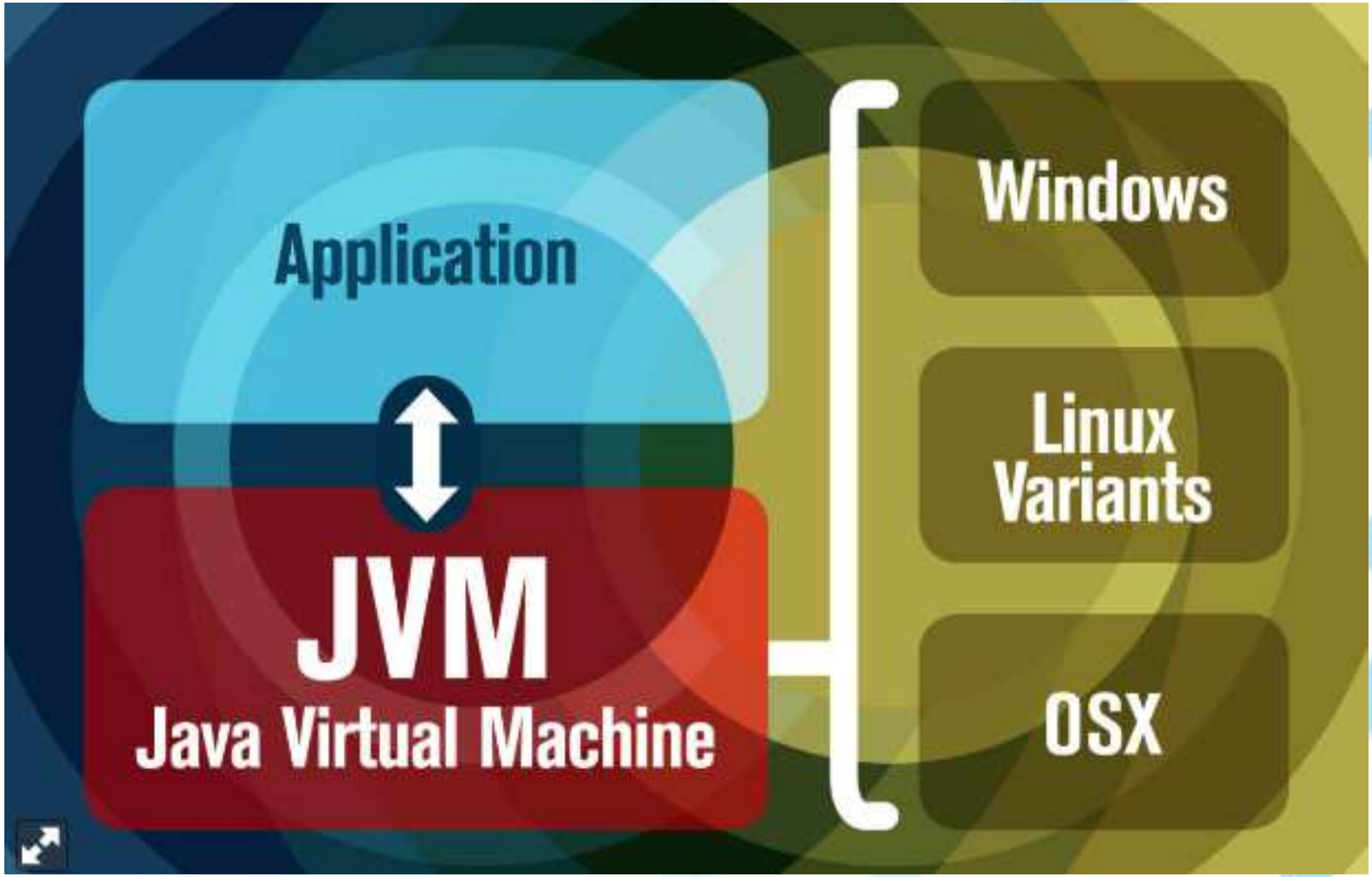

Figure: (Source: https://www.javaworld.com/article/3272244/what-is-the-jvm-introducing-the-java-virtual-machine.html)

Integrated development environment

We will use an integrated development environment **(IDE)**:

- Provides comprehensive facilities for software development;
- An IDE normally consists of:
  - Source code editor;
  - Build automation tools;
  - Debugger;

There are a lot of IDEs, two of the most famous are:

Figure: Netbeans IDE.

Figure: Eclipse IDE.

I think the laboratories have **Netbeans** installed...

- Focus on the **I think** =P

References I

Eckel, B. (2017).

On Java 8.

MindView LLC.

Liang, Y. (2014).

Introduction to Java Programming.

Pearson Education.