Introduction to Computer Science COSC1046W

Ping Luo

Myself

- Bioinformatics Specialist (Data Science), Princess Margaret Cancer Centre
- ping.luo@algomau.ca
- Office hour: 2:30-3:00

COSC1046W announcement

Ping Luo <ping.luo@algomau.ca>   Fri, Jan 5, 1:08 PM (4 days ago)

to COSC1046W24W

Hello Everyone,

My name is Ping Luo, and I'm your instructor for the course Introduction to Computer Science (COSC1046W) this term. I hope you're doing well and are ready for an engaging and productive term ahead.

I'm writing to inform you that our course will officially commence next Tuesday (Jan 9), and I'm eager to embark on this academic journey with each one of you and look forward to a semester filled with learning, collaboration, and growth.

Please take note that there will be no lab session scheduled for next week. However, regular lab sessions will commence thereafter, and an outline will be uploaded to Moodle next week.

If you have any questions or concerns before the start of the course, feel free to reach out. I am here to assist you and ensure a smooth transition into the semester.

Wishing you all a successful and rewarding term!

Best regards,
Ping

Learning Objectives

- Understand the structure and components of the Java programming language;

- Utilize common software components and IDEs to create, compile, run and test Java programs;

- Analyze complex problems involving multiple data types, including objects, strings, arrays, and multiple types of control structures;

- Understand the context of the programming process within the field of Computer Science.

Textbook
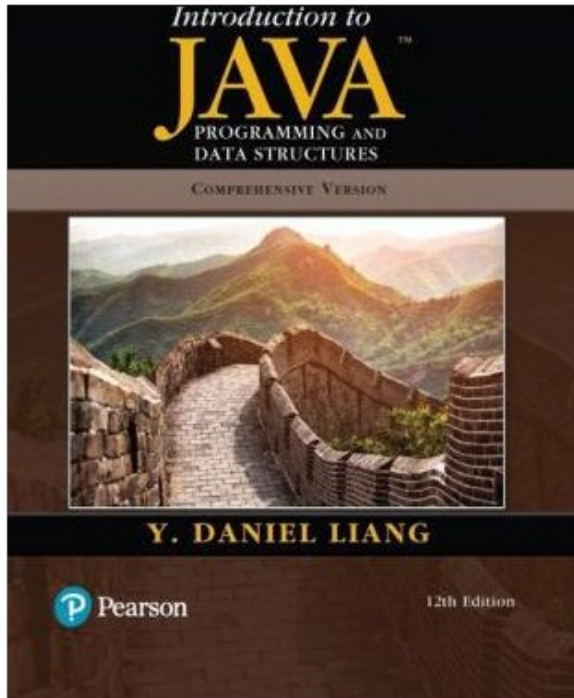
Intro to Java Programming & Data Structures, Comprehensive Version

by Liang

$104.00 to $241.00

Buy

New $241.00

Backordered ?

Digital ? $104.00
Requirements

ADD TO BAG → ♡ **Add to Wishlist**

https://www.bkstr.com/algomastore/product/introduction-to-java-programming-and-data-structures--comprehensive-version-938848-1

https://www.pearson.com/en-ca/subject-catalog/p/introduction-to-java-programming-and-data-structures/P200000003470/9780137554768

Textbook

Brief Contents

Course Assessment

Assessment	Weighting
Quiz (4)	20%
Lab (10)	20%
Mid-term Exam	25%
Final Exam	35%
Total	100%

Late assignment & Missed Test and Exam

- Late assignments and missed exams will receive a grade of 0, except in cases of valid medical issues supported by a doctor's note.

Programs

Computer **programs**, known as **software**, are instructions to the computer.

You tell a computer what to do through programs. Without programs, a computer is an empty machine. Computers do not understand human languages, so you need to use computer languages to communicate with them.

Programs are written using programming languages.

Programming Languages

Machine Language   Assembly Language   High-Level Language

Machine language is a set of primitive instructions built into every computer. The instructions are in the form of binary code, so you have to enter binary codes for various instructions. Program with native machine language is a tedious process. Moreover the programs are highly difficult to read and modify. For example, to add two numbers, you might write an instruction in binary like this:

`1101101010011010`

Programming Languages

Machine Language   Assembly Language   High-Level Language

Assembly languages were developed to make programming easy. Since the computer cannot understand assembly language, however, a program called assembler is used to convert assembly language programs into machine code. For example, to add two numbers, you might write an instruction in assembly code like this:

ADDF3 R1, R2, R3

Programming Languages

Machine Language   Assembly Language   High-Level Language

The high-level languages are English-like and easy to learn and program. For example, the following is a high-level language statement that computes the area of a circle with radius 5:

$$area = 5 * 5 * 3.1415;$$

Interpreting/Compiling Source Code

A program written in a high-level language is called a **source program** or **source code**. Because a computer cannot understand a source program, a source program must be translated into machine code for execution. The translation can be done using another programming tool called an **interpreter** or a **compiler**.

Why Java?

The answer is that Java enables users to develop and deploy applications on the Internet for servers, desktop computers, and small hand-held devices. The future of computing is being profoundly influenced by the Internet, and Java promises to remain a big part of that future. Java is the Internet programming language.

- Java is a general purpose programming language.

- Java is the Internet programming language.

- Java is the most popular language in **software development**.

Java, Web, and Beyond

- Java can be used to develop standalone applications.

- Java can be used to develop applications running from a browser.

- Java can also be used to develop applications for hand-held devices.

- Java can be used to develop applications for Web servers.

Java's History

- James Gosling and Sun Microsystems

- Oak (original name)

- Early History Website:

  http://www.java.com/en/javahistory/index.jsp

Characteristics of Java

- Java Is Simple

- Java Is Object-Oriented

- Java Is Distributed

- Java Is Interpreted

- Java Is Robust

- Java Is Secure

- Java Is Architecture-Neutral

- Java Is Portable

- Java's Performance

- Java Is Multithreaded

- Java Is Dynamic

https://liveexample.pearsoncmg.com/etc/JavaCharacteristics.pdf

JDK (Development Kit) Editions

- Java Standard Edition (J2SE)

  - J2SE can be used to develop client-side standalone applications or applets.

- Java Enterprise Edition (J2EE)

  - J2EE can be used to develop server-side applications such as Java servlets, Java ServerPages, and Java ServerFaces.

- Java Micro Edition (J2ME).

  - J2ME can be used to develop applications for mobile devices such as cell phones.

Download Java

- https://www.oracle.com/java/technologies/downloads/
- JDK 21 oracle

Popular Java IDEs

- IDE: Integrated Development Environment
    - NetBeans
    - Eclipse
    - …

A Simple Java Program

Listing 1.1

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

```java
// This program prints Welcome to Java!
public class Welcome {
   public static void main(String[] args) {
      System.out.println("Welcome to Java!");
   }
}
```

Command Prompt

```
C:\book>java Welcome
Welcome to Java!

C:\book>
```

print a message to the console

Compiling Java Source Code

You can port a source program to any machine with appropriate compilers. The source program must be recompiled, however, because the object program can only run on a specific machine. Nowadays computers are networked to work together. Java was designed to run object programs on any platform. With Java, you write the program once, and compile the source program into a special type of object code, known as **bytecode**. The bytecode can then run on any computer with a Java Virtual Machine, as shown below. Java Virtual Machine is a software that interprets Java bytecode.

Compiling and Running Java From the Command Window

- Set path to JDK bin directory
  - set path=c:\Program Files\java\jdk1.8.0\bin
- Set classpath to include the current directory
  - set classpath=.
- Compile
  - javac Welcome.java
- Run
  - java Welcome

Anatomy of a Java Program

- Class name
- Main method
- Statements
- Statement terminator
- Reserved words
- Comments
- Blocks

Class Name

Every Java program must have at least one class. Each class has a name. By convention, class names start with an uppercase letter. In this example, the class name is Welcome.

```java
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Main Method

Line 2 defines the main method. In order to run a class, the class must contain a method named main. The program is executed from the main method.

```
// This program prints Welcome to Java!
public class Welcome {
  public static void main(String[] args) {
    System.out.println("Welcome to Java!");
  }
}
```

Statement

A statement represents an action or a sequence of actions. The statement System.out.println("Welcome to Java!") in the program in Listing 1.1 is a statement to display the greeting "Welcome to Java!".

```java
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Statement Terminator

Every statement in Java ends with a semicolon (;).

```java
// This program prints Welcome to Java!
public class Welcome {
    public static void main(String[] args) {
        System.out.println("Welcome to Java!");
    }
}
```

Special Symbols

Character	Name	Description
{ }	Opening and closing braces	Denotes a block to enclose statements.
( )	Opening and closing parentheses	Used with methods.
[ ]	Opening and closing brackets	Denotes an array.
/ /	Double slashes	Precedes a comment line.
" "	Opening and closing quotation marks	Enclosing a string (i.e., sequence of characters).
;	Semicolon	Marks the end of a statement.

Programming Style and Documentation

- Appropriate Comments

- Naming Conventions

- Proper Indentation and Spacing Lines

- Block Styles

- Google Style Guides

- https://google.github.io/styleguide/

Programming Errors

- Syntax Errors
  - Detected by the compiler
- Runtime Errors
  - Causes the program to abort
- Logic Errors
  - Produces incorrect result

Syntax Errors

```java
public class ShowSyntaxErrors {
  public static main(String[] args) {
    System.out.println("Welcome to Java);
  }
}
```

Runtime Errors

```java
public class ShowRuntimeErrors {
  public static void main(String[] args) {
    System.out.println(1 / 0);
  }
}
```

Logic Errors

```java
public class ShowLogicErrors {
  public static void main(String[] args) {
    System.out.println("Celsius 35 is
    Fahrenheit degree ");
    System.out.println((9 / 5) * 35 + 32);
  }
}
```

Find Your Java IDE

- IDE: integrated development environment
- Java IDE: Eclipse, NetBeans, VScode with Java extension, etc.