Course Overview

- Web site
- Syllabus
- Schedule
- Lecture slides
- Reading Assignments & Textbook
- Handouts

The Java Programming Language

One of the most successful programming languages ever
Over 9 million developers in every major industry
The Java Programming Language

Designed by James Gosling and the "Green Team"

at Sun Microsystems, Inc.

The Java Programming Language

Originally intended to control consumer devices
special-purpose language

Soon changed to a much broader scope
general-purpose language

First public release: 1995
Sun Microsystems was purchased by Oracle in 2010

The Java Programming Language

Key features when Java was first released:

Platform Independence – not tied to one type of computer
Applets – Programs that run in a web browser
Object-oriented – A effective programming paradigm
Garbage Collection – Java "cleaned up" memory by getting rid of unused objects
Hello, World!

A **program** is a sequence of instructions expressed in a particular **programming language** such as Java. We’ll start with a classic first example.

**Program statement:**

```java
System.out.println("Hello, world!");
```

**Output:**

```
hello, world!
```

Hello, World!

`System.out` is an object representing the console window, and `println` is the name of a method. A method is a group of program statements that can be **called** (or **invoked**). The `println` method is part of a big library of code that we can make use of in any Java program.

```java
public static void main(String[] args) {
    System.out.println("Hello, world!");
}
```

The **main method** is the starting point of any Java program. The **header** of the main method must be written like this:

For now, just consider this the scaffolding necessary to write a main method. The code between the `{` and `}` is called the **method body**.
Finally, every Java method must be defined in a class.

```java
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
```

Similar to a method, a class has a class header and a class body.

Consistent indentation makes the program more readable.

Spaces and tabs in a program are called white space.

Another example:

```java
public class Proverb {
    public static void main(String[] args) {
        System.out.println("Tell me and I forget.");
        System.out.println("Show me and I remember.");
        System.out.println("Involve me and I understand.");
    }
}
```

Tell me and I forget.
Show me and I remember.
Involve me and I understand.

Make use of the download and R & R buttons on code in the textbook.

- Download the program to your computer
- Run & Revise the code right in the browser
The print and println Methods

The System.out object prints output to the console window

The println method moves to the next line after it prints its output

The print method does not

System.out.println("One");
System.out.println("Two");
System.out.println("Three");

One
Two
Three

System.out.print("One, ");
System.out.print("Two, ");
System.out.println("Buckle my shoe.");
System.out.println();
System.out.print("Three, ");
System.out.print("Four, ");
System.out.println("Close the door.");

One, Two, Buckle my shoe.
Three, Four, Close the door.

The print and println Methods

The print and println methods can print any type of data

System.out.println(25);

This is an example of method overloading – a method accepting different types of data

Expressions in the arguments are evaluated and the results are sent to the method

System.out.println(38 + 31);

69
The print and println Methods

Character strings cannot be broken across lines:

`System.out.println("No word in the English language rhymes with the words month or orange");`

The plus sign can be used to perform string concatenation

`System.out.println("No word in the English language " + "rhymes with the words month or orange");`

The two strings are joined into one long string which is passed to the method.

The print and println Methods

Strings can be concatenated to numbers:

`System.out.println("The total is: " + 4321);`

The number is converted to a string and the strings are joined. That's more helpful when using variables:

`System.out.println("The total is: " + total);`

The plus sign is an overloaded operator – it operates on different types of data. It determines which operation to perform based on the types of its operands. The plus operator is evaluated left to right.

`System.out.println("Concatenated: " + 123 + 456);`

Concatenated: 123456

`System.out.println("Added: " + (123 + 456));`

Added: 579
Comments and Programming Style

Comments explain a program's purpose and processing. They are intended for the human reader – they have no effect on the program. A single-line comment begins with a // and continues until the end of the line.

```java
// This is a comment
balance = balance - fees; // deduct monthly fees
```

It might be put on the end of a line of code.

A multi-line comment begins with a /* and ends with a */. It might span multiple lines.

```java
/*
A multi-line comment that only spans one line */
/*
A multi-line comment that spans multiple lines might be formatted like this.
*/
```

A variation of the multi-line comment begins with /** and is called a JavaDoc comment.

JavaDoc comments are used to generate online documentation.

```java
/*
* Demonstrates the use of comments.
* public class SimpsonQuote {
* /
* prints a quote from The Simpsons TV show.
* /
* public static void main(String[] args) {
* // from the episode "Part vs. Thanksgiving"
System.out.println("Operator, what's the number " +
"(for 9-1-1?)"); // season 2, ep 7
System.out.println( " = Homer Simpson");
* }
* }
```
A software development environment is software that helps you develop software create, run, organize, modify, test, debug. There are many options, some free and some not. It's important to get comfortable with whatever development environment you use. There are only a few crucial tools you need initially – learn the rest over time. Two categories: command-line and integrated environments.

A command-line environment is a suite of separate tools executed as individual commands in a console window. The Java Development Kit (JDK) from Oracle is a free command-line environment.

An integrated development environment is one large program that combines the various tools. This one is Eclipse. Others are BlueJ, DrJava, JEdit, jGRASP, NetBeans.
Compiling and Executing a Java Program

Bytecode is a "low-level" version of a Java program – it's not something you edit directly.

Bytecode is not associated with any particular type of computer. That's what makes Java platform-independent.

The Java Virtual Machine is implemented in software designed to execute the bytecode.

A Java program can be run on any computer with a JVM.

Java's original slogan:

*Write once, run anywhere.*

Programming Errors

The key is the relationship between two language aspects:

- **syntax** – the rules that determine how words and symbols can be combined
- **semantics** – the meaning of a syntactic element

An English sentence can be syntactically valid and have multiple semantic interpretations.

In a programming language, **syntax determines semantics**.

If a programming statement is syntactically valid, it has only one interpretation.

Programming Errors

Programming errors occur when we violate the language syntax rules or when the semantics we set up are not what we intended.

There are three types of programming errors:

- syntax errors
- runtime errors
- logic errors

Errors happen. They are a part of programming. Don’t sweat it. Just learn how to deal with them.
Programming Errors

The compiler analyzes a program and reports any syntax errors

```java
public class Mistakes {
    public static void main(String[] args) {
        System.out.println("No mistakes, just lessons.");
    }
}
```

Programming Errors

Syntax errors are reported in various ways depending on the development environment

```
Mistakes.java:3: <identifier> expected
  public static void main(String[] args)
     ^
Mistakes.java:5: unclosed string literal
  System.out.println("No mistakes, just lessons.");
     ^
Mistakes.java:5: ';' expected
  System.out.println("No mistakes, just lessons.");
     ^
Mistakes.java:7: reached end of file while parsing
}
^
4 errors
```

Programming Errors

A runtime error occurs when an operation can't be carried out for some reason

The program terminates (crashes) immediately

In Java, a runtime error is usually represented by an exception

```
System.out.println(123 / 0)
```

Exception in thread "main" java.lang.ArithmeticException: / by zero
  at DivisionAttempt.main(DivisionAttempt.java:5)
```
Programming Errors

**A logic error** occurs when the program doesn’t produce the desired results.

```java
System.out.println("The perimeter of a square with " +
"3 inch sides is ");
System.out.println(4 * 4 + "feet.");
```

The perimeter of a square with 3 inch sides is 16 feet

- perimeter is spelled wrong
- wrong answer (16)
- wrong output units (feet)
- missing space between 16 and feet

The Java API

Java is supported by a massive class library of code you can use.
The standard class library is called the Java API (Application Programmer Interface).
There are hundreds of classes in the Java API containing thousands of methods.
For example, the System class is part of the Java API.
It’s rare to write a program that doesn’t use classes from the API.

The Java API

Java programmers rely heavily on the online documentation about the Java API.

docs.oracle.com/javase/8/docs/api/

Bookmark it. Get comfortable navigating it. Make it your friend.
The textbook contains links to specific classes as needed.