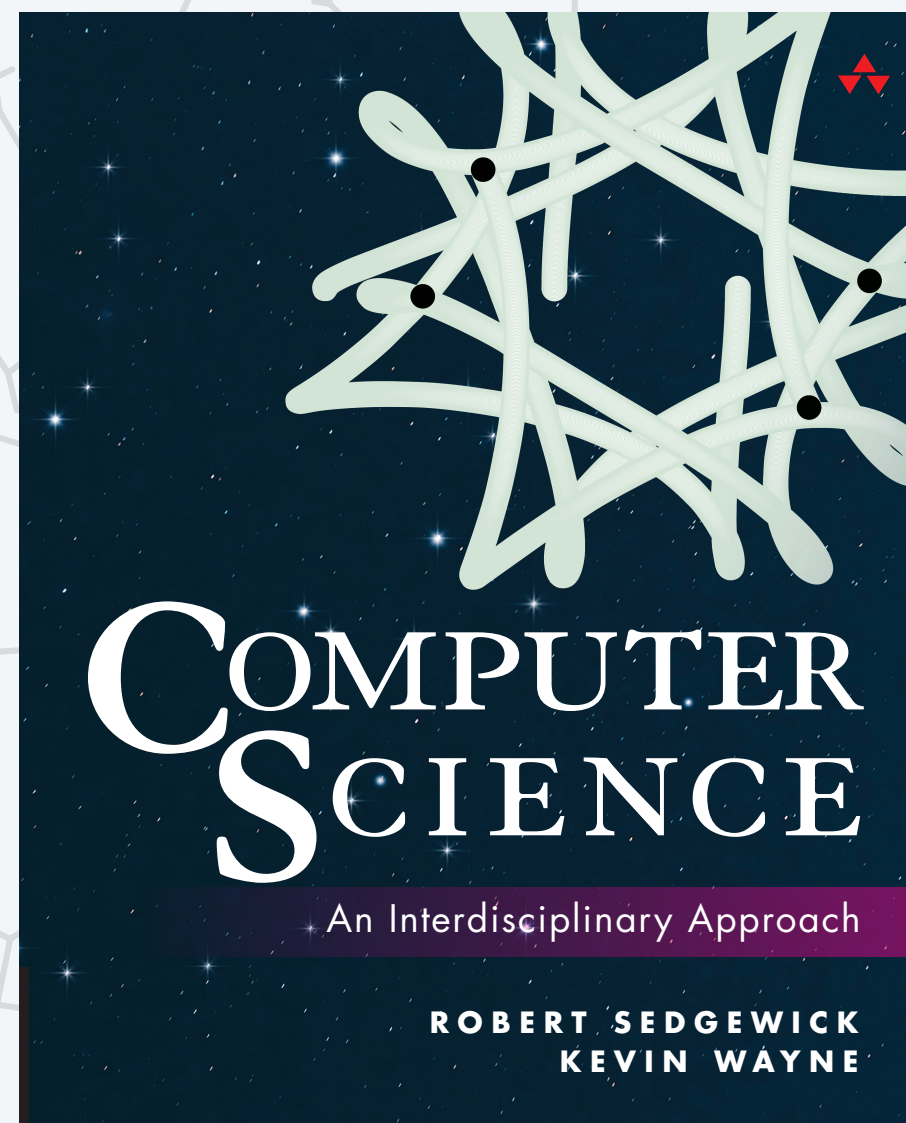


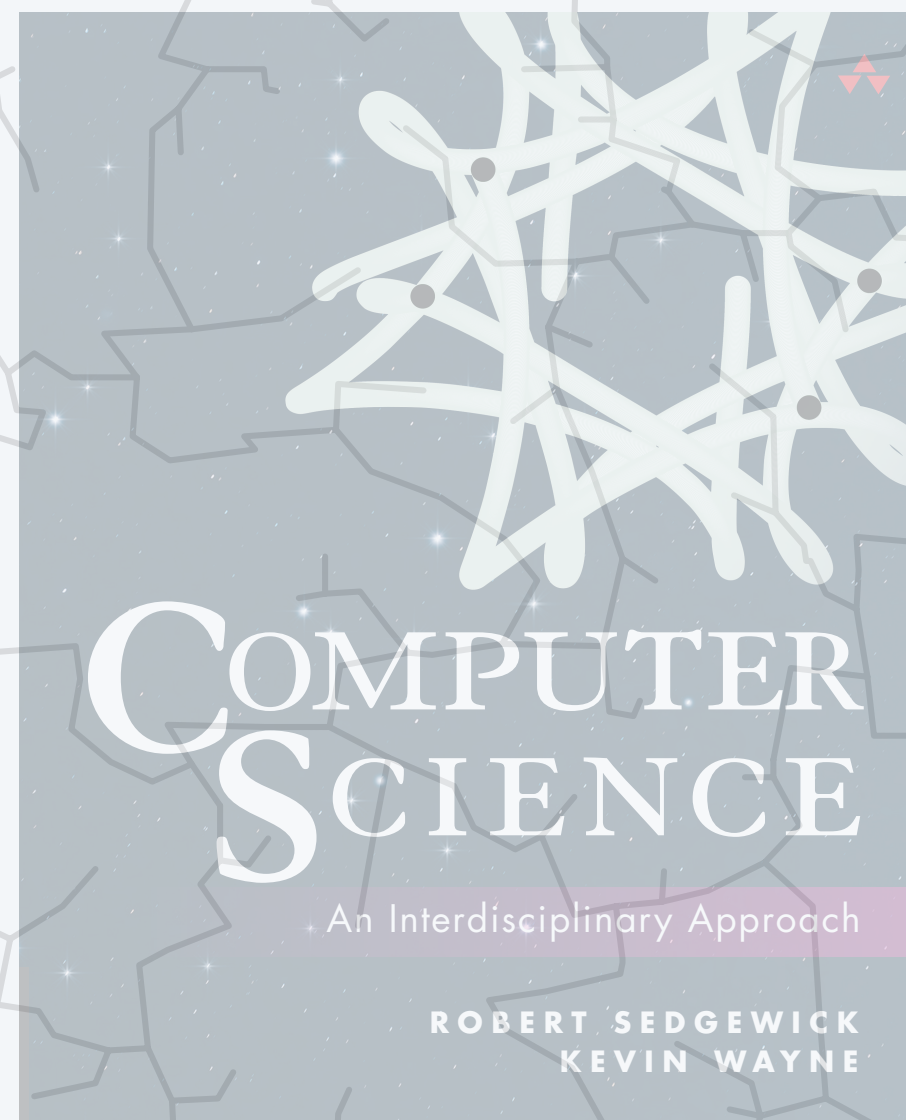


<https://introcs.cs.princeton.edu>

1.1 HELLO, WORLD

- ▶ *why programming?*
- ▶ *your first program*
- ▶ *program development*





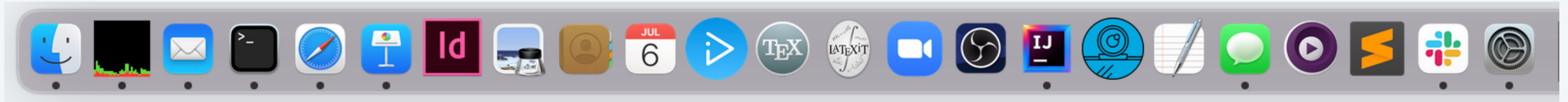
<https://introcs.cs.princeton.edu>

1.1 HELLO, WORLD

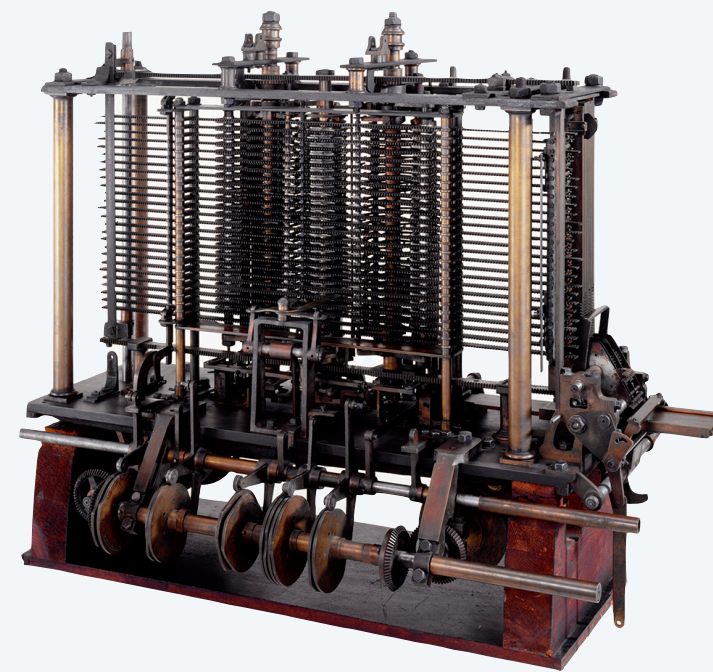
- ▶ *why programming?*
- ▶ *your first program*
- ▶ *program development*

Why you need to know how to program

Apps. Great when they do exactly what you want.



Programming. Essential when you want the computer to do something **new**.



Analytical Engine
(first "computer")



Ada Lovelace
(first programmer)

Telling a computer what to do

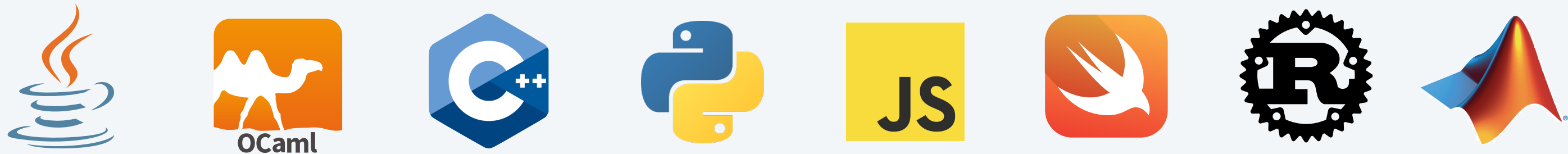
Machine languages. Native for **computers**; painful for people.

```
0011111100100001100110111001011101110011101000000010001110111100101
1101110111011111000011100010010100001001110000011010100110100001...
```

Natural languages. Convenient for **people**; ambiguous for computers.



High-level programming languages. Learnable for **people**; precise for **computers**.



GenAI tools. Rapid progress in translating natural language into code. ←

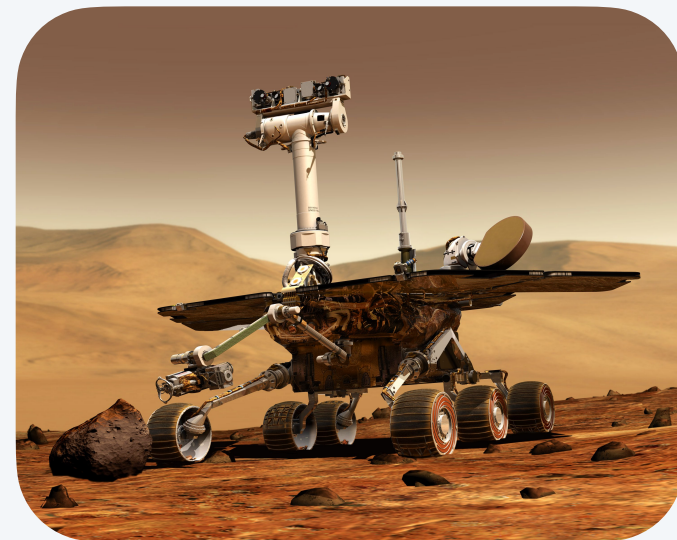
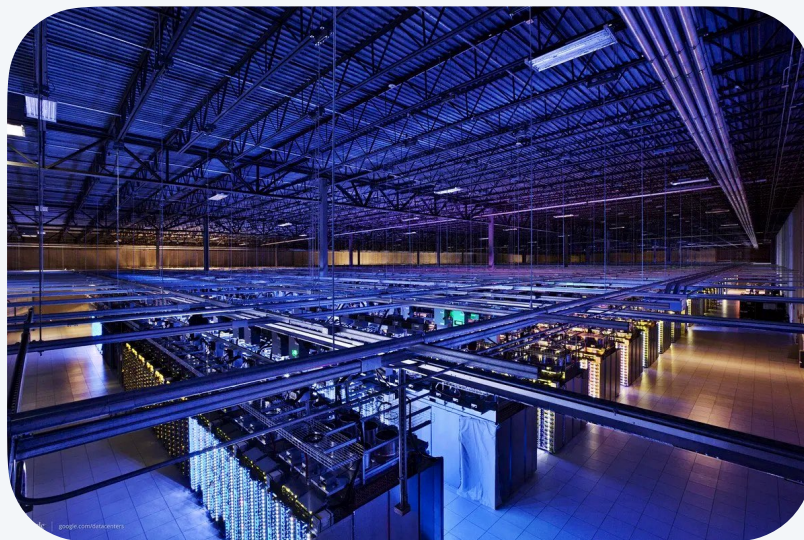
*Learning to reason about code
crucial for most effective use.*



Java features.

- Supports modern abstractions (types, objects, modularity) and rich libraries.
- Provides compile-time and runtime checks that catch many errors early.
- Runs on macOS, Windows, and Linux (free tools).
- Powers billions of devices.

Ex. Android phones/TVs, web servers, Mars rover, medical devices, Internet of Things, ...



Reality. Different domains gravitate toward different languages.

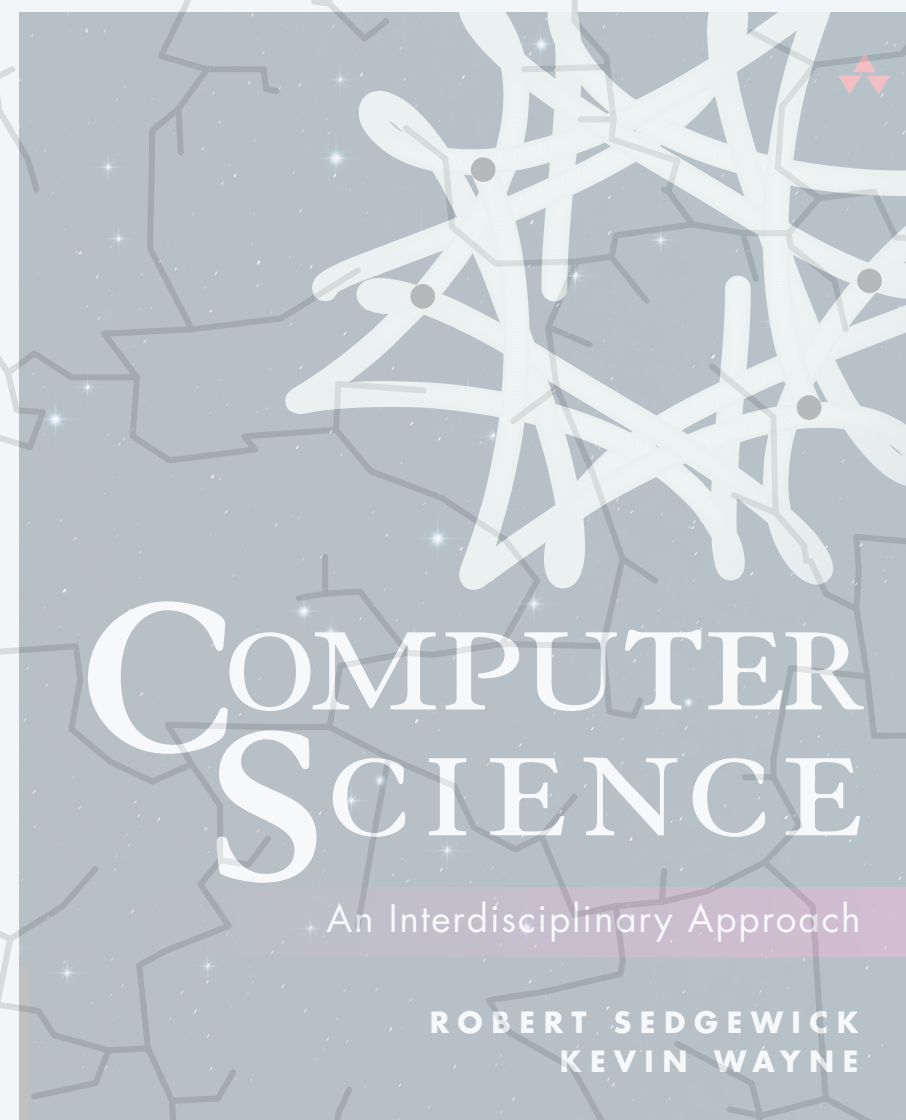
Programming fundamentals. This course uses Java to teach **transferable programming skills**.

A rich subset of the Java language



In this course, these constructs are all we need. *← seems like a lot to learn, but you already know >20,000 words!*

data types		arithmetic		boolean		control flow		classes / methods		strings	
int		+	-	true	false	if	else	public	private	+	
double		*	/	&&		while	for	class	new	length()	
boolean		%		!	^	do	return	static	final	charAt()	
char						break	continue	void	main	compareTo()	
String								import		toString()	
punctuation		comments		assignment		math library		type conversion		textbook libraries	
{ }		/*	*/	=		Math.min()	Math.max()	Integer.parseInt()		StdIn/In	
()		//		++ --		Math.sqrt()	Math.abs()	Double.parseDouble()		StdOut/Out	
. ,		< >		+= -=		Math.log()	Math.exp()	System.out.print()		StdPicture/Picture	
' "		<= >=		*= /=		Math.round()	Math.pow()			StdDraw/Draw	
; <>		== !=		arrays		Math.sin()	Math.cos()	System.out.println()		StdAudio/Audio	
				[]		Math.toDegrees()	Math.PI	System.out.printf()			
				length		Math.toRadians()	Math.E				

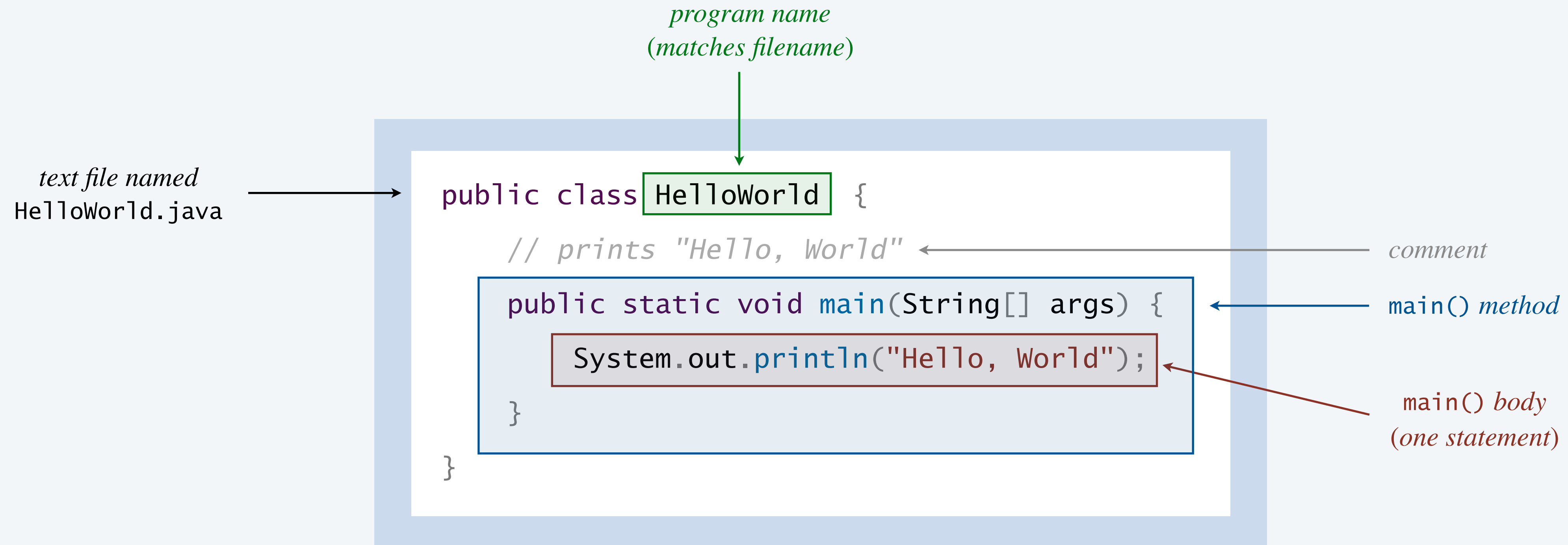


<https://introcs.cs.princeton.edu>

1.1 HELLO, WORLD

- *why programming?*
- *your first program*
- *program development*

Our first Java program: Hello, World



terminal window →

```
~/cos126/hello> javac HelloWorld.java ← compile
```

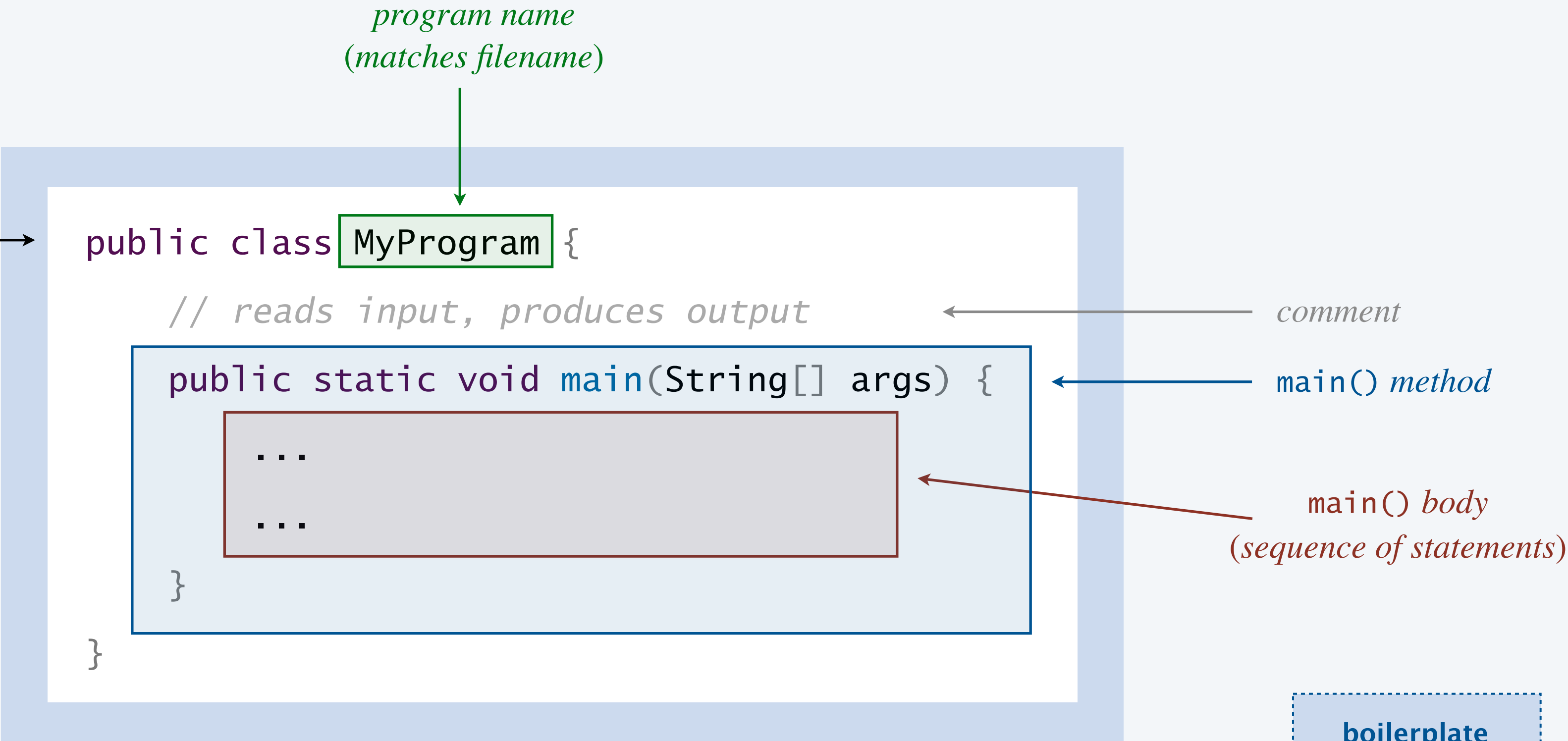
```
~/cos126/hello> java HelloWorld ← run
```

```
Hello, World
```

output →

Template for our first Java programs

text file named
MyProgram.java



```
~/cos126/hello> javac MyProgram.java  
~/cos126/hello> java MyProgram  
[output]
```

boilerplate
public
class
static
void
main
String[]

we'll learn what
these mean later

Adding a command-line argument

Command-line arguments. Input passed to `main()` when the program starts.

```
public class HelloWorldWithArgument {  
    // prints a command-line argument  
    public static void main(String[] args) {  
        System.out.print("Hello, World ");  
        System.out.println(args[0]);  
    }  
}
```

*print() does not
print a newline*

command-line argument

```
~/cos126/hello> javac HelloWorldWithArgument.java
```

```
~/cos126/hello> java HelloWorldWithArgument Kevin  
Hello, World Kevin
```

*command-line
argument*

```
~/cos126/hello> java HelloWorldWithArgument 🐅🐅🐅  
Hello, World 🐅🐅🐅
```



Standard audio. A library for sound.

```
public class HelloWorldWithAudio {  
    // prints and speaks "Hello, World"  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
        StdAudio.play("HelloWorld.wav");  
    }  
}
```

an audio file



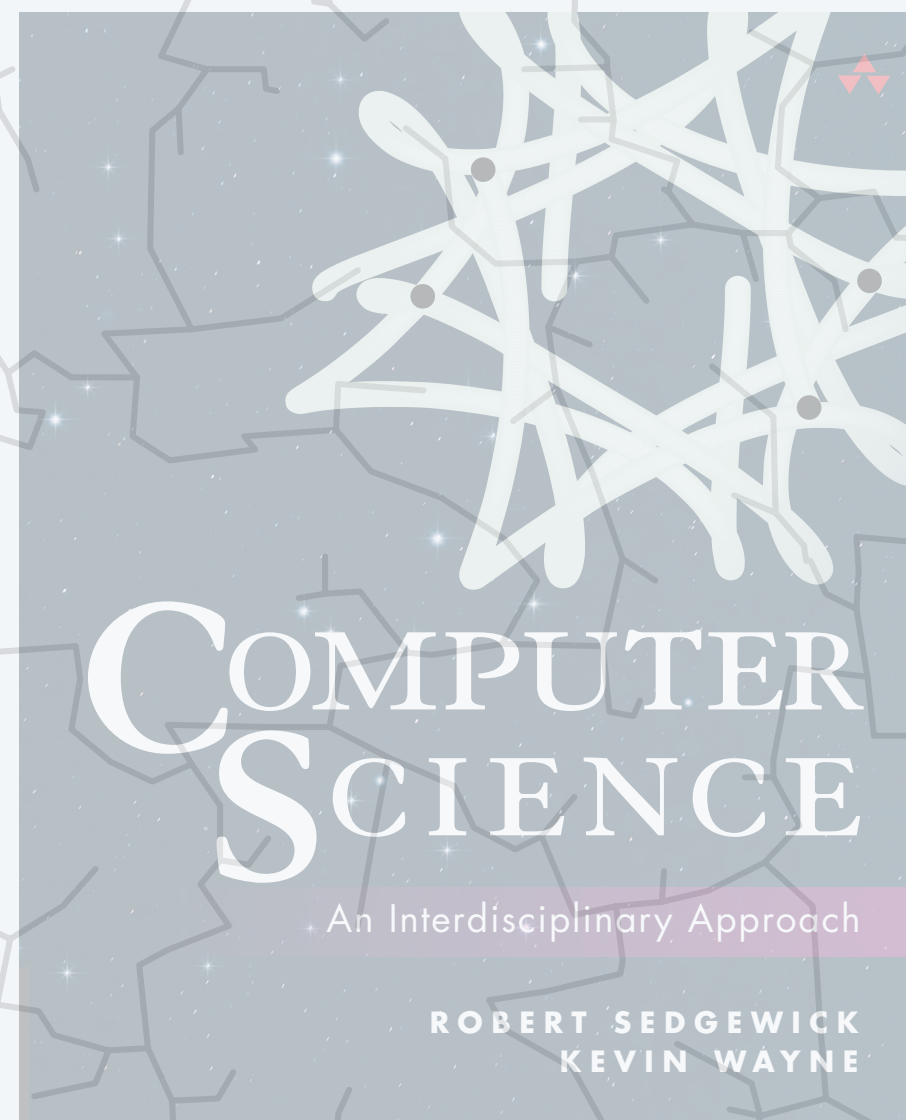
HelloWorld.wav

```
~/cos126/hello> javac-introcs HelloWorldWithAudio.java
```

```
~/cos126/hello> java-introcs HelloWorldWithAudio  
Hello, World
```

🔊 [speaks "Hello, World"]

*use these commands so
Java can find StdAudio*



<https://introcs.cs.princeton.edu>

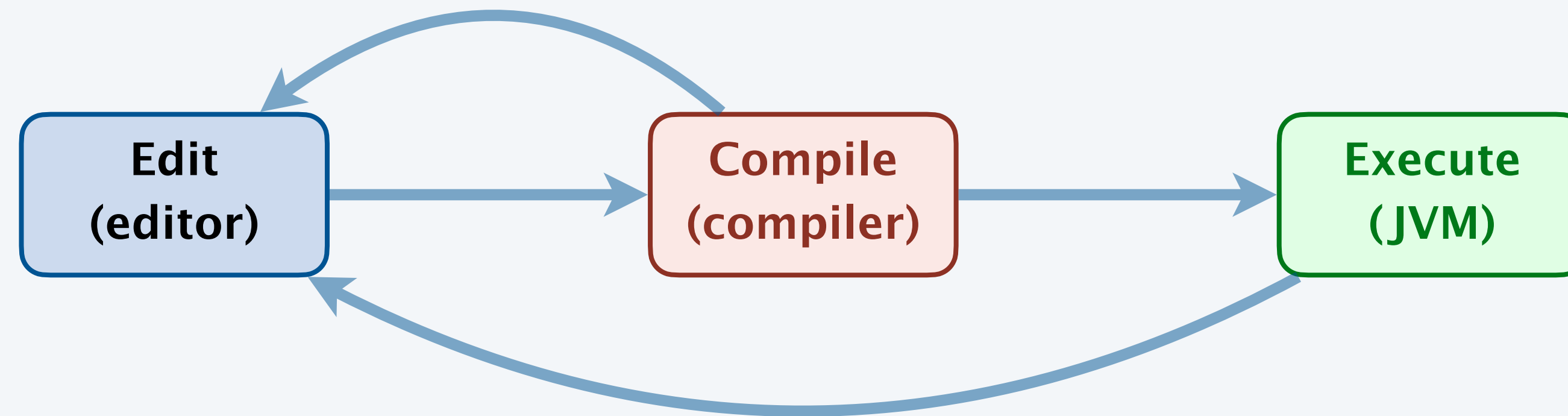
1.1 HELLO, WORLD

- *why programming?*
- *program development*

Program development in Java

Developing a Java program involves three phases:

- **Edit:** write the program.
- **Compile:** translate it into a machine language for the JVM (bytecode).
- **Execute:** run the program on inputs to produce outputs.



analogous to other creative processes
(compose–rehearse–perform)

Almost always requires multiple edit–compile–execute cycles:

- If it won't compile (not valid Java), fix your code.
- If it compiles but crashes or produces wrong output, debug and fix your code.

Compile-time errors

A **compile-time error** is an error reported by the compiler for an invalid Java program.  *the program does not run*

Ex 1. Missing **void**.

```
public class CompileTimeError1 {  
    public static main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

```
~/cos126/hello> javac CompileTimeError1.java  
CompileTimeError1.java:2: error: invalid method declaration;  
return type required  
    public static main(String[] args) {  
                ^  
1 error
```

Compile-time errors

A **compile-time error** is an error reported by the compiler for an invalid Java program.  *the program does not run*

Ex 2. Missing right curly brace.

```
public class CompileTimeError2 {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

```
~/cos126/hello> javac CompileTimeError2.java  
CompileTimeError2.java:4: error: reached end of file while  
parsing  
}  
 ^  
1 error
```

Compile-time errors

A **compile-time error** is an error reported by the compiler for an invalid Java program.

Ex 3. Java cannot find the textbook libraries.

```
public class CompileTimeError3 {  
    public static void main(String[] args) {  
        StdAudio.play("HelloWorld.wav");  
    }  
}
```

```
~/cos126/hello> javac CompileTimeError3.java  
CompileTimeError3.java:3: error: cannot find symbol  
    StdAudio.play("HelloWorld.wav");  
    ^  
symbol:   variable StdAudio  
location: class CompileTimeError3  
  
~/cos126/hello> javac-introcs CompileTimeError3.java
```

Runtime exceptions

A **runtime exception** is an exception thrown while the program is running.  *the program terminates abruptly*

Ex 1. Missing file (wrong filename).

```
public class RuntimeException1 {  
    public static void main(String[] args) {  
        StdAudio.play("GoodbyeWorld.wav");  
    }  
}
```

```
~/cos126/hello> javac-introcs RuntimeException1.java  
  
~/cos126/hello> java-introcs RuntimeException1  
Exception in thread "main" java.lang.IllegalArgumentException:  
could not read audio file 'GoodbyeWorld.wav'  
    at StdAudio.getAudioInputStreamFromFile(StdAudio.java:305)  
    at StdAudio.play(StdAudio.java:402)  
    at RuntimeException1.main(RuntimeException1.java:3)
```

Runtime exceptions

A **runtime exception** is an exception thrown while the program is running.  *the program terminates abruptly*

Ex 2. Missing command-line argument.

```
public class RuntimeException2 {  
    public static void main(String[] args) {  
        System.out.print("Hello, World ");  
        System.out.println(args[0]);  
    }  
}
```

```
~/cos126/hello> javac RuntimeException2.java
```

```
~/cos126/hello> java RuntimeException2
```

```
Hello, World
```

```
Exception in thread "main"
```

```
java.lang.ArrayIndexOutOfBoundsException:
```

```
Index 0 out of bounds for length 0
```

```
    at RuntimeException1.main(RuntimeException2.java:4)
```

line #



Coding style

Coding style. Indentation, whitespace, naming conventions, comments, ...

Goal. Make your programs easier for people (including you!) to read and understand.

textbook style

Program 1.1.1 Hello, World

```
public class HelloWorld
{
    public static void main(String[] args)
    {
        // Prints "Hello, World" in the terminal window.
        System.out.println("Hello, World");
    }
}
```

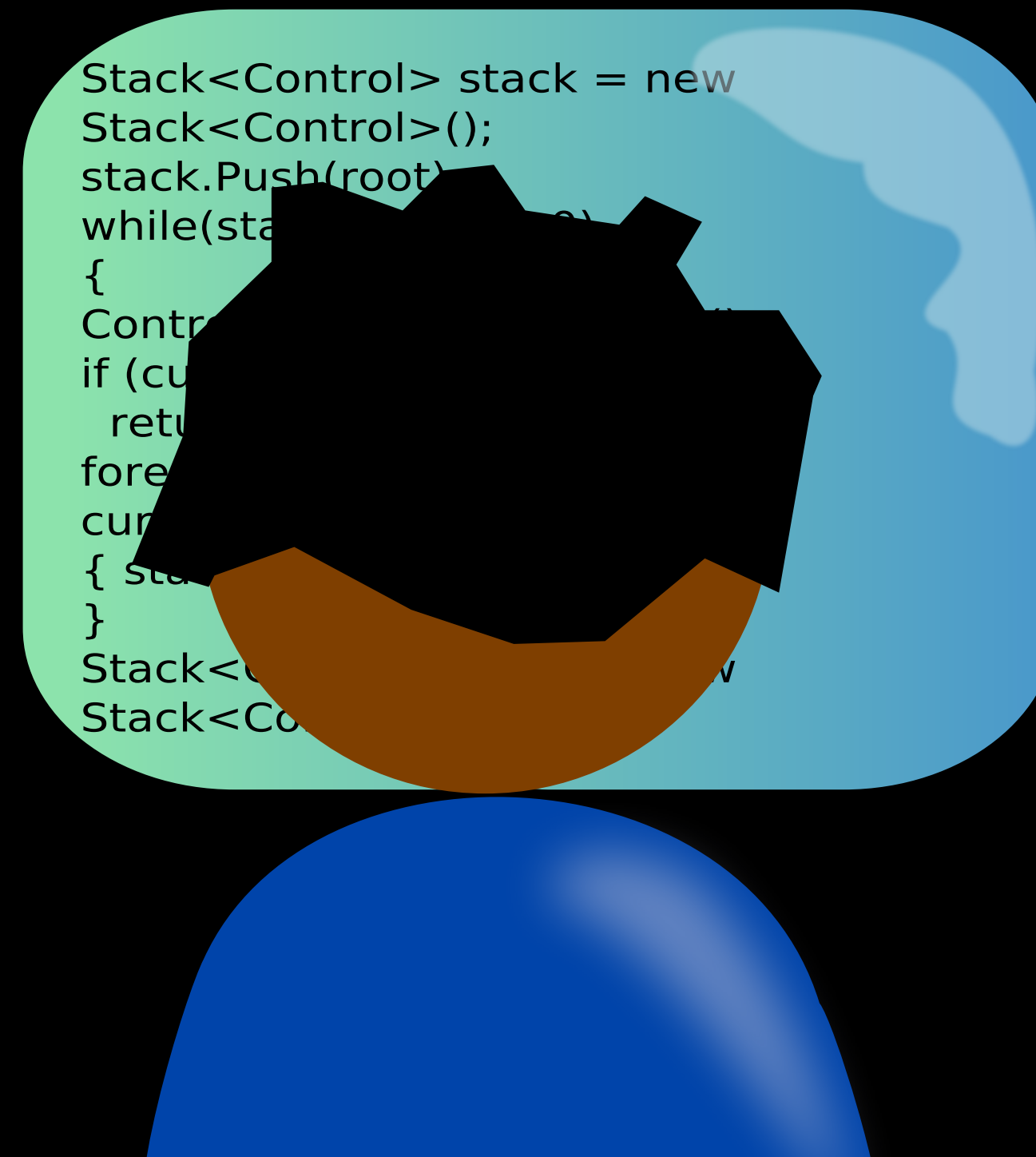
IDE style (IntelliJ)

```
1  /*****
2   * Prints "Hello, World". By tradition, this is everyone's first program.
3   *
4   * These first 6 lines of text are comments. They are not part of the program;
5   * they serve to remind us about its properties.
6   *****/
7
8  public class HelloWorld {
9      public static void main(String[] args) {
10
11          // Prints "Hello, World" in the terminal window.
12          System.out.println("Hello, World");
13      }
14  }
```

compiler format

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

YOU'RE NOW READY TO PROGRAM!



More questions



attend office hours (or stay after lecture)



ask on Ed



Credits

media	source	license
<i>Hello, World</i>	<u>Adobe Stock</u>	<u>education license</u>
<i>Programming</i>	<u>Adobe Stock</u>	<u>education license</u>
<i>Time Enough for Love</i>	<u>Robert A. Heinlein</u>	
<i>Ada Lovelace</i>	<u>Margaret Sarah Carpenter</u>	<u>public domain</u>
<i>Babbage's Analytic Engine</i>	<u>Science Museum, London</u>	<u>CC BY-SA 2.0</u>
<i>Java Logo</i>	<u>Sun Microsystems</u>	
<i>Android Phone</i>	<u>nicepng.com</u>	<u>public domain</u>
<i>Google Data Center</i>	<u>Google / Connie Zhou</u>	
<i>Mars Rover</i>	<u>NASA / JPL / Cornell</u>	<u>public domain</u>
<i>MRI Machine</i>	<u>Adobe Stock</u>	<u>education license</u>
<i>Internet of Things</i>	<u>Adobe Stock</u>	<u>education license</u>

Credits

media	source	license
<i>Jazz Musician Band</i>	<u>Adobe Stock</u>	<u>education license</u>
<i>Programmer</i>	<u>Jaime Botero</u>	<u>public domain</u>
<i>Students Asking Questions</i>	<u>Adobe Stock</u>	<u>education license</u>
<i>Question Marks</i>	<u>pikpng.com</u>	<u>non-commercial use</u>