

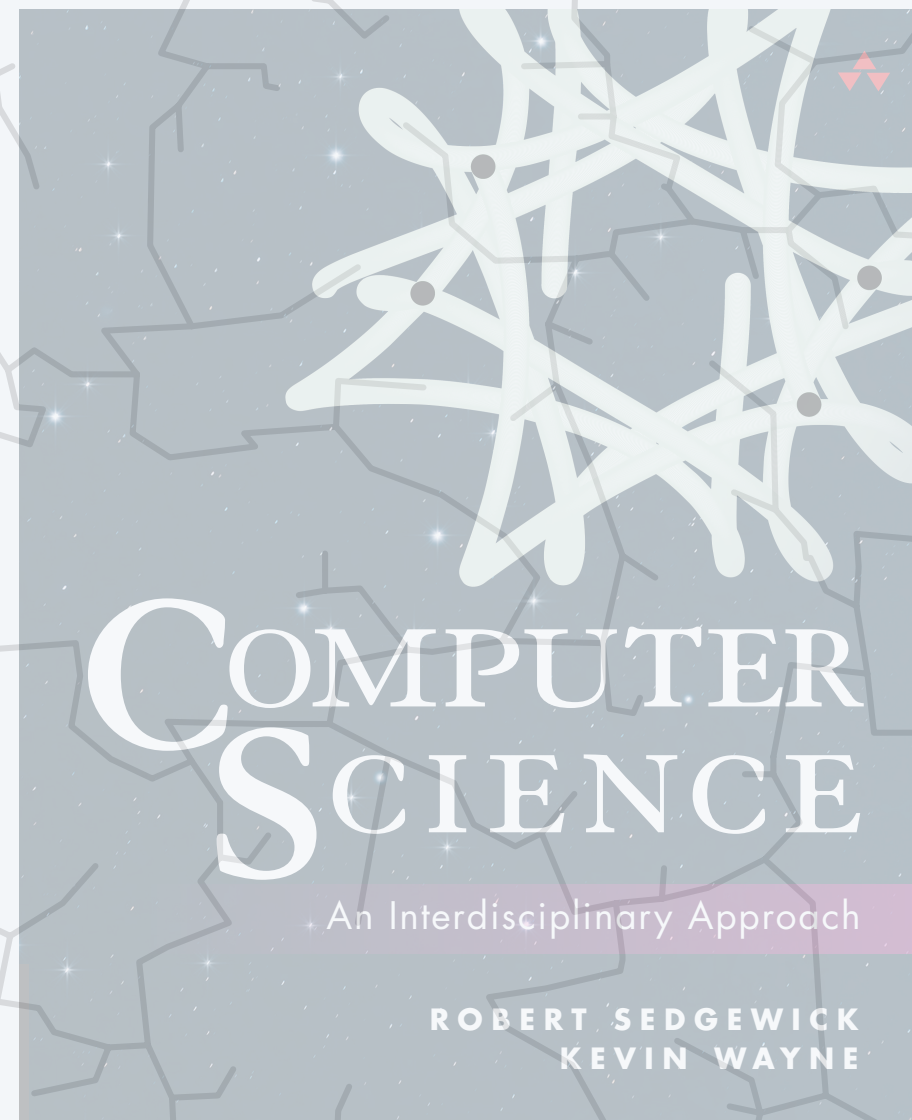
<https://introc.cs.princeton.edu>

1.1

1.1 HELLO, WORLD

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



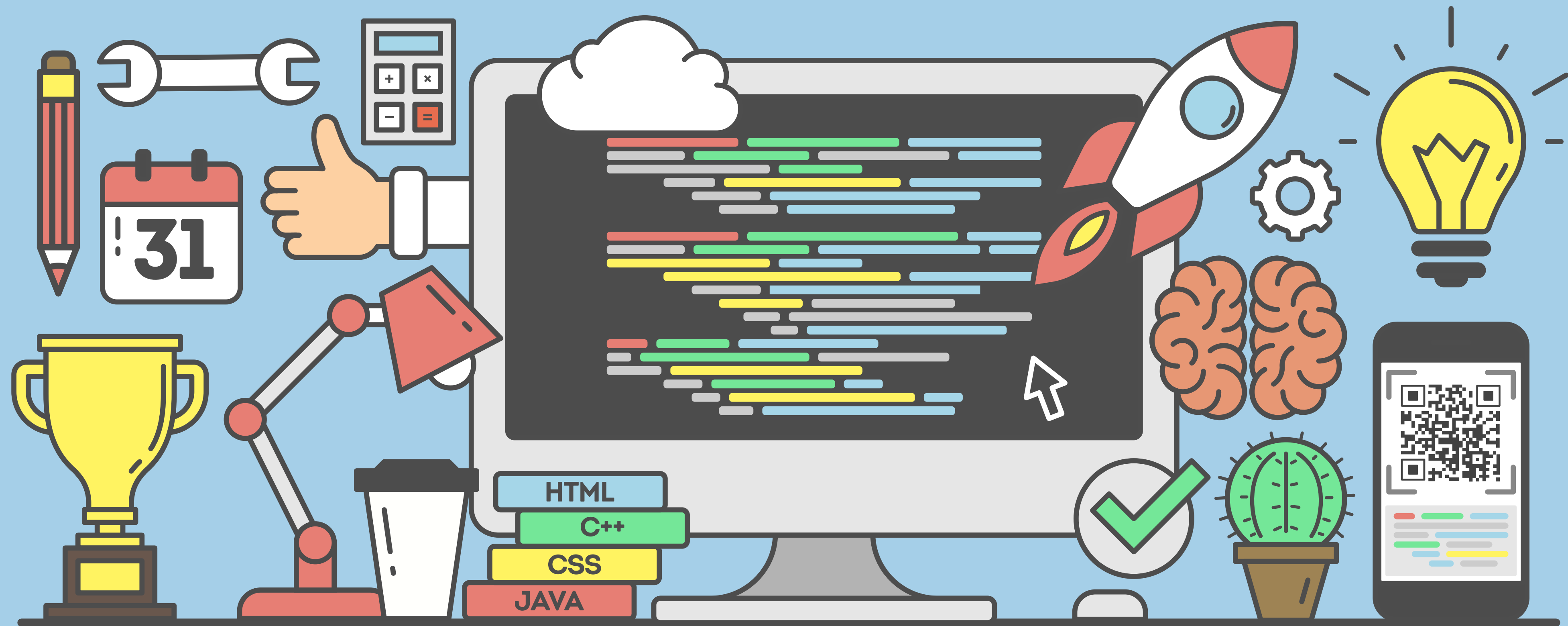


<https://introcs.cs.princeton.edu>

1.1 HELLO, WORLD

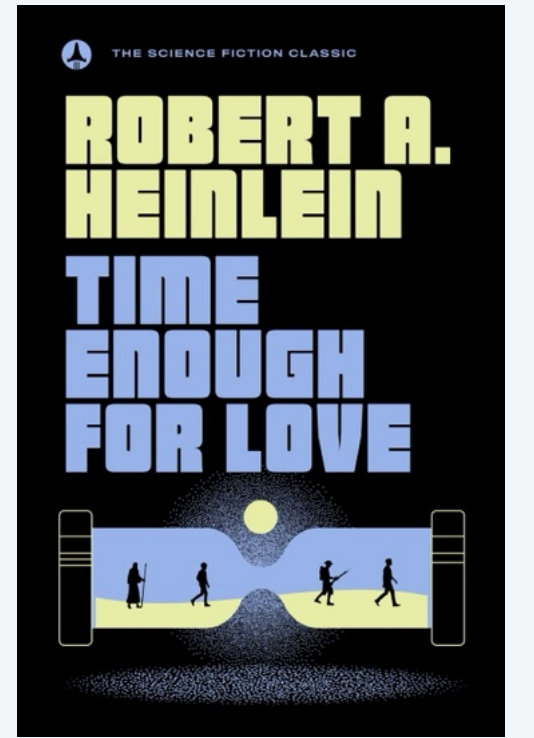
- ▶ *why programming?*
- ▶ *your first program*
- ▶ *program development*

PROGRAMMING



"Time Enough for Love" (1973) by Robert A. Heinlein

A human being should be able to
change a diaper,
plan an invasion,
butcher a hog,
conn a ship,
design a building,
write a sonnet,
balance accounts,
build a wall,
set a bone,
comfort the dying,
take orders,
give orders,
cooperate,
act alone,
solve equations,
analyze a new problem,
pitch manure,
a natural, satisfying, and creative endeavor
(leading to accomplishments not otherwise possible) → **program a computer,**
cook a tasty meal,
fight efficiently,
die gallantly.

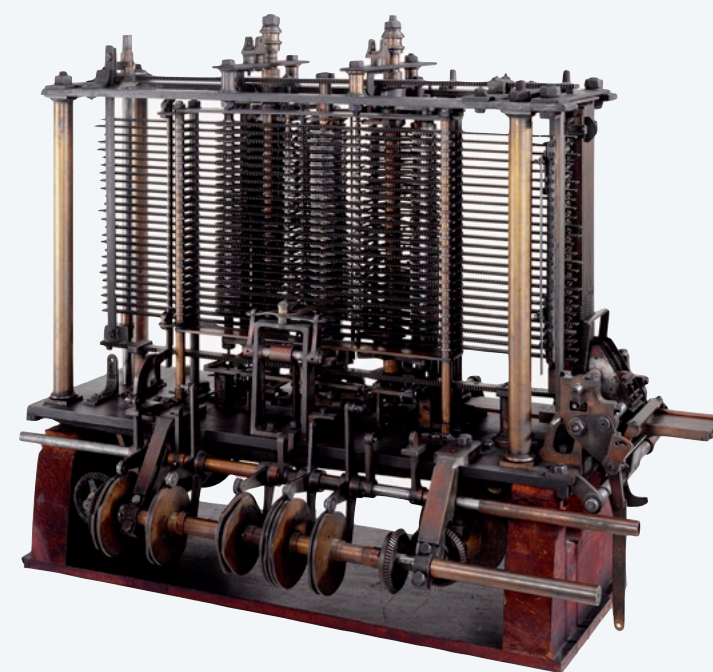


You need to know how to program

Prepackaged solutions (apps). Great when what they do is what you want.



Programming. Empowers **you** to tell a computer what **you** want it to do.



**Analytical Engine
(first computer)**



**Ada Lovelace
(first programmer)**

Telling a computer what to do

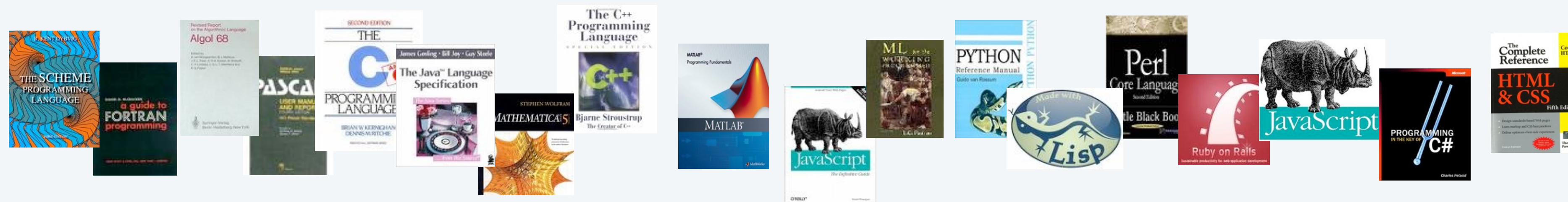
Machine languages. Easy for **computers**; error-prone for people.

```
001111110010000110011011100101110111001110111011110100111111001100011110111111001111011110  
01111011110110001100010000001001110010111001111110011011010001010011100000110000101001000  
111101110111011111000011100010010100001001110000011010100110100001010110001001110010001...
```

Natural languages. Easy for **people**; error-prone for computers. ← *rapid progress in last 2 years (but not as robust as desired)*



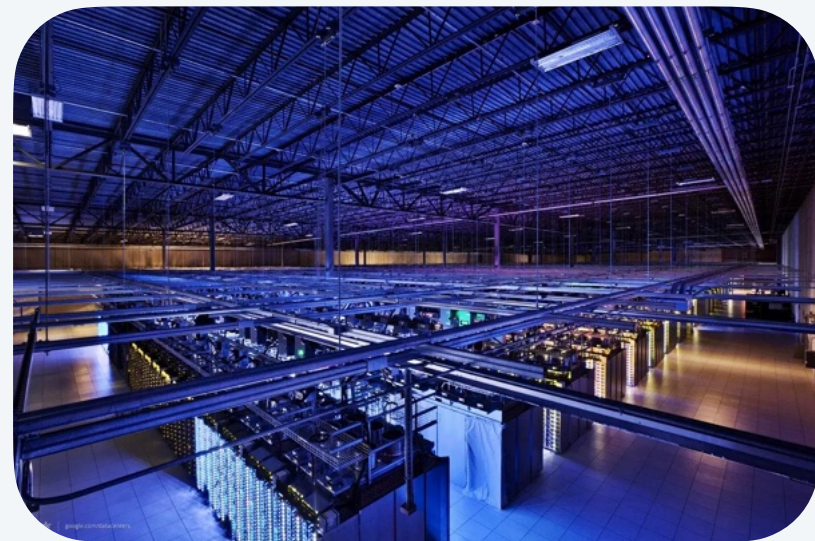
High-level programming languages. Enables **people and computers** to communicate effectively.



Java features.

- Embraces full set of modern abstractions.
- Freely available for macOS, Windows, and Linux.
- Variety of automatic checks for mistakes in programs.
- Widely used: millions of developers; billions of devices. ← *among top 3 languages for past two decades*

Ex. Android phones/TVs, web servers, Mars rover, medical devices, internet of things, ...



Reality. Use different programming languages, depending on domain.

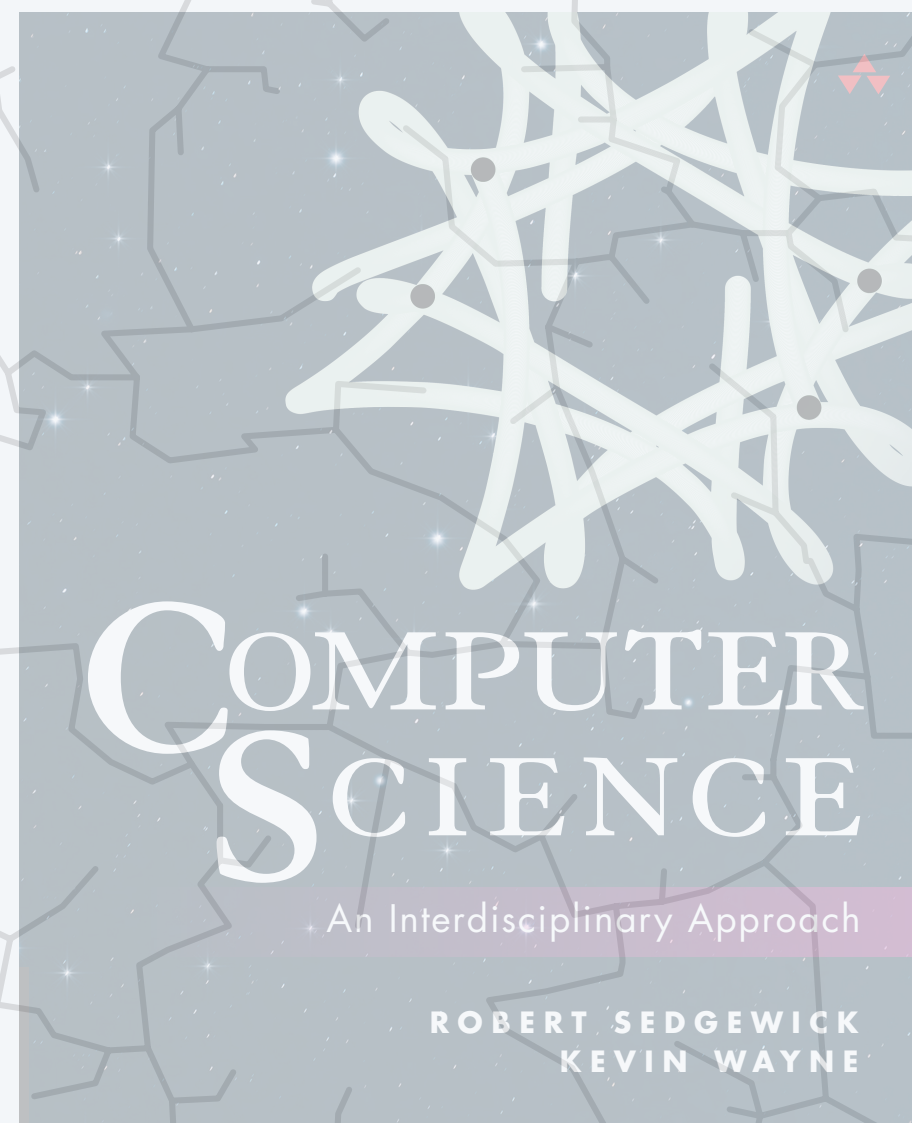
A rich subset of the Java language



Your programs will primarily consist of these plus identifiers (names) that you make up.

← *seems like a lot,
but typical English
vocabulary is 20K words!*

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

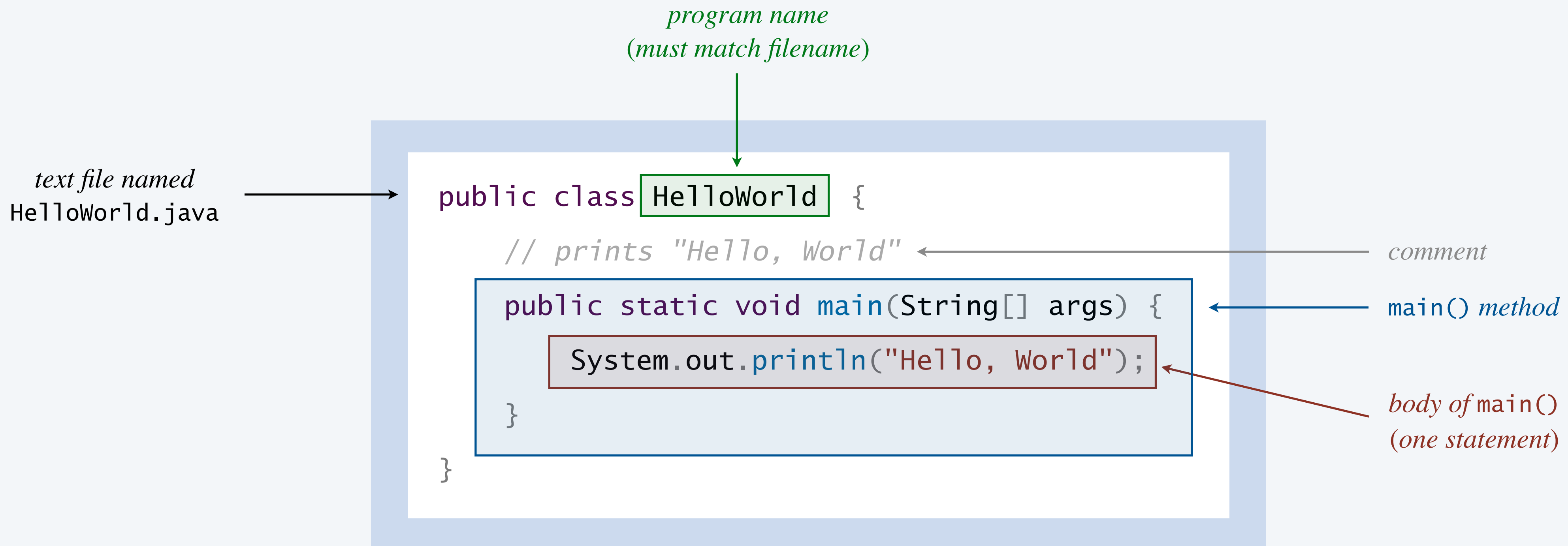


<https://introc.cs.princeton.edu>

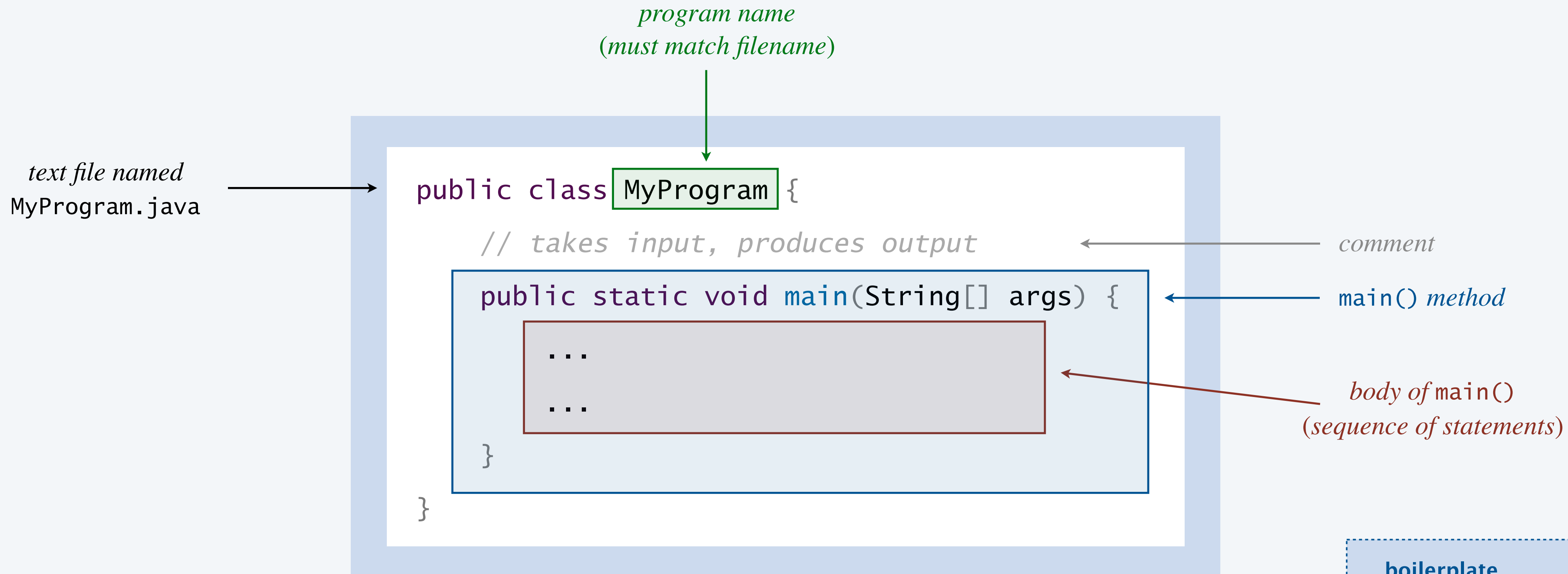
1.1 HELLO, WORLD

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

Anatomy of your first Java program



Anatomy of your first few Java programs



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

boilerplate

- public
- class
- static ← *don't worry, we'll learn their meaning in due time*
- void
- main
- String[]

Hello World with audio output



Standard audio. Our course library for playing 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

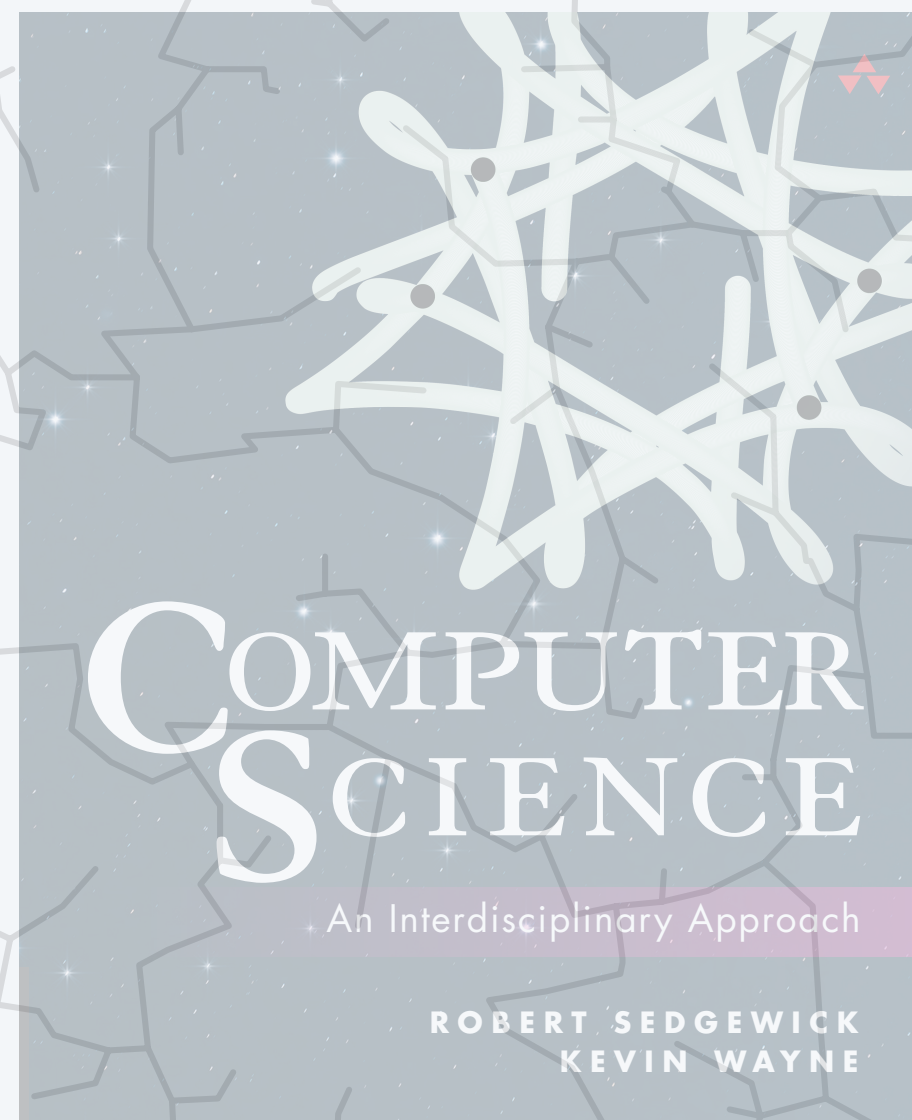
```
~/cos126/hello> javac-introcs HelloWorldWithAudio.java
```

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

```
Hello, World
```

```
🔊 [speaks "Hello, World"]
```

*the javac-introcs and java-introcs commands
tell Java where to find our course libraries*



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

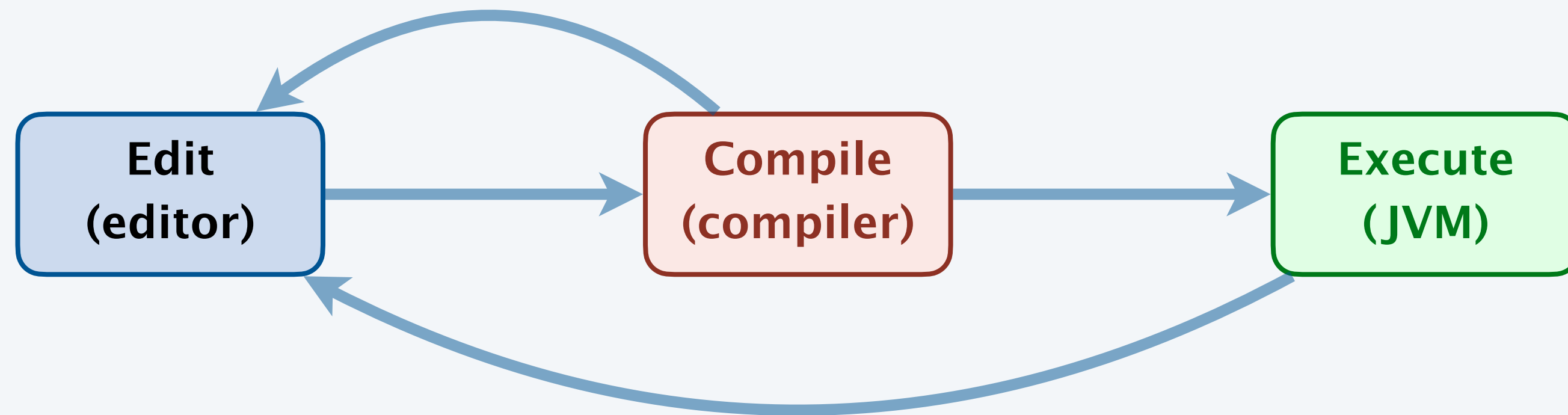
1.1 HELLO, WORLD

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

Program development in Java

Developing a Java program involves three steps:

- **Edit:** write your program.
- **Compile:** create a “machine–language” version of your program.
- **Execute:** run your program, taking input and producing output.



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

Almost always requires cyclic refinement:

- Not a legal Java program (compile–time error) \Rightarrow need to re–edit.
- A legal Java program that does the wrong thing \Rightarrow need to re–edit.

*run–time error or
produces incorrect output*

Coding style

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

Goal. Make it easier for programmers (including you!) to read and understand the code.

textbook

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");
    }
}
```

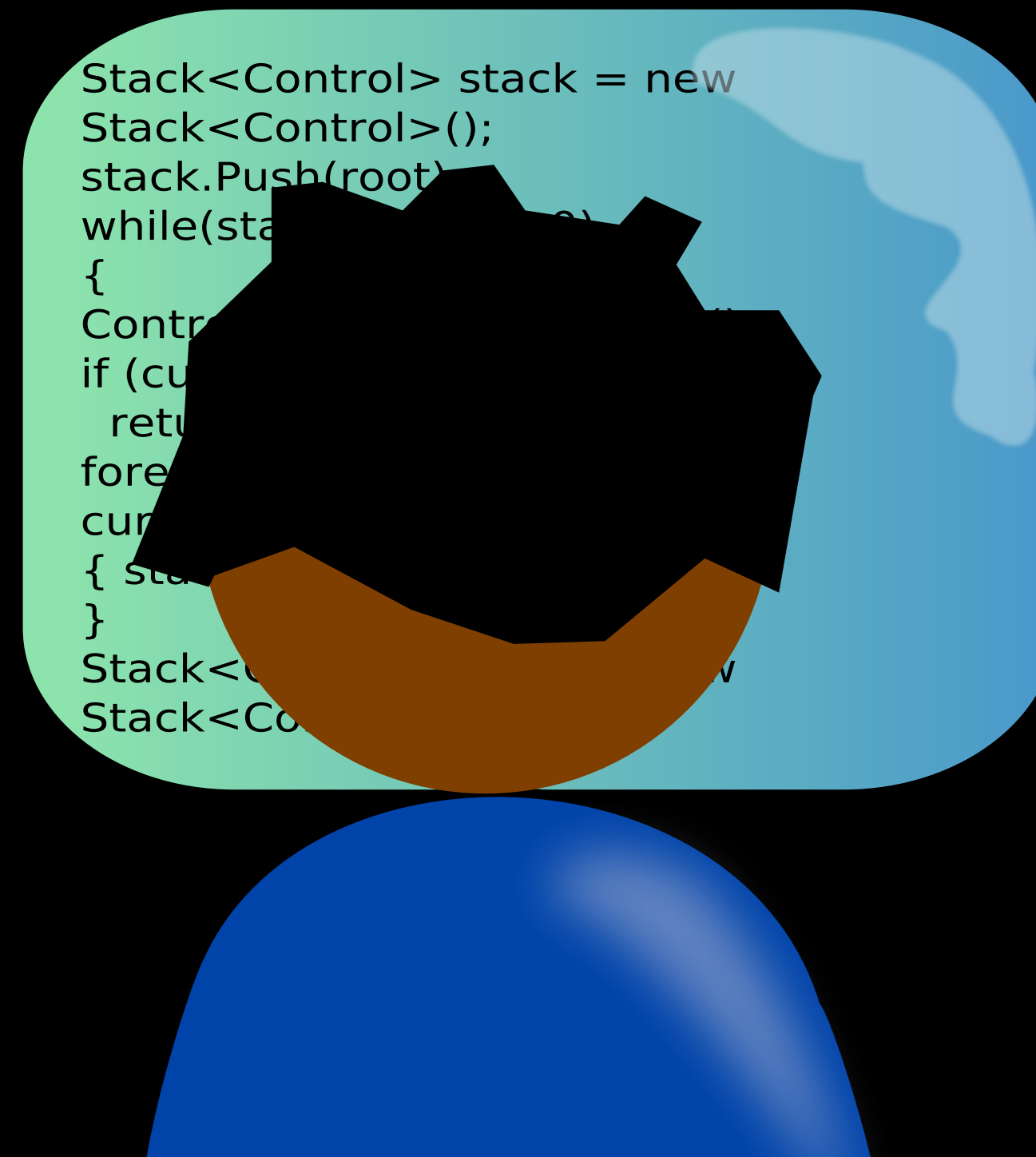
IntelliJ

```
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

```
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>