

Plans, Programs, and Bugs

Introduction to algorithms, programming, and compilers



Instructions for a robot

PAUSE AND TRY

A robot does exactly what you write and nothing else. What goes wrong?

MAKE A BOWL OF CEREAL

1. Pour the cereal.
2. Add some milk.
3. Eat.

With a partner: find two places the robot gets stuck or makes a mess. Rewrite one step so the robot can't get it wrong.

By the end of class you can...

- 1 Explain what an algorithm is and make a vague step precise.
- 2 Show the same algorithm in written steps and as a flowchart.
- 3 Predict what happens when the order of steps changes.
- 4 Describe how code goes from an editor or IDE, through the compiler, to a running program.
- 5 Tell syntax, logic, and run-time errors apart, and say how each one is found.

1

PART 1

What is an algorithm?

Steps a literal follower can carry out



An **algorithm** is a step-by-step process for completing a task or solving a problem. A computer follows it **literally**: it fills in nothing.

Vague step	A literal follower asks...	Precise step
Add some fruit.	<i>How much? Which fruit?</i>	Add 1 cup of strawberries.
Blend it.	<i>For how long? Lid on?</i>	Put the lid on and blend for 30 seconds.
Serve.	<i>Into what?</i>	Pour the smoothie into a glass.

Making a smoothie: same task, but now nobody has to guess.

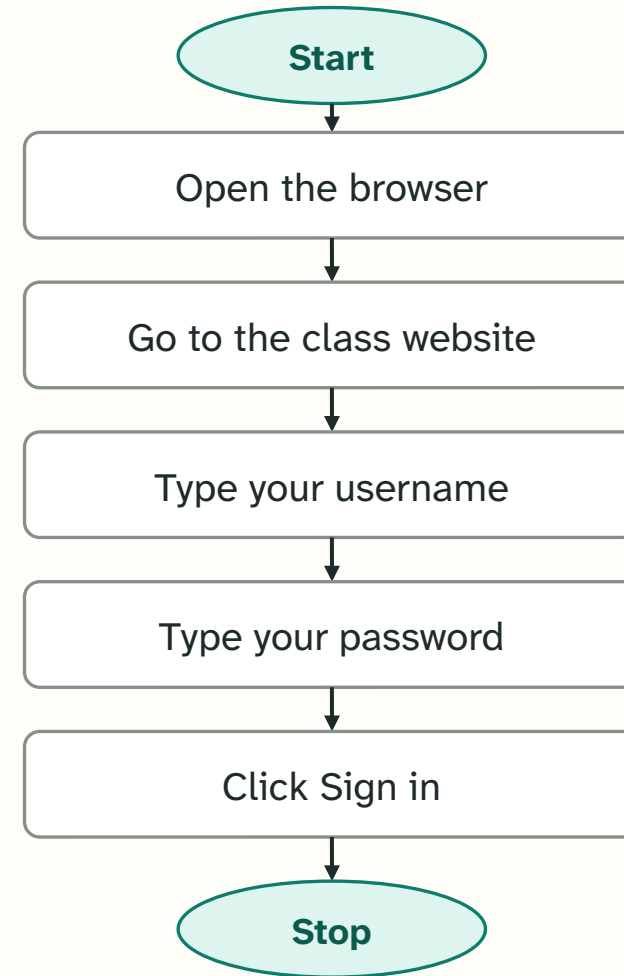
One algorithm, two representations

IN WORDS

1. Open the browser.
2. Go to the class website.
3. Type your username.
4. Type your password.
5. Click Sign in.



Same steps, same order: **same algorithm**, whether it is written in sentences or drawn as boxes and arrows.



A large, bold, teal-colored number '2' centered within a white rounded square with a slight drop shadow.

2

PART 2

Sequencing: order matters

Same steps, different order, different result



Sequencing defines the order in which steps are completed. Steps are completed **one at a time**.

ORIGINAL ORDER

1. Bake the cake.
2. Let it cool.
3. Spread the icing.

A cooled, iced cake.

STEPS 2 AND 3 SWAPPED

1. Bake the cake.
2. Spread the icing.
3. Let it cool.

The icing melts on the hot cake. Same steps, worse result.

The swap test

PAUSE AND TRY

Which two steps can swap safely? Which swap breaks it?

MAKE A CUP OF TEA

1. Get a cup.
2. Get a tea bag.
3. Put the tea bag in the cup.
4. Pour hot water into the cup.

For each step, ask: was everything it needs already done by an earlier step?

A large, white, rounded square containing the number '3' in a bold, teal, sans-serif font.

3

PART 3

From algorithm to program

Your first look at a Java program

WELCOME.JAVA

```
public class Welcome {  
    public static void main(String[] args) {  
        System.out.println("Hi, AP CSA!");  
        System.out.println("Time to code.");  
    }  
}
```

OUTPUT

```
Hi, AP CSA!  
Time to code.
```

Text editor

Any app that edits plain text can hold code.

IDE

Integrated development environment: write, compile, and run in one place.

Statement

One instruction. Statements run one at a time, in order.

Compile first, then run

Write

YOU, IN AN EDITOR OR IDE

Type the source code.

Compile

THE COMPILER

Checks the code for some errors, then translates it.

Run

THE COMPUTER

Carries out the statements one at a time.



Errors the compiler detects must be fixed **before the program can run**. If the compiler finds even one, there is no translated program, so nothing runs.

A large, white, rounded square containing a bold, dark teal number '4'.

4

PART 4

Three kinds of errors

A syntax error breaks Java's rules

POSTER.JAVA

```
public class Poster {  
    public static void main(String[] args) {  
        System.out.println("Art Club today");  
        System.out.println("Room 204");  
    }  
}
```

COMPILER SAYS

```
Poster.java:3: error: unclosed string literal  
    System.out.println("Art Club today");  
                        ^  
1 error
```

Where

Poster.java:3 means line 3.

What

unclosed string literal: a quote was opened but never closed.

Exactly where

The ^ points at the opening quote.

"The first lines are fine, so they'll print"



Compiling covers the **whole** program and happens **before** running. One compiler error anywhere means no line runs.

INSIDE MAIN

```
System.out.println("Period 1: Math");  
System.out.println("Period 2: English");  
System.out.println("Period 3: Science")
```

OUTPUT

```
(nothing)  
error: ';' expected
```

Add the semicolon, compile again, and all three lines appear.

A logic error compiles, runs, and is wrong

SCRAMBLED EGGS, INSIDE MAIN

```
System.out.println("Crack the eggs.");  
System.out.println("Cook the eggs.");  
System.out.println("Whisk the eggs.");
```

Expected outcome	Actual output
Crack the eggs.	Crack the eggs.
Whisk the eggs.	Cook the eggs.
Cook the eggs.	Whisk the eggs.



A **logic error** makes a program behave incorrectly or unexpectedly. The compiler can't see it. **Testing** finds it: run the program with specific data and check for the expected outcome.

THE FIX

Swap the last two statements. Now the actual output matches the expected outcome.

A run-time error stops a running program

INSIDE MAIN

```
System.out.println("Movie night!");  
System.out.println("Splitting 12 snacks...");  
System.out.println(12 / 0);  
System.out.println("Enjoy the show!");
```



A **run-time error** happens while the program runs and usually ends it abnormally. An **exception** is one the compiler did not catch. It interrupts the normal flow.

OUTPUT

```
Movie night!  
Splitting 12 snacks...  
Exception in thread "main" java.lang.ArithmeticException: / by zero
```

Three errors, three moments

	Syntax error	Run-time error	Logic error
Does it compile?	No	Yes	Yes
Does it start running?	No	Yes	Yes
How does it end?	Never runs	Stops early	Finishes with a wrong result
What finds it?	The compiler	Running it: an exception message	Testing against the expected outcome
Example	Missing ;	Dividing by zero	Steps in the wrong order

Name that error

PAUSE AND TRY

Syntax, run-time, or logic? How can you tell?

1

You click Run. The only thing on the screen is:
error: cannot find symbol

2

Two lines appear, then:
Exception in thread "main"

3

The program finishes with no messages, but the total it prints is 10 points short.

Words to know

Algorithm

A step-by-step process to complete a task or solve a problem.

Sequencing

The order steps are completed in, one at a time.

Compiler

Checks code for some errors and translates it so it can run.

Syntax error

Breaks the language's rules. The compiler finds it.

Logic error

Runs but behaves incorrectly. Testing finds it.

Exception

A run-time error the compiler missed. It interrupts the program.

Before, during, or after?



Explain to your partner when each kind of error is found, and what finds it.

SENTENCE STARTER

A syntax error is found ____ the program runs, by _____. A logic error is found by _____.

Next: the Plans, Programs, and Bugs puzzle worksheet.