

Exercise walk-through

1.1

Introduction to Algorithms, Programming, and Compilers

AP Computer Science A

Integer division truncates, and the cast position decides

the trap

`7 / 2` is 3, not 3.5 – two ints divide to an int

the fix

cast **one operand** first: `(double) 7 / 2` is 3.5

the wrong fix

`(double) (7 / 2)` is 3.0 – the truncation already happened inside the brackets

so

the **position of the cast** decides whether the truncation happens at all

escape sequence

`\n` and `\t` inside a String – a backslash changes what the next character means

Watch for integer division producing a truncated result where a decimal was wanted. It compiles, it runs, and it is silently wrong – a **logic error**.

Class methods, instance methods, and where code comes from

class (static) method

belongs to the [class itself](#), so you call it on the class name: `Math.abs(x)`. [No object needed](#)

instance method

belongs to an [object](#), so you call it on a reference: `s.length()`

a library

pre-written classes you can use; Java's is the [API](#)

compiled

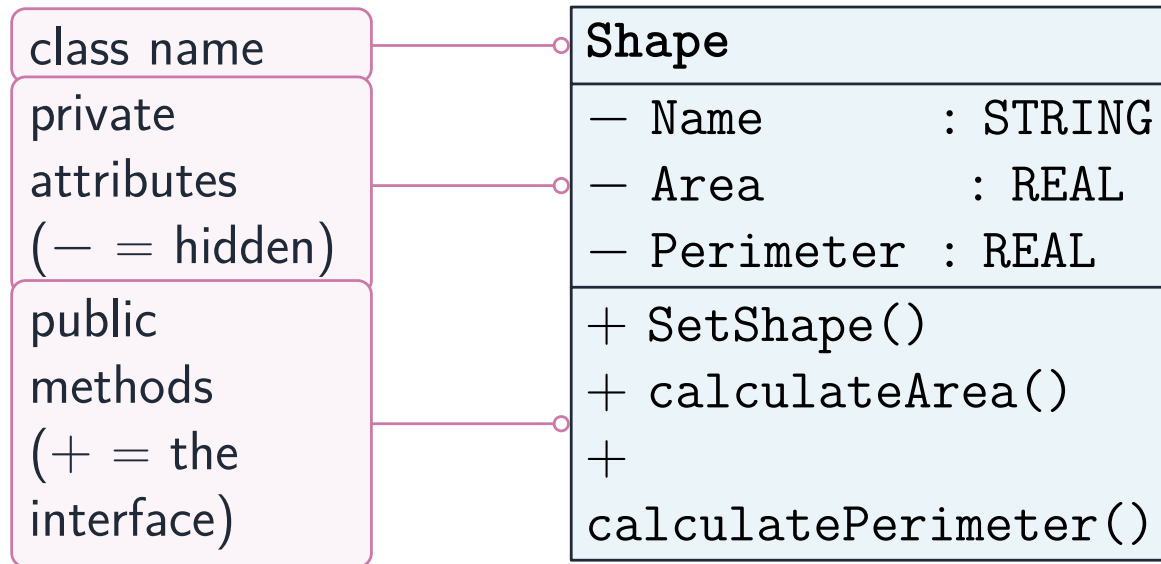
Java source is compiled once to bytecode, then run – so syntax errors surface [before](#) the program starts

abstraction

using a method by its name and contract, without knowing how it works inside

If you find yourself creating an object just to call a method on it, check whether the method is `static` – `Math.sqrt` on a new `Math()` does not compile.

Methods, the API & the Math class, in detail



A class diagram: private attributes and public methods

Exercises 2.1-2.2

2.1 State **two** properties of a good algorithm.

[2]

2.2 Name the error in each: (i) `System.out.println("Hi")` with no semicolon; (ii) a loop meant to print 1 to 10 prints 1 to 9; (iii) `int x = 5 / 0;`.

[3]

Exercises 2.1-2.2

2.1 State **two** properties of a good algorithm.

[2]

2.1

- any two: each step is precise and unambiguous; the steps are in order; it finishes after a finite number of steps; it solves the problem

2.2 Name the error in each: (i) `System.out.println("Hi")` with no semicolon; (ii) a loop meant to print 1 to 10 prints 1 to 9; (iii) `int x = 5 / 0;`.

[3]

2.2

- (i) **syntax**
- (ii) **logic**
- (iii) **run-time**

Exercises 2.3-3.1

2.3 Describe what the compiler does with a Java source file.

[2]

3.1 Which error stops a program from compiling?

[1]

☐

- A** a logic error
- B** a syntax error
- C** a run-time error
- D** an infinite loop

Exercises 2.3-3.1

2.3 Describe what the compiler does with a Java source file.

[2]

2.3

- checks the code for syntax errors
- translates it into bytecode that the JVM can run

3.1 Which error stops a program from compiling?

[1]

☐

- A** a logic error
- B** a syntax error
- C** a run-time error
- D** an infinite loop

3.1 B

- the compiler rejects code that breaks Java's rules

Exercises 3.2-3.3

3.2 Where does a Java program start running?

[1]

☐

A at the first line of the file

B at the first method in the class

C at the `main` method

D at the class header

3.3 A program compiles and runs, but prints the area of a circle as 0. What kind of error is most likely?

[1]

☐

A syntax

B compile-time

C logic

D none: 0 is correct

Exercises 3.2-3.3

3.2 Where does a Java program start running?

[1]

☐

- A at the first line of the file
- B at the first method in the class
- C at the `main` method
- D at the class header

3.2 C

- the JVM calls `main`

3.3 A program compiles and runs, but prints the area of a circle as 0. What kind of error is most likely?

[1]

☐

- A syntax
- B compile-time
- C logic
- D none: 0 is correct

3.3 C

- it runs but gives a wrong result

Setup for exercise 3.4

3.4 A student writes this program to print the change from a 50-dollar note after a purchase of `cost` dollars:

```
public class Change
{
    public static void main(String[] args)
    {
        int cost = 32
        System.out.println(cost - 50);
    }
}
```

Exercise 3.4 (a)

(a) Identify the syntax error and correct it.

[2]

Exercise 3.4 (a)

(a) Identify the syntax error and correct it.

[2]

(a)

- missing ; after `int cost = 32`
- `int cost = 32;`

Exercise 3.4 (b)

(b) After that fix, the program prints `-18`. Identify the error type, and correct the line. [2]

Exercise 3.4 (b)

(b) After that fix, the program prints -18. Identify the error type, and correct the line. [2]

(b)

- a **logic** error
- `System.out.println(50 - cost);`

Exercise 3.4 (c)

(c) Describe **one** test the student should run, giving the input and expected output. [2]

Exercise 3.4 (c)

(c) Describe **one** test the student should run, giving the input and expected output. [2]

(c)

- e.g. `cost 32` should print 18; or a boundary such as `cost 50` prints 0