

Exercise walk-through

1.15

String Manipulation

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

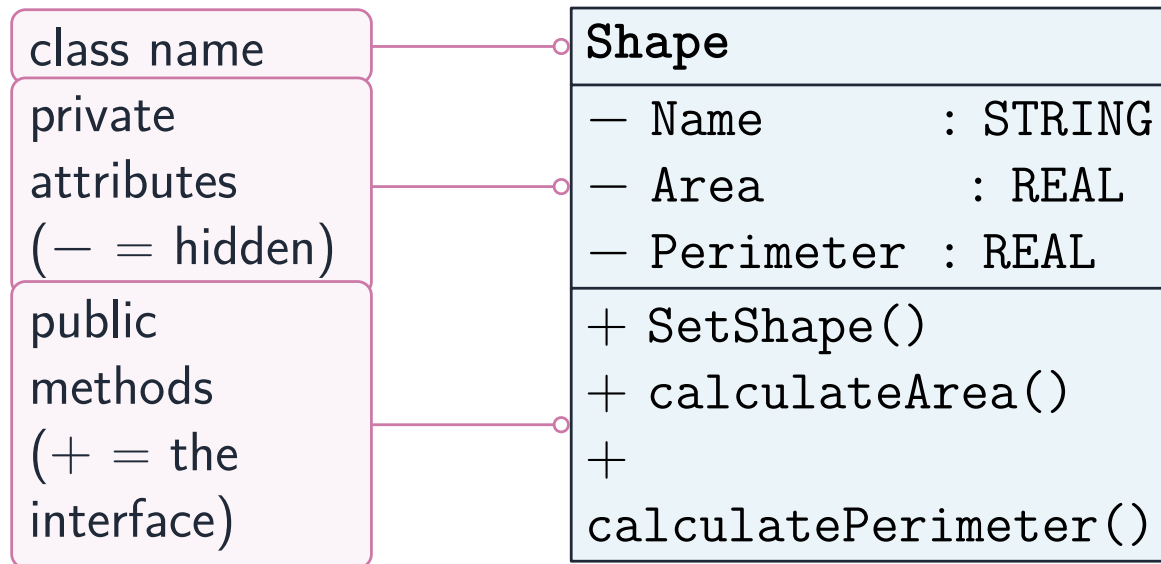
the **position of the cast** decides whether the truncation happens at all

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**.

Methods, the API & the Math class, in detail



A class diagram: private attributes and public methods

Check yourself

- 1 "COMPUTER".substring(0,4) gives what?
- 2 What does indexOf return when the string is absent?
- 3 Is int a primitive or a reference type?
- 4 What does the compiler turn source code into?

Answers 1. "COMP" 2. -1 3. primitive 4. bytecode, run by the JVM

Exercises 2.1-2.2

2.1 String `s = "ELEPHANT"`; State (i) `s.length()`; (ii) `s.substring(2, 5)`; (iii) `s.indexOf('H')`; (iv) `s.substring(6)`. [4]

2.2 State whether `"apple".compareTo("banana")` is negative, zero or positive, and why. [1]

Exercises 2.1-2.2

2.1 String `s = "ELEPHANT"`; State (i) `s.length()`; (ii) `s.substring(2, 5)`; (iii) `s.indexOf('H')`; (iv) `s.substring(6)`. [4]

2.1

- (i) 8
- (ii) indexes 2 to 4: "EPH"
- (iii) 0
- (iv) "NT"

2.2 State whether `"apple".compareTo("banana")` is negative, zero or positive, and why. [1]

2.2

- **negative:** "apple" comes before "banana" in dictionary order

Exercises 2.3-3.1

2.3 Write an expression for the **last** character of any non-empty `String` `w`. [1]

3.1 What is `"PROGRAM".substring(3)`? [1]

☐

A "PRO"

B "GRAM"

C "OGRAM"

D "G"

Exercises 2.3-3.1

2.3 Write an expression for the **last** character of any non-empty `String` `w`. [1]

2.3

- `w.substring(w.length() - 1)`

3.1 What is `"PROGRAM".substring(3)`? [1]

☐

A "PRO"

B "GRAM"

C "OGRAM"

D "G"

3.1 B

- from index 3, G, to the end

Exercises 3.2-3.3

3.2 `String a = "java"; a.toUpperCase(); System.out.println(a + a.length());`

What is printed?

[1]

☐

A JAVA4

B java4

C java5

D JAVA

3.3 Which call throws an exception for `String s = "code";`?

[1]

☐

A `s.substring(4)`

B `s.substring(1, 4)`

C `s.substring(2, 5)`

D `s.indexOf("z")`

Exercises 3.2-3.3

3.2 `String a = "java"; a.toUpperCase(); System.out.println(a + a.length());`

What is printed?

[1]

☐

A JAVA4

B java4

C java5

D JAVA

3.2 B

- the result of `toUpperCase` is not stored

3.3 Which call throws an exception for `String s = "code";`?

[1]

☐

A `s.substring(4)`

B `s.substring(1, 4)`

C `s.substring(2, 5)`

D `s.indexOf("z")`

3.3 C

- the end index 5 is past `length()`, 4

Exercise 3.4 (a)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(a) Write an expression for the part **before** the `@`.

[2]

Exercise 3.4 (a)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(a) Write an expression for the part **before** the `@`.

[2]

(a)

- `email.substring(0, email.indexOf("@"))`

Exercise 3.4 (b)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(b) Write an expression for the part **after** the `@`.

[1]

Exercise 3.4 (b)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(b) Write an expression for the part **after** the `@`.

[1]

(b)

- `email.substring(email.indexOf("@") + 1)`

Exercise 3.4 (c)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(c) Give the values of both expressions for `"li.wei@school.cn"`.

[2]

Exercise 3.4 (c)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(c) Give the values of both expressions for `"li.wei@school.cn"`. [2]

(c)

- `"li.wei"`
- `"school.cn"`

Exercise 3.4 (d)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(d) Write a Boolean expression that is `true` when `email` contains no `@`.

[1]

Exercise 3.4 (d)

3.4 An email address is stored in `String email`, such as `"li.wei@school.cn"`.

(d) Write a Boolean expression that is `true` when `email` contains no `@`. [1]

(d)

- `email.indexOf("@") == -1`