

Exercise walk-through

4.8

ArrayList Methods

AP Computer Science A

The ArrayList toolbox

An **ArrayList** grows and shrinks, unlike a fixed array. It stores objects, so numbers are **wrapped** (Integer, Double).

add(x)
append

get(i)
read

set(i,x)
replace

remove(i)
delete

Traverse it with a for or an enhanced for-each loop: `for (int x : list).`

Removing from a list while looping over it

the problem

`remove(i)` shifts every later element down, so the element after the removed one is skipped

fix one

loop backwards, from `size()-1` down to 0 – later shifts cannot affect indices you have passed

fix two

do **not** increment `i` after a removal, so the next pass re-examines the same index

never

remove inside an enhanced for loop – it throws a `ConcurrentModificationException`

the rest

the array algorithms carry over unchanged, plus **insertion** and **deletion**, which arrays cannot do easily

This is the single most-tested ArrayList point. If a question removes items in a loop, it is testing exactly this.

Exercises 2.1-2.3

2.1 State one way an `ArrayList` differs from an array.

[1]

2.2 Name the method that returns the number of elements.

[1]

2.3 State what `list.get(0)` returns.

[1]

Exercises 2.1-2.3

2.1 State one way an `ArrayList` differs from an array. [1]

2.1

- an `ArrayList` can grow and shrink, while an array has a fixed size
-

2.2 Name the method that returns the number of elements. [1]

2.2

- `size()`
-

2.3 State what `list.get(0)` returns. [1]

2.3

- the first element of the list

Exercises 3.1-3.2

3.1 To add an element to an ArrayList, you use

[1]

☐

A add

B length

C new

D size

☐

A .length

B .size()

C .count

D .get()

Exercises 3.1-3.2

3.1 To add an element to an ArrayList, you use

[1]

- ☐ A `add`
☐ B `length`
☐ C `new`
☐ D `size`

3.1 A

- `add`.

3.2 The number of elements in an ArrayList is found with

[1]

- ☐ A `.length`
☐ B `.size()`
☐ C `.count`
☐ D `.get()`

3.2 B

- `.size()`.

Exercise 3.3 (a)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(a) State `a.size()`.

[1]

Exercise 3.3 (a)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(a) State `a.size()`.

[1]

(a)

- 2

Exercise 3.3 (b)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(b) State `a.get(1)`.

[1]

Exercise 3.3 (b)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(b) State `a.get(1)`.

[1]

(b)

- "bye"

Exercise 3.3 (c)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(c) State whether an `ArrayList` can grow.

[1]

Exercise 3.3 (c)

3.3 `ArrayList<String> a = new ArrayList<>(); a.add("hi"); a.add("bye");`

(c) State whether an `ArrayList` can grow.

[1]

(c)

- yes