Test 6

- When: Friday March 14, during the lab (14:30–16:00)
- Where: Lassonde building, labs 1006, 1004, 1002
- Material: Chapter 1–8 of the textbook, with a focus on Chapter 8
- What: One programming question (8 marks) and two short answer questions (2 marks)
- Note: For the programming question, you get 1 mark for the fact that your code compiles
- Note: Your code is not only marked for correctness but also style (1 mark)

- Monday: review session by a teaching assistant on loops.
- Wednesday: no lecture.
- Friday: lab.

Review session on aggregation in LSB 106, 17:00-19:00.



A Classroom has a collection of Students.

May a list contain duplicates?

May a list contain duplicates?

Answer

Yes.

moodle.yorku.ca CSE 1020

May a list contain duplicates?

Answer

Yes.

Question

Are the elements of a list ordered?

May a list contain duplicates?

Answer

Yes.

Question

Are the elements of a list ordered?

Answer

Yes.



Score for each test







The list is implemented by means of an array.



The list is implemented by means of a "links."



The list is implemented by means of an array and multiple threads can manipulate the list at the same time.

These different lists can be classified based on

- the type of the elements of the list (Integer, Double, Boolean, ...) and
- the way the list is implemented (using an array, using "links," ...).

To abstract from the type of the elements of the list, we exploit generics.



E is a type parameter. The elements of the list are of type E.

To abstract from the way the list is implemented, we exploit interfaces.



interface specification what? class implementation how?

```
final int LECTURES = 24;
List<Integer> attendance =
    new ArrayList<Integer>(LECTURES);
```

- The type of the elements is Integer and
- the list is implemented by means of an array.

```
final int LECTURES = 24;
List<Integer> attendance =
    new ArrayList<Integer>(LECTURES);
```

- The type of the elements is Integer and
- the list is implemented by means of an array.

Question

Why can we assign an object of type ArrayList<Integer> to a variable of type List<Integer>?

```
final int LECTURES = 24;
List<Integer> attendance =
    new ArrayList<Integer>(LECTURES);
```

- The type of the elements is Integer and
- the list is implemented by means of an array.

Question

Why can we assign an object of type ArrayList<Integer> to a variable of type List<Integer>?

Answer

Because the class ArrayList<E> implements the interface List<E>.

List<Double> tests = new LinkedList<Double>();

- The type of the elements is Double and
- the list is implemented by means of "links."

ArrayList, LinkedList or Vector?

Depends on which operations on the list are performed.

Question

How many milliseconds does it take to add n elements to the end of a list?

ArrayList, LinkedList or Vector?

Depends on which operations on the list are performed.

Question

How many milliseconds does it take to add n elements to the end of a list?

Answer			
n	ArrayList	LinkedList	Vector
10 ⁵	9	12	14
10 ⁶	47	92	113
10^{7}	442	824	1041
$2 imes 10^7$	913	1,650	2,076
$3 imes 10^7$	1,350	143,616	3,230
$4 imes 10^7$	2,527		4,103
$5 imes 10^7$	2,689		6,119

- Adding to or deleting from the beginning of a list takes
 LinkedList O(1), whereas it takes ArrayList and Vector
 O(n) where n is the size of the list.
- Adding and deleting while traversing a list takes LinkedList O(1), whereas it takes ArrayList and Vector O(n) where n is the size of the list.
- In most other cases, ArrayList outperforms LinkedList and Vector.

What does "adding to the beginning of a list takes LinkedList O(1)" mean?

What does "adding to the beginning of a list takes LinkedList O(1)" mean?

Let $a : \mathbb{N} \to \mathbb{N}$ be the function that given the size of the list ℓ return the number $a(\ell)$ of basic instructions that need to be executed to add an element to the beginning of the list.

What does "adding to the beginning of a list takes LinkedList O(1)" mean?

Let $a : \mathbb{N} \to \mathbb{N}$ be the function that given the size of the list ℓ return the number $a(\ell)$ of basic instructions that need to be executed to add an element to the beginning of the list.

Answer $a \in O(1).$



What does $a \in O(1)$ mean?



What does $a \in O(1)$ mean?

Answer

$$\exists M \in \mathbb{N} : \exists F \in \mathbb{N} : \forall \ell \geq M : a(\ell) \leq F \times 1.$$

There exist M and F so that adding an element to the beginning of the list of size at least M takes at most F basic instructions. That is, it takes (at most) a constant number of instructions.



final int COLUMNS = 8; List<Piece> row = new ArrayList<Piece>(COLUMNS);

- The type of the elements is Piece and
- the list is implemented by means of an array.

List <e></e>				
≪interface≫				
add(E) : boolean				
add(int, E)				
contains(E) : boolean				
get(int) : E				
<pre>iterator() : Iterator<e></e></pre>				
remove(int) : E				
set(int, E) : E				
size() : int				

Create an empty row of a chess board.

Answer

```
final int COLUMNS = 8;
List<Piece> row = new ArrayList<Piece>(COLUMNS);
for (int c = 0; c < COLUMNS; c++) {
  row.add(null);
}
```

Place a black rook on the first and the last square of the row.



Answer

Rook rook = new Rook(Color.BLACK); row.set(0, rook); row.set(COLUMNS - 1, rook);

Place a white pawn on each square of the row.

Answer

```
Pawn pawn = new Pawn(Color.WHITE);
for (int c = 0; c < COLUMNS; c++) {
  row.set(c, pawn);
}
```

Print the row.

An empty square is represented by two spaces. A non-empty square is represented by the representation of the piece on that square. For example, a black king is represented by BK and a white queen is represented by WQ.

The squares are separated by a single space.

Answer

```
StringBuffer representation = new StringBuffer();
for (Piece piece : row) {
    if (piece == null) {
        representation.append(" ");
    } else {
        representation.append(piece.toString());
    }
    representation.append(" ");
}
output.println(representation.toString());
```

• Study Section 10.1 of the textbook.