Programming for Mobile Computing

EECS 1022

moodle.yorku.ca

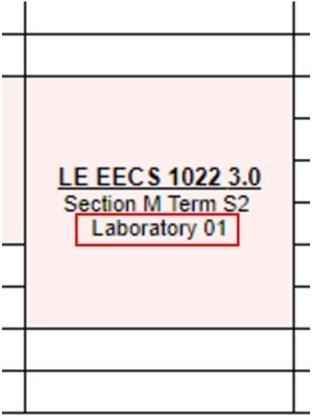
Final exam (programming part)

Final exam (programming part) will take place on Thursday July 27 during lab time. In particular,

- Lab 01: 20:00-21:15
- Lab 02: 18:30-19:45
- Lab 03: 15:30-16:45

Final exam (programming part)

To find out in which lab you are officially registered, please check your online lecture schedule.



LE EECS 1022 3.0

Section M Term S2

Laboratory 01

Final exam (programming part)

- Question 1: 10 marks, partial marks awarded
- Question 2: 5 marks, partial marks awarded
- Question 3: 5 marks, partial marks awarded
- Question 4: 2.5 marks, no partial marks awarded
- Question 5: 2.5 marks, no partial marks awarded
- Question 6: 2.5 marks, no partial marks awarded
- Question 7: 2.5 marks, no partial marks awarded

Extra labs

- Tuesday August 1, 17:30-19:30
- Thursday August 3, 17:30-19:30

Search

Question

Consider

```
List<String> words = ...;
String search = ...;
boolean found = false;
" for each word in the list words"
{
    found = word.equals(seach) || found;
}
```

How do we express "for each word in the list words"?

Search

Question

Consider

```
List<String> words = ...;
String search = ...;
boolean found = false;
“ for each word in the list words”
{
    found = word.equals(seach) || found;
}
```

How do we express “for each word in the list words”?

Answer

Let us check the API of the List interface: `https://docs.oracle.com/javase/8/docs/api/java/util/List.html`

Iterator

“for each word in the list words” can be expressed as

```
Iterator<String> iterator = words.iterator();
while (iterator.hasNext())
{
  String word = iterator.next();
   ...
}
```

Advanced for-loop

```
Iterator<String> iterator = words.iterator();
while (iterator.hasNext())
{
  String word = iterator.next();
   ...
}
```

can be abbreviated to

```
for (String word : words)
{
   ...
}
```

Search

Question

Consider

```
List<String> words = ...;
String search = ...;
boolean found = false;
for (String word : words)
{
    found = word.equals(seach) || found;
}
```

Given that the list contains *n* elements, how many times is the method equals invoked?

Search

Question

Consider

```
List<String> words = ...;
String search = ...;
boolean found = false;
for (String word : words)
{
    found = word.equals(seach) || found;
}
```

Given that the list contains n elements, how many times is the method equals invoked?

Answer

n times.

Binary search

```
int index = Collections.binarySearch(list, element);
```

- The list must be sorted.
- If the element is contained in the list then the method returns the index at which the element can be found.
- If the element is not in the list then the method returns -1.

Binary search

```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```

1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

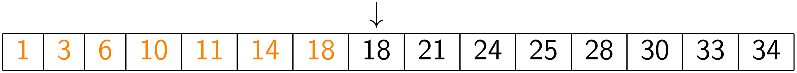
Binary search

```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```

							↓							
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

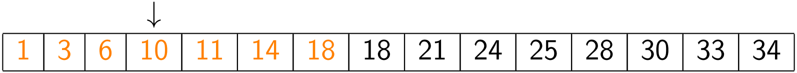
```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```



							↓							
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

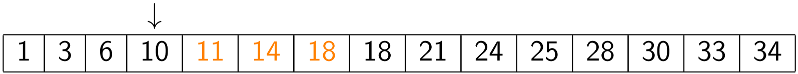
```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```



			↓											
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

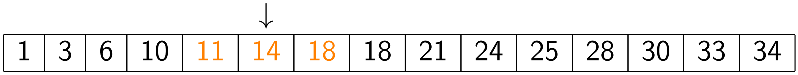
```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```



			↓											
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```



1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```

					↓									
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 11;
int index = Collections.binarySearch(list, ELEMENT);
```

				↓										
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

`index` gets assigned the value 4.

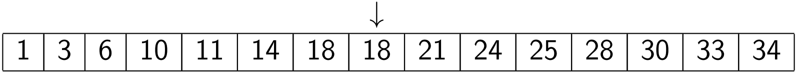
Binary search

```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```

1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

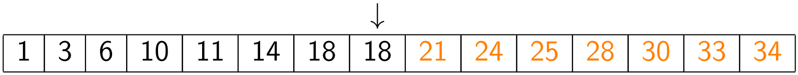
```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```



							↓							
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

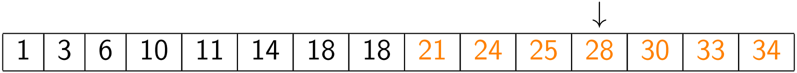
```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```



							↓							
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```



											↓			
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```

											↓			
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

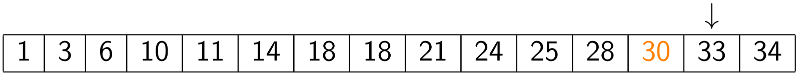
Binary search

```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```

													↓	
1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

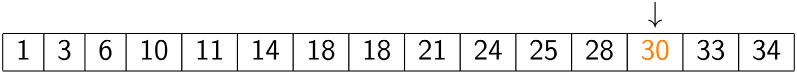
```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```



1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

Binary search

```
final int ELEMENT = 32;
int index = Collections.binarySearch(list, ELEMENT);
```



1	3	6	10	11	14	18	18	21	24	25	28	30	33	34

index gets assigned the value -1.

Temperature app

The temperature app

- randomly selects a city in Ontario,
- reads a corresponding URL, and
- extracts the current temperature.

Temperature app

For each city, we need a corresponding URL. These can be stored in a file.

```
on-122_metric_e.html  Alexandria
on-1_metric_e.html  Algonquin Park (Brent)
on-29_metric_e.html  Algonquin Park (Lake of Two Rivers)
on-114_metric_e.html  Alliston
on-30_metric_e.html  Apsley
on-111_metric_e.html  Armstrong
on-148_metric_e.html  Atikokan
on-164_metric_e.html  Attawapiskat
...
```

Temperature app

Question

What is the most appropriate collection to store the cities and their URLs? A list, a set or a map?

Temperature app

Question

What is the most appropriate collection to store the cities and their URLs? A list, a set or a map?

Answer

A map.

Temperature app

Question

What is the most appropriate collection to store the cities and their URLs? A list, a set or a map?

Answer

A map.

Question

What are the types of the keys and values of the map?

Temperature app

Question

What is the most appropriate collection to store the cities and their URLs? A list, a set or a map?

Answer

A map.

Question

What are the types of the keys and values of the map?

Answer

String and URL.

Object Serialization

Rather than reading a string representation of an object from a file and creating the object, we can also read the object from a file directly.

```
ObjectInputStream objectInput =
   new ObjectInputStream(
      new FileInputStream("cities.dat"));
Object object = objectInput.readObject();
if (object instance of Map)
{
   Map<String, URL> map = (Map) object;
}
objectInput.close();
```

Object Serialization

Rather than writing a string representation of an object to a file, we can also save the object to a file directly.

```
ObjectOutputStream objectOutput =
  new ObjectOutputStream(
     new FileOutputStream("cities.dat"));
objectOutput.writeObject(map);
objectOutput.close();
```

Object Serialization

Question

Which objects can be serialized?

Object Serialization

Question

Which objects can be serialized?

Answer

Those objects that are an instance of a class that implements the interface Serializable .