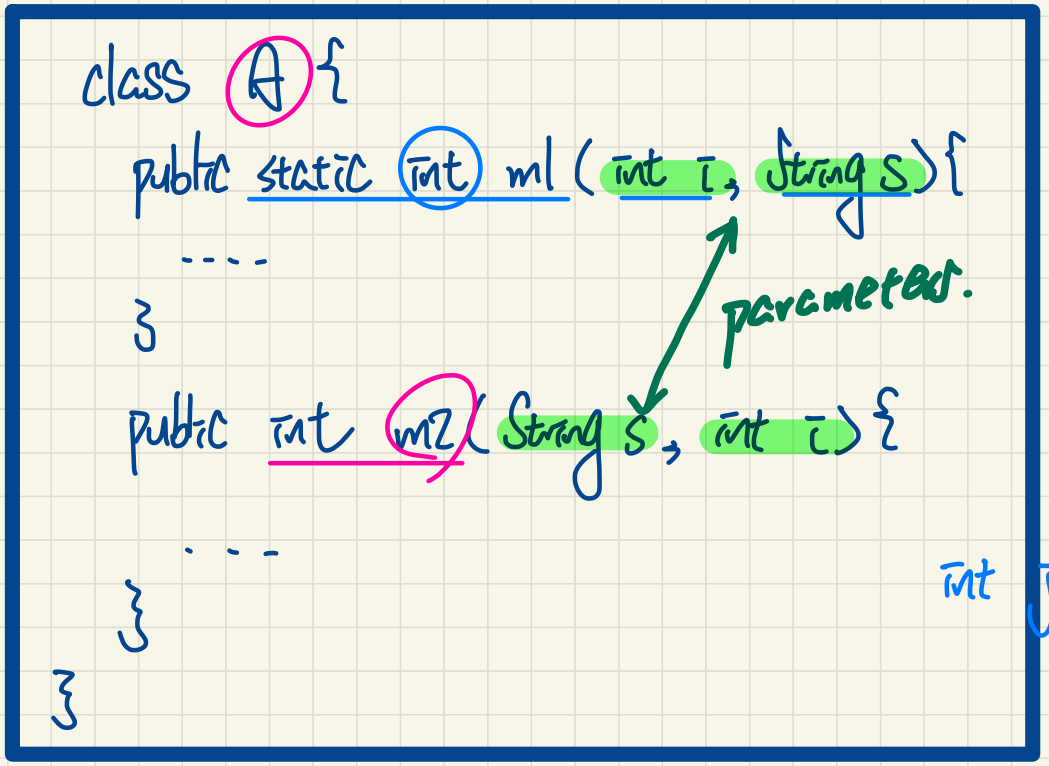
Lecture 6

Part A

API of Java Library -
Method Headers,
Static vs. Non-Static Methods



Tester.

arguments.

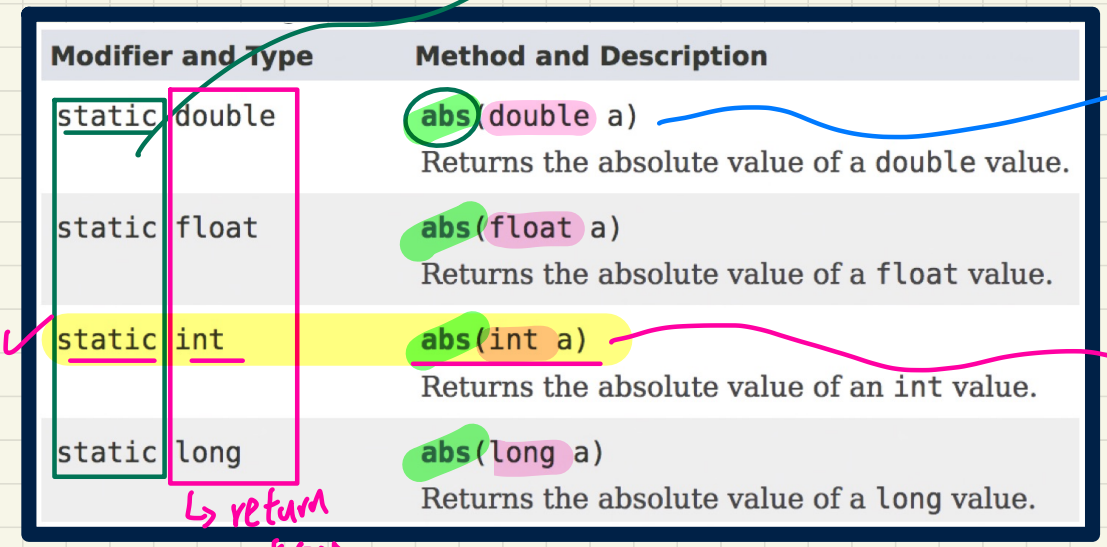
```
int j = A.m1(23, "Alan");
A obj = new A();
int k = obj.m2("Tom", 46);
```

Lecture 6

Part B

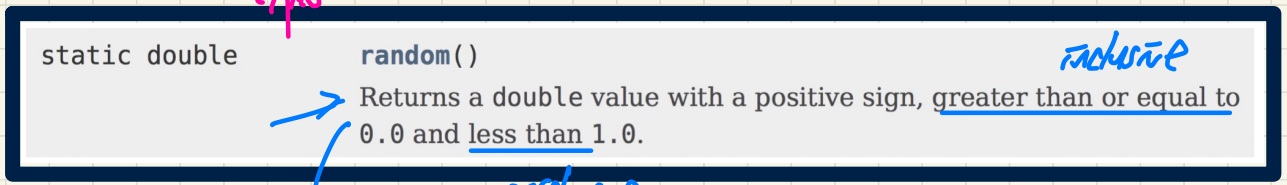
API of Java Library -
Case Study: Math Class

Java API: Math



Modifier and Type	Method and Description
static double	abs(double a) Returns the absolute value of a double value.
static float	abs(float a) Returns the absolute value of a float value.
static int	abs(int a) Returns the absolute value of an int value.
static long	abs(long a) Returns the absolute value of a long value.

modifier
Math.abs(...).
↳ return types
method overloading
- same method names
- different types of parameter types.
method header:
public class Math {
public static int abs(int a) {...}
}



static double	random() Returns a double value with a positive sign, greater than or equal to 0.0 and less than 1.0.

inclusive
exclusive
Math.random() → [0.0, 1.0)
↳ [0.0, 100.0)
eg. 0.01 * 100 / 100 → 0.01234
↳ x100 → 1.234
0.123 * 100 → 12.3 → 12
output

Lecture 6

Part C

API of Java Library -
Case Study: ArrayList Class

API: ArrayList<E>

declaration generic parameter type elements.

ArrayList<Person> list = new ArrayList<Person>();

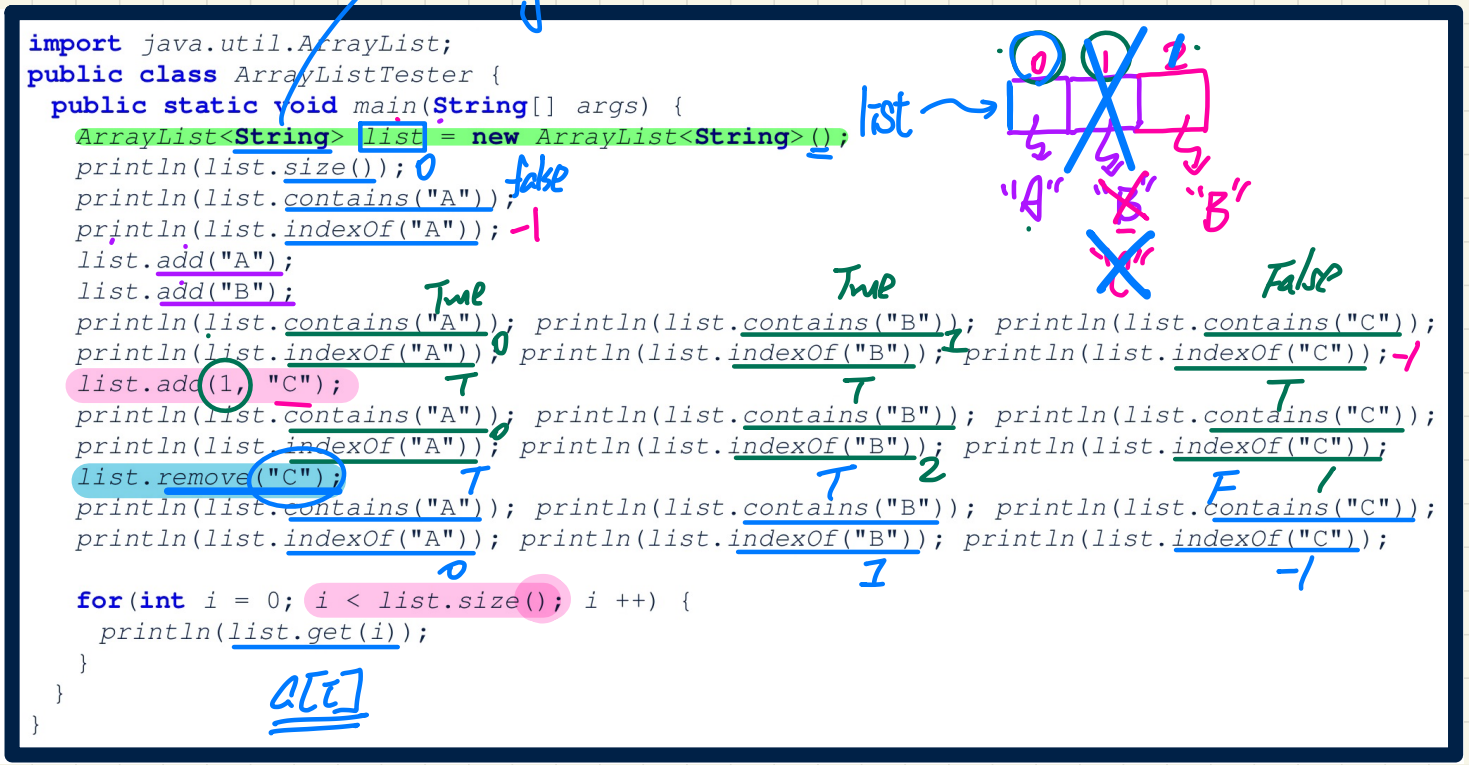
instantiation of E by Person

non-static

int	**size()** Returns the number of elements in this list.
boolean	**add(E e)** Person. Appends the specified element to the end of this list.
void	**add(int index, E element)** Person Inserts the specified element at the specified position in this list.
boolean	**contains(Object o)** Returns true if this list contains the specified element.
E Person	**remove(int index)** 2 Removes the element at the specified position in this list.
boolean	**remove(Object o)** Removes the first occurrence of the specified element from this list, if it is present.
int	**indexOf(Object o)** Returns the index of the first occurrence of the specified element in this list, or -1 if this list does not contain the element.
E Person	**get(int index)** a[index]. Returns the element at the specified position in this list.

overloaded methods.

Use of ArrayList<String>




```
1  import java.util.ArrayList;
2  public class ArrayListTester {
3    public static void main(String[] args) {
4      ArrayList<String> list = new ArrayList<String>();
5      println(list.size());
6      println(list.contains("A"));
7      println(list.indexOf("A"));
8      list.add("A");
9      list.add("B");
10     println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
11     println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));
12     list.add(1, "C");
13     println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
14     println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));
15     list.remove("C");
16     println(list.contains("A")); println(list.contains("B")); println(list.contains("C"));
17     println(list.indexOf("A")); println(list.indexOf("B")); println(list.indexOf("C"));
18
19     for(int i = 0; i < list.size(); i ++) {
20       println(list.get(i));
21     }
22   }
23 }
```

Lecture 6

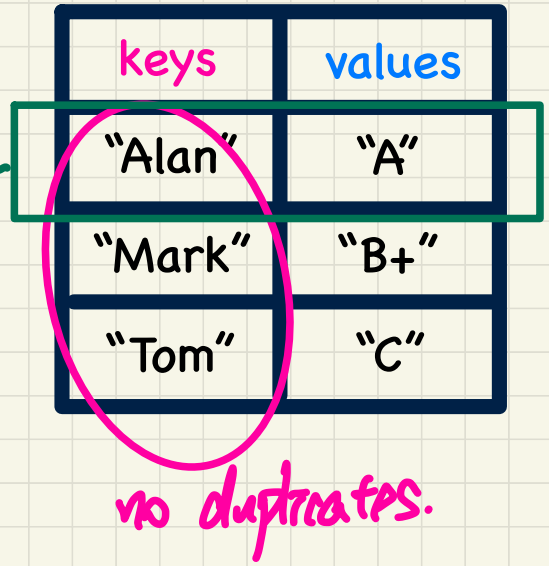
Part D

API of Java Library -
Case Study: Hashtable Class

Hash Table

- 2-column table
- Column of keys contain no duplicates.
- Column of values may contain duplicates.
- Each key uniquely identifies an entry (k, v)
  - key
  - value

grades.



API: HashTable<K, V>

generic parameters.
K → type of keys
V → type of values

Hashtable<String, Person> t = new Hashtable<>();

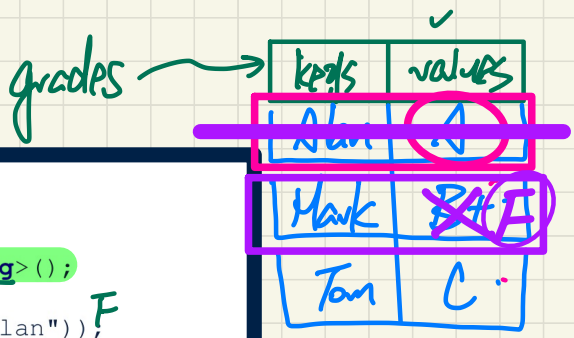
int	**size()** Returns the number of keys in this hashtable.
boolean	**containsKey(Object key)** Tests if the specified object is a key in this hashtable.
boolean	**containsValue(Object value)** Returns true if this hashtable maps one or more keys to this value.
~~V~~ Person	**get(Object key)** Returns the value to which the specified key is mapped, or `null` if this map contains no mapping for the key.
~~V~~	**put(~~K~~ String key, ~~V~~ value)** Maps the specified key to the specified value in this hashtable.
~~V~~	**remove(Object key)** Removes the key (and its corresponding value) from this hashtable.

t.get(_) → Person

t.put("key1", new Person(...));

used to uniquely identify an entry.

Use of HashTable<String, String>




```
1  import java.util.Hashtable;
2  public class HashTableTester {
3    public static void main(String[] args) {
4      Hashtable<String, String> grades = new Hashtable<String, String>();
5      System.out.println("Size of table: " + grades.size());
6      System.out.println("Key Alan exists: " + grades.containsKey("Alan"));
7      System.out.println("Value B+ exists: " + grades.containsValue("B+"));
8      grades.put("Alan", "A");
9      grades.put("Mark", "B+");
10     grades.put("Tom", "C");
11     System.out.println("Size of table: " + grades.size());
12     System.out.println("Key Alan exists: " + grades.containsKey("Alan"));
13     System.out.println("Key Mark exists: " + grades.containsKey("Mark"));
14     System.out.println("Key Tom exists: " + grades.containsKey("Tom"));
15     System.out.println("Key Simon exists: " + grades.containsKey("Simon"));
16     System.out.println("Value A exists: " + grades.containsValue("A"));
17     System.out.println("Value B+ exists: " + grades.containsValue("B+"));
18     System.out.println("Value C exists: " + grades.containsValue("C"));
19     System.out.println("Value A+ exists: " + grades.containsValue("A+"));
20     System.out.println("Value of existing key Alan: " + grades.get("Alan"));
21     System.out.println("Value of existing key Mark: " + grades.get("Mark"));
22     System.out.println("Value of existing key Tom: " + grades.get("Tom"));
23     System.out.println("Value of non-existing key Simon: " + grades.get("Simon"));
24     grades.put("Mark", "F");
25     System.out.println("Value of existing key Mark: " + grades.get("Mark"));
26     grades.remove("Alan");
27     System.out.println("Key Alan exists: " + grades.containsKey("Alan"));
28     System.out.println("Value of non-existing key Alan: " + grades.get("Alan"));
```