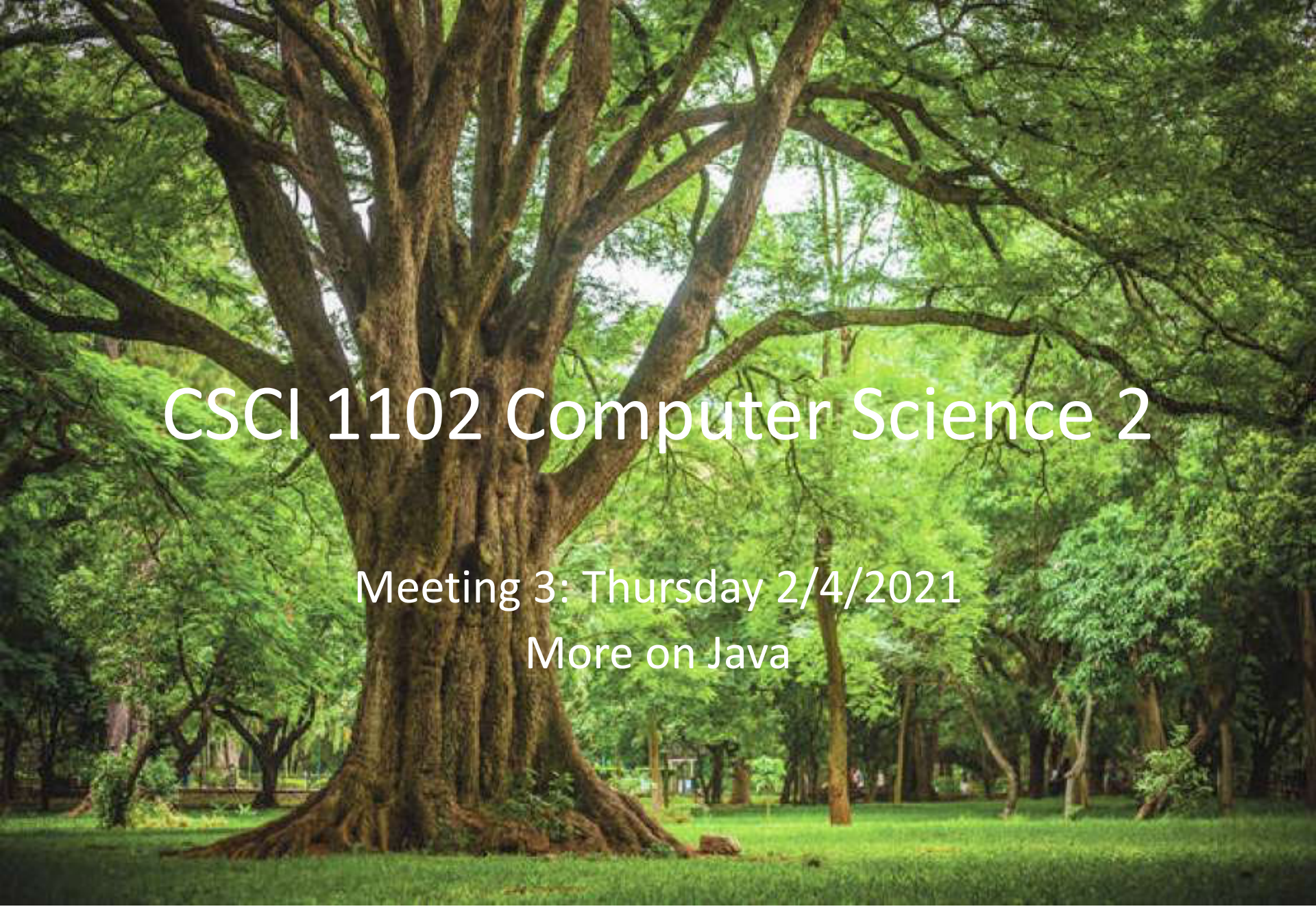


CSCI 1102 Computer Science 2

Meeting 3: Thursday 2/4/2021

More on Java

Python & Dynamic Typing

```
def f(a):
  return (a[0], a + 6)
```

Me, looking at a list of advantages of dynamic typing.

```java
static int fib(int n) {
  if (n < 2)
    return 1;
  else
    return fib(n - 1) + fib(n - 2);
}
```

Two wildly inefficient Fibonacci functions.

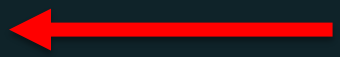
```python
def fib(n):
    if (n < 2):
        return 1
    else:
        return fib(n - 1) + fib(n - 2)
```

Run-times on VMs for fib(38)

```
> javac Fib.java
> time java Fib 38
fib(38) = 63245986

real    0m0.327s
user    0m0.322s
sys     0m0.035s
> time python fib.py 38
63245986

real    0m15.247s
user    0m14.767s
sys     0m0.125s
>
```



The Python time includes time for translation.

But Java is Huge & Wildly Complicated

Today

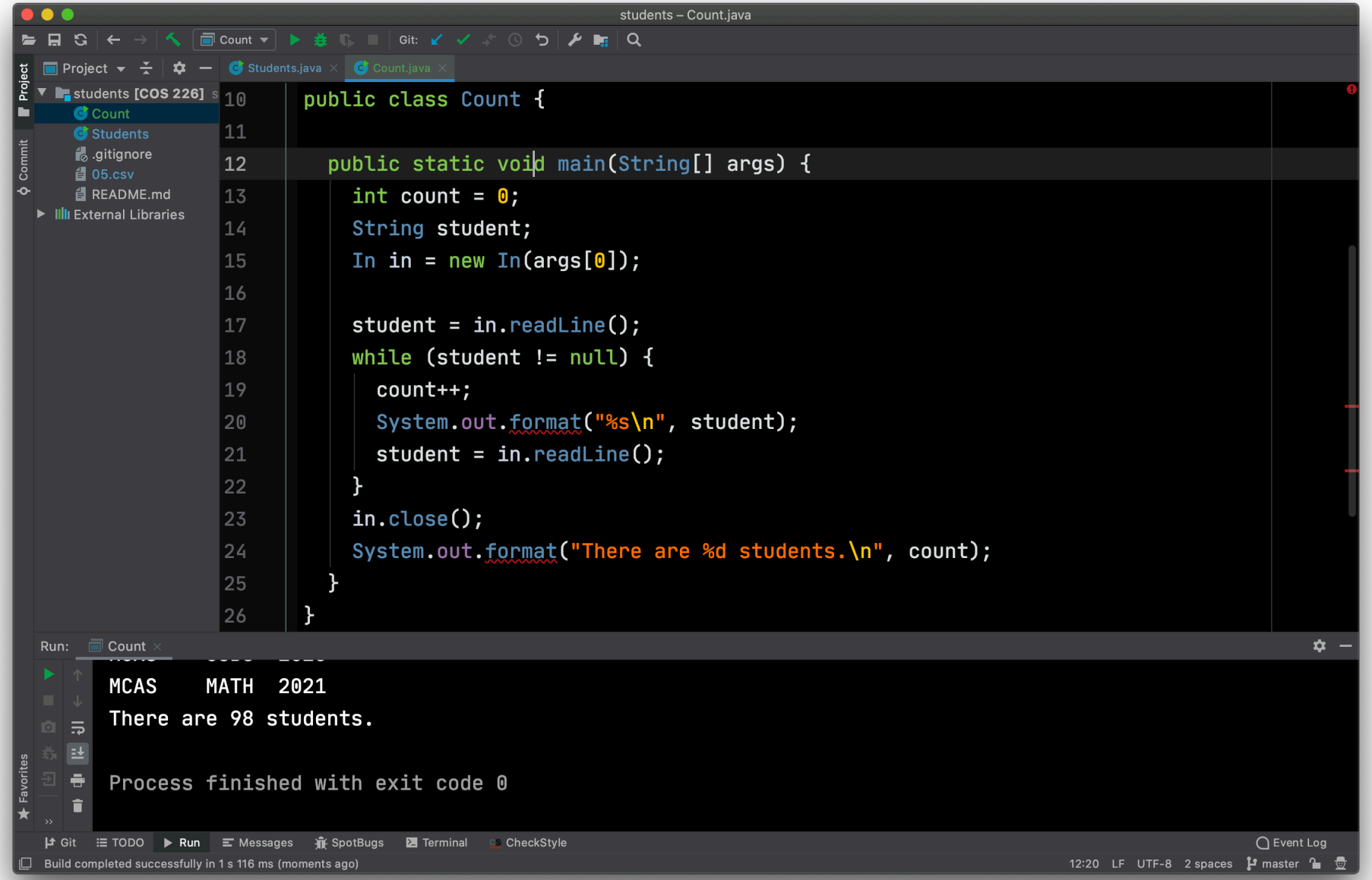
- Message-Passing Style
- Frequency Table Example
- Abstract Data Types

Example: Who is taking CS2?

s	School	Major	Grad Year
nas@bc	Morrissey College, A	computer science b.	2023
i@bc.e	Morrissey College, A	computer science b.	2024
ne@bc	Morrissey College, A	computer science b.	2024
c.edu	Morrissey College, A	computer science b.	2022
@bc.e	Morrissey College, A	philosophy	2022
@bc.e	Morrissey College, A	computer science b.	2024
y@bc.e	Morrissey College, A	computer science b.	2023
dia@bc	Morrissey College, A	mathematics bs	2022
an@bc	Carroll School of Ma	information systems	2023
is.5@b	Morrissey College, A	biology	2021
c.edu	Carroll School of Ma	undeclared	2023
@bc.e	Morrissey College, A	computer science b.	2024
bc.edu	Morrissey College, A	computer science b.	2024
za@bc.	Morrissey College, A	computer science b.	2024
.edu	Morrissey College, A	computer science b.	2024
.2@bc	Morrissey College, A	computer science b.	2024
@bc.ec	Morrissey College, A	computer science b.	2024
es@bc.	Morrissey College, A	computer science b.	2024

MCAS	CSBA	2024
MCAS	PHIL	2022
MCAS	CSBA	2024
MCAS	CSBA	2022
MCAS	CSBS	2024
MCAS	CSBA	2023
MCAS	MATH	2022
CSOM	ISYS	2023
MCAS	BIOL	2021
CSOM	UNDC	2023
MCAS	CSBA	2024
MCAS	CSBA	2024
MCAS	CSBS	2024
MCAS	CSBS	2024
MCAS	CSBS	2024
MCAS	CSBA	2024
MCAS	MATH	2024
MCAS	CSBS	2024
MCAG	BIOL	2026
MCAS	ECON	2023

Count Enrolled Students



Message Passing Style

- In the Count example, in is an *input channel* and readLine is a function.

- In standard mathematical style, we read a line from the input channel using standard function call notation: readLine(in)

- static functions are called using the standard notation.

Message Passing Style

- Non-static (or “dynamic”) functions are called in message-passing style: in.readLine()

```
11
12    public static void main(String[] args) {
13      int count = 0;
14      String student;
15      In in = new In(args[0]);
16
17      student = in.readLine();
18      while (student != null) {
19        count++;
20        System.out.format("%s\n", student);
21        student = in.readLine();
22      }
23      in.close();
24
```

Message Passing Style

- The input channel in is understood to be a structure (*aka*, an "object") containing functions

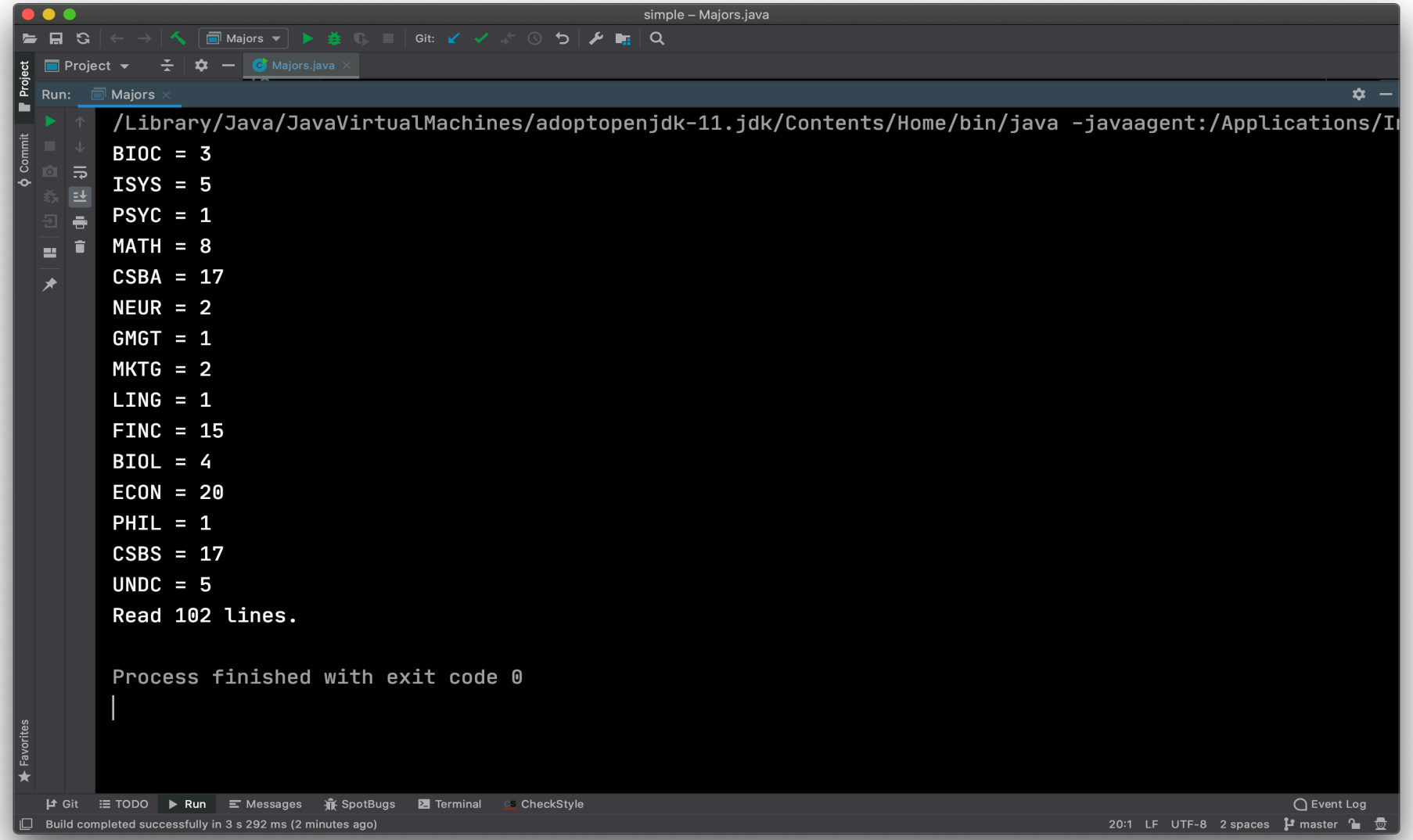
```
11
12    public static void main(String[] args) {
13      int count = 0;
14      String student;
15      In in = new In(args[0]);
16
17      student = in.readLine();
18      while (student != null) {
19        count++;
20        System.out.format("%s\n", student);
21        student = in.readLine();
22      }
23      in.close();
24
```

Message Passing Style

- in.readLine is sometimes interpreted as "*send the readLine message to channel in*"

```
11
12    public static void main(String[] args) {
13      int count = 0;
14      String student;
15      In in = new In(args[0]);
16
17      student = in.readLine();
18      while (student != null) {
19        count++;
20        System.out.format("%s\n", student);
21        student = in.readLine();
22      }
23      in.close();
24
```

Compute a Frequency Table of Majors



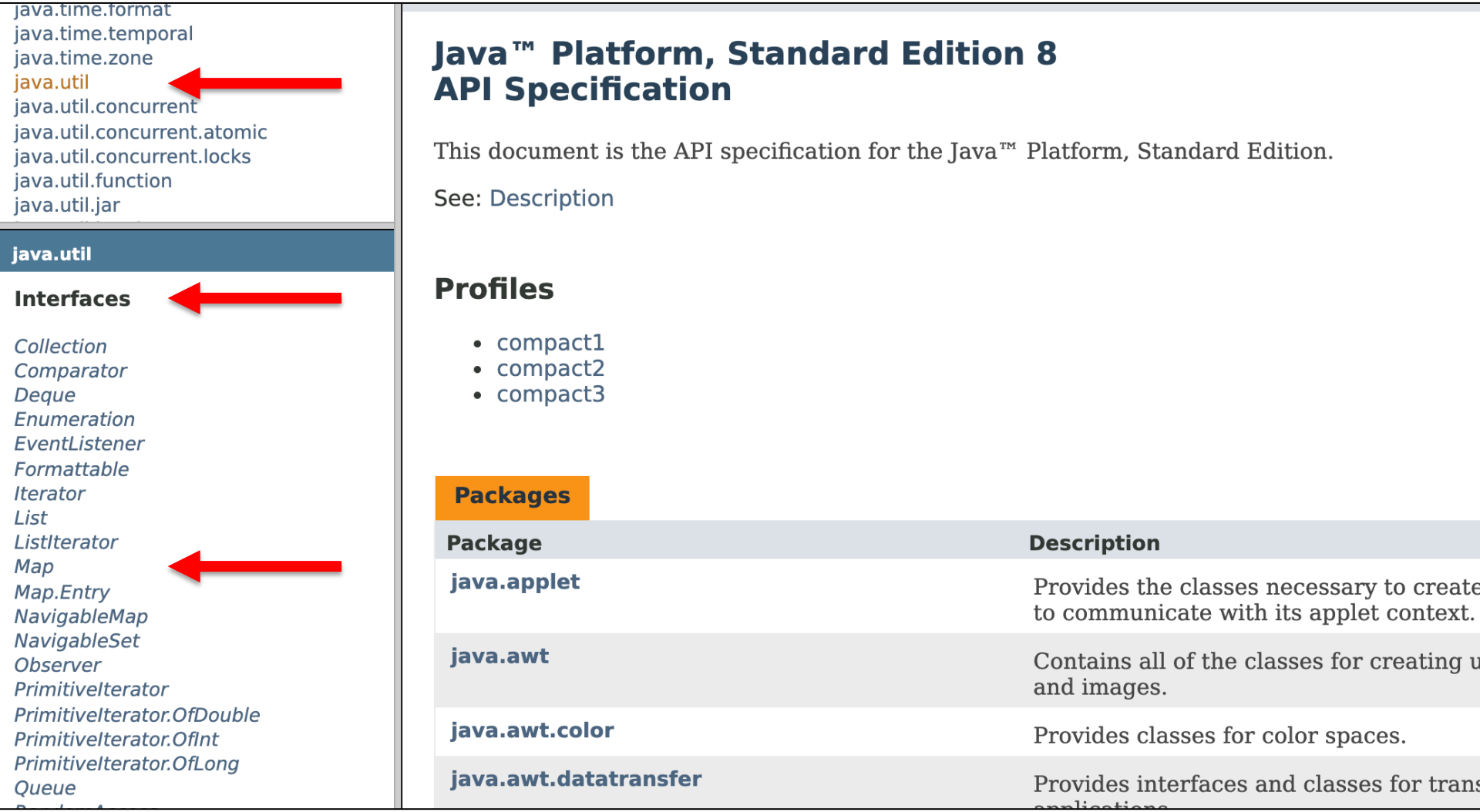
Map/Dictionary -- Python

```
2     # PYTHON
3
4     frequency = { "BIOC" : 3, "ISYS" : 5}  # etcetera
5
6     biochemistry = frequency["BIOC"]
7
```

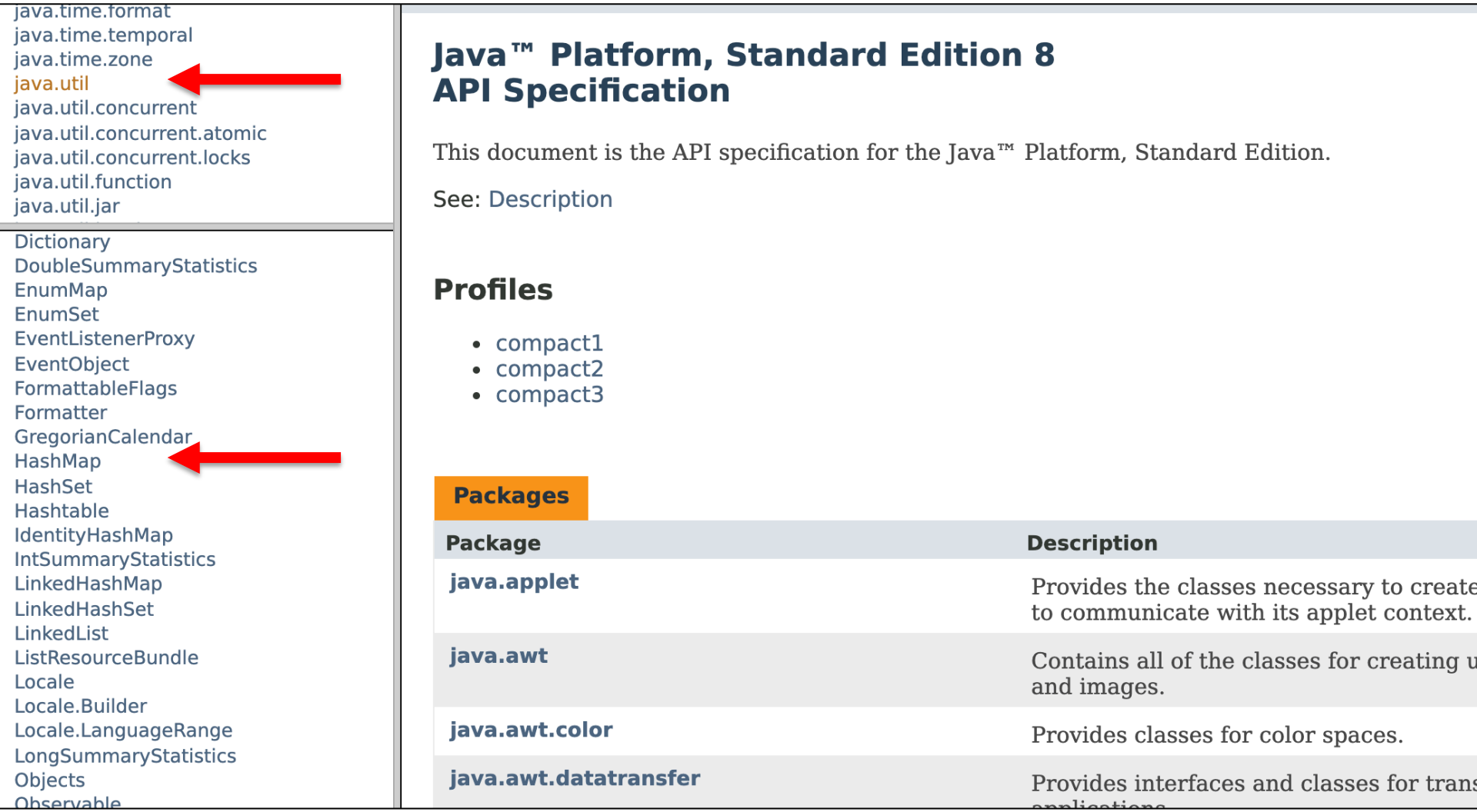
Map/Dictionary -- OCaml

```
2    (* OCaml *)
3
4    module Map = Map.Make(String)
5
6    let frequency = Map.add "BIOC" 3 (Map.add "ISYS" 5 Map.empty)
7
8    let biochemistry = Map.find "BIOC" frequency
9
```

Map/Dictionary -- Java



Map/Dictionary -- Java



Map/Dictionary -- Java

```
6   // Java
7
8   Map<String, Integer> frequency = new HashMap();
9
10  frequency.put("BIOC", 3);
11  frequency.put("ISYS", 5);
12
13  Integer biochemistry = frequency.get("BIOC");
14
```

Map/Dictionary -- Java

```
6   // Java
7
8   Map<String, Integer> frequency = new HashMap();
9
10  frequency.put("BIOC", 3);
11  frequency.put("ISYS", 5);
12
13  Integer biochemistry = frequency.get("BIOC");
14
```

An interface

A class

NB: This Works but Avoid if Possible

```
6     // Java
7
8     HashMap<String, Integer> frequency = new HashMap();
9
10    frequency.put("BIOC", 3);
11    frequency.put("ISYS", 5);
12
13    Integer biochemistry = frequency.get("BIOC");
14
```

A class

A class

Map/Dictionary -- Java

```
6   // Java
7
8   Map<String, Integer> frequency = new HashMap();
9
10  frequency.put("BIOC", 3);
11  frequency.put("ISYS", 5);
12
13  Integer biochemistry = frequency.get("BIOC");
14
```

Key

Value

Digression 1: – Base Types & Wrapper Classes

```
6   // Java
7
8   Map<String, Integer> frequency = new HashMap();
9
10  frequency.put("BIOC", 3);
11  frequency.put("ISYS", 5);
12
13  Integer biochemistry = frequency.get("BIOC");
14
```

A *reference type*

Values of *base type* int

Base Types & Wrapper Classes

Base Type	Wrapper Class/Type
int (32-bit integers)	Integer
long (64-bit integers)	Long
short (16-bit integers)	Short
byte (8-bit integers)	Byte
float (32-bit floats)	Float
double (64-bit floats)	Double
char	Character
boolean	Boolean

java.io
java.lang
java.lang.annotation
java.lang.instrument
java.lang.invoke
java.lang.management
java.lang.ref
java.lang.reflect
java.math
java.net
java.nio

Classes

Boolean
Byte
Character
Character.Subset
Character.UnicodeBlock
Class
ClassLoader
ClassValue
Compiler
Double
Enum
Float
InheritableThreadLocal
Integer
Long
Math
Number
Object

	Returns the hash code value for t
boolean	**isEmpty**() Returns true if this map contains
Set<**K**>	**keySet**() Returns a **Set** view of the keys co
default **V**	**merge**(**K** key, **V** value, **BiFunct** **V**> remappingFunction) If the specified key is not already it with the given non-null value.
V	**put**(**K** key, **V** value) Associates the specified value wit
void	**putAll**(**Map**<? extends **K**,? exte Copies all of the mappings from th
default **V**	**putIfAbsent**(**K** key, **V** value) If the specified key is not already with the given value and returns
V	**remove**(**Object** key) Removes the mapping for a key fr
default boolean	**remove**(**Object** key, **Object** va Removes the entry for the specifie

Base Types & Wrapper Classes

```
16
17      Integer wrapped = new Integer(343);
18      String succeed = wrapped.toString();  // Integer has a toString fu
19
20      String fail = 343.toString();
21
22      Integer wrapped = 343;                // Autoboxing
23      String succeed = wrapped.toString();
24
25      int succeed = 12 * wrapped;           // Auto-unboxing
26
```

Digression 2: Exceptions

- Each line of the input file has two strings and one integer;
- The algs4 In class has readString and readInt functions;
- What happens if there is no string or int to read?

readString

```
public String readString()
```

Reads the next token from this input stream and returns it as a `String`.

Returns:

`the next String in this input stream`

Throws:

`NoSuchElementException` - if the input stream is empty

If an exception is thrown we need to catch it.

readInt

```
public int readInt()
```

Reads the next token from this input stream, parses it as a `int`, and returns the `int`.

Returns:

`the next int in this input stream`

Throws:

`NoSuchElementException` - if the input stream is empty

`InputMismatchException` - if the next token cannot be parsed as an int

java.time.format
java.time.temporal
java.time.zone
java.util
java.util.concurrent
java.util.concurrent.atomic
java.util.concurrent.locks
java.util.function
java.util.jar
java.util.logging

Formatter.BigDecimalLayoutForm
Locale.Category
Locale.FilteringMode

Exceptions

ConcurrentModificationException
DuplicateFormatFlagsException
EmptyStackException
FormatFlagsConversionMismatchExcepti
FormatterClosedException
IllegalFormatCodePointException
IllegalFormatConversionException
IllegalFormatException
IllegalFormatFlagsException
IllegalFormatPrecisionException
IllegalFormatWidthException
IllformedLocaleException
InputMismatchException
InvalidPropertiesFormatException
MissingFormatArgumentException
MissingFormatWidthException
MissingResourceException
NoSuchElementException
TooManyListenersException
UnknownFormatConversionException
UnknownFormatFlagsException

default V	**putIfAbsent**(K key, V value) If the specified key is not already associat with the given value and returns null, els
V	**remove**(Object key) Removes the mapping for a key from this
default boolean	**remove**(Object key, Object value) Removes the entry for the specified key or
default V	**replace**(K key, V value) Replaces the entry for the specified key or
default boolean	**replace**(K key, V oldValue, V newValu Replaces the entry for the specified key or
default void	**replaceAll**(BiFunction<? super K,? su Replaces each entry's value with the resul entries have been processed or the functio
int	**size**() Returns the number of key-value mapping
Collection<V>	**values**() Returns a **Collection** view of the values c

Method Detail

size

Handling Exceptions

```
27
28      try {
29          ... code that may throw an exception ...
30      }
31      catch (Exception e) {
32          ... recovery code ...
33      }
34
```

Code

Compute a Frequency Table of Majors