CSCI 134 Fall 2021:
Introduction to Java III
Dec 6, 2021
Shikha Singh, 9AM
Jeannie Albrecht, 10AM

Announcements & Logistics

• **HW 9** due tonight @ 10pm
  • Covers a lot of “advanced” topics from recent lectures
  • Review special methods, iterators, generators, efficiency
  • Office hours today (Shikha): 3-5 pm if you have questions
  • Office hours this week: Check calendar

• **Lab 10 Selection Sort in Java**: due Wed/Thurs @ 10 pm
  • Hope most of you will start and finish during your lab session

• **Final exam reminder**: Dec 18 @ 9:30 am
  • Practice problems: we will release later this week
  • Review session during reading period: details TBD

• **Course evals on Friday**: bring a laptop to class if possible
  • Short wrap up (~30 min), end early for you to fill out evals in class
Last Time

• Discussed an example of how to read input in Java, do basic arithmetic and print the output

• Introduced **data types** in Java:
  
  • **Strings**
  
  • **ArrayLists** and **Arrays** (like Python lists)
  
  • **HashMaps** (like Python dictionaries)
Today

- Discuss **loops** and **conditionals** in Java
  - If else statements, for loops, while loops
  - Review Python syntax as well!
- Begin discussing **methods** and **return** types in Java
Programming Language Features

- **Basic features:**
  - Data Types
  - Reading user input
  - Loops
  - **Conditionals**

- **Advanced topics:**
  - Classes
  - Interfaces
  - Collections
  - Graphical User Interface Programming
Booleans

- **Boolean** (or **boolean**) values in Java:
  - **true** and **false** (no capitalization)
  - Example: **Boolean b = true**

- **Boolean** operators in Java:
  - **&&** - and
  - **||** - or
  - **!** - not
  - Most other operators (**<**, **>**, **==**, etc) are the same as Python
Conditional Statements

- **Conditional** (if-else) statements in Python and Java are very similar.
- Let's consider three basic patterns:
  1. Simple if in Python:
     ```python
     if condition:
         statement1
         statement2
         ...
     ```
  2. Simple if in Java:
     ```java
     if (condition) {
         statement1;
         statement2;
         ...
     }
     ```

Note the use of `( )` and `{ }`
Conditional Statements

- **Conditional** (if-else) statements in Python and Java are very similar.

- Let’s consider three basic patterns:

  2. if else in Python:

```python
if condition:
    statement1
    statement2
    ...
else:
    statement1
    statement2
    ...
```

  if else in Java:

```java
if (condition) {
    statement1;
    statement2;
    ...
} else {
    statement1;
    statement2;
    ...
}
```
Conditional Statements

- **Conditional** (if-else) statements in Python and Java are very similar.
- Let's consider three basic patterns.

3. If elif else in Python:

```python
if condition:
    statement1
    statement2
...
elif condition:
    statement1
    statement2
...
else:
    statement1
    statement2
...
```

Nested if else if in Java:

```java
if (condition) {
    statement1;
    statement2;
    ...
} else if (condition) {
    statement1;
    statement2;
    ...
} else {
    statement1;
    statement2;
    ...
}
```

Java does not have an elif equivalent.
Conditional Statements

Python:

```python
a = 1
b = 2
if a < b:
    print("a < b")
a < b

if a > b:
    print("a > b")
else:
    print("a < b")
a < b

c = 3
if a > b and a > c:
    print("a is largest")
elif b > a and b > c:
    print("b is largest")
else:
    print("c is largest")
c is largest
```

Java:

```java
int a = 1;
int b = 2;
if (a < b) {
    System.out.println("a < b");
}
a < b

if (a > b) {
    System.out.println("a > b");
} else {
    System.out.println("a < b");
}
a < b

int c = 3;
if (a > b && a > c) {
    System.out.println("a is largest");
} else if (b > a && b > c) {
    System.out.println("b is largest");
} else {
    System.out.println("c is largest");
}
c is largest
```

Notice the && (logical AND) operator.
Recall one of the first examples we looked at involving conditionals in Python (slightly modified to accept user input)

```python
1  def weather(temp):
2      if temp > 80:
3          print("It is a hot one out there.")
4      elif temp >= 60:
5          print("Nice day out, enjoy!")
6      elif temp >= 40:
7          print("Chilly day, wear a sweater.")
8      else:
9          print("Its freezing out, bring a winter jacket!")
10     
11     if __name__ == "__main__":
12         temp = int(input("Enter temp: "))
13         weather(temp)
```
Lecture 5 Revisited

• Let’s write it in Java!
Lecture 5 Revisited

```java
import java.util.Scanner;

public class WeatherFinal {
    public static void main (String args[]) {
        int temp;
        Scanner in;

        in = new Scanner(System.in);
        System.out.print("Enter temp: ");
        temp = in.nextInt();

        if (temp > 80) {
            System.out.println("It is a hot one out there. ");
        } else if (temp >= 60) {
            System.out.println("Nice day out, enjoy!");
        } else if (temp >= 40) {
            System.out.println("Chilly day, wear a sweater.");
        } else {
            System.out.println("Its freezing out, bring a winter jacket!");
        }
    }
}
```

Could use Integer here as well or Double for floating pt values.
Programming Language Features

• **Basic features:**
  • Data Types
  • Reading user input
  • **Loops**
  • Conditionals

• **Advanced topics:**
  • Classes
  • Interfaces
  • Collections
  • Graphical User Interface Programming
Loops

• We studied two different kinds of loops this semester
  • *Indefinite* loops (runs indefinitely until condition turns false)
    • **While loops**
      
      ```
      while condition:
          # do something
      ```
    • *Definite* loops (runs a specific number of times)
      • **For loops**
        
        ```
        for el in seq:
            # do something
        ```
  • We'll look at both of these in Java
While Loops

- While loops in both languages are exactly the same (except for `()` `{}`)

Python:

```python
num = 10
while num > 0:
    print(num)
    num = num // 2
```

Java:

```java
int num = 10;
while (num > 0){
    System.out.println(num);
    num = num / 2;
}
```

When dividing Integers, Java automatically performs integer division. (No `//` in Java)
For Loops and Range Review

- Recall Python’s `range` type: `range(start,stop,step)`
  - Example: `range(100,-1,-5)`
    - *Start* at 100, *stop* at -1, count *backward by 5*
    - Often use `range` object as part of for loop
  - Java does not have a `range` data type
  - Java’s for loop syntax captures start and stop conditions explicitly

```java
for (start clause; stop clause; step clause) {
    statement1;
    statement2;
    ...
}
```

- Let’s look at a few examples
For Loops

- **Python** for loops allow you to iterate directly over any *iterable*
- **Java** syntax is a bit different and there is no range equivalent

for loops in Python:

```python
for i in range(10):
    print(i)
...
for el in seq:
    print(el)
...
```

for loops in Java:

```java
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
...
for (int i : myArray) {
    System.out.println(i);
}
```
For Loops

- **Python** for loops allow you to iterate directly over any **iterable**
- **Java** syntax is a bit different and there is no range equivalent

for loops in Python:

```python
for i in range(10):
    print(i)
...
```

```python
for el in seq:
    print(el)
...
```

for loops in Java:

```java
for (int i = 0; i < 10; i++) {
    System.out.println(i);
    ...
}
```

```java
for (int i : myArray) {
    System.out.println(i);
    ...
}
```

Start at i=0
Continue while i < 10
Same as i += 1 or i = i + 1
For Loops

- **Python** for loops allow you to iterate directly over any *iterable*
- **Java** syntax is a bit different and there is no range equivalent

for loops in Python:

```python
for i in range(10):
    print(i)
...
for el in seq:
    print(el)
...
```

for loops in Java:

```java
for (int i = 0; i < 10; i++) {
    System.out.println(i);
    ...
}
for (int i : myArray) {
    System.out.println(i);
    ...
}
```
For Loops

- **Python** for loops allow you to iterate directly over any *iterable*
- **Java** syntax is a bit different and there is no range equivalent

**for loops in Python:**

```python
for i in range(10):
    print(i)
    ...

for el in seq:
    print(el)
    ...
```

**for loops in Java:**

```java
for (int i = 0; i < 10; i++) {
    System.out.println(i);
    ...
}

for (int i : myArray) {
    System.out.println(i);
    ...
}
```

Called a **for each** loop in Java

For each i in myArray (of ints)
Lecture 6 Revisited: First for loop

- **Python** for loops also allow you to iterate directly over an **iterable**
  - Without using indices or knowing the length of the sequence
  - Recall this simple example from Lecture 6
  - Now we also know what happens behind the scenes (uses `iter`)

```python
word = "Williams"
for char in word:
    print(char)
```

```python
try:
    it = iter(word)
    while True:
        char = next(it)
        print(char)
except StopIteration:
    pass
```
Lecture 6 Revisited: First for loop

• Java **for each** loops internally use iterators just like Python and are equivalent to Python **for** loops (aside from data type complications)

• **for each** loops can easily iterate over arrays and Collections in Java

Python:

```python
word = "Williams"
for char in word:
    print(char)
```

Java (for each):

```java
String word = "Williams";
for (char c: word.toCharArray()) {
    System.out.println(c);
}
```
Lecture 6 Revisited: First for loop

- Java **for loops** explicitly use indices and specify the stopping condition (length of sequence) ahead of time.

- In Python, we can rewrite our **for** loop as shown below to use indices, length, and range.

- After rewriting, it will be easier to convert to Java.

```
word = "Williams"
for char in word:
    print(char)

# In Python

word = "Williams"
size = len(word)
for i in range(size):
    print(word[i])
```


Lecture 6 Revisited: First for loop

- Java **for loops** explicitly use indices and specify the stopping condition (length of sequence) ahead of time.

- No range equivalent in Java.

### Python:

```python
word = "Williams"
size = len(word)
for i in range(size):
    print(word[i])
```

### Java (for):

```java
String word = "Williams";
int len = word.length();
for (int i = 0; i < len; i++) {
    System.out.println(word.charAt(i));
}
```
countVowels

- Recall the `countVowels` function from Lecture 6 that combined for loops and conditionals

- Notice that our docstring specifies input & output type of function, but this is just convention in Python (not required)

```python
def countVowels(word):
    """ Takes a str word and returns a the number of vowels in it (int) """
    count = 0
    for char in word:
        if char.lower() in "aieou":
            count += 1
    return count
```
countVowels in Java

• Writing the same method in Java

```java
public static int countVowels(String word) {
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)) {
            count++;
        }
    }

    return count;
}
```
Writing the same method in Java

```java
public static int countVowels(String word){
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)){
            count++;
        }
    }
    return count;
}
```
countVowels in Java

• Writing the same method in Java

```java
import java.lang.String;

public static int countVowels(String word) {
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)) {
            count++;
        }
    }

    return count;
}
```
countVowels in Java

• Writing the same method in Java

```java
public static int countVowels(String word){
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)){
            count++;
        }
    }
    return count;
}
```

Define vowel string & compute length of word
countVowels in Java

- Writing the same method in Java

```java
public static int countVowels(String word) {
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)) {
            count++;
        }
    }
    return count;
}
```

- `charAt` returns a `char` (primitive type), no equivalent in Python
- `String.valueOf(letter)` is like `str(letter)` in Python and converts `letter` to a `String`
countVowels in Java

• Writing the same method in Java

```java
public static int countVowels(String word) {
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)) {
            count++;
        }
    }

    return count;
}
```

Similar to `s in vowels` in Python
countVowels in Java

• Writing the same method in Java

```java
public static int countVowels(String word){
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)) {
            count++;
        }
    }

    return count;
}
```

Can also say count += 1
countVowels in Java

- Writing the same method in Java

```java
public static int countVowels(String word){
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();
    for (int i = 0; i < len; i++) {
        char letter = word.charAt(i);
        String s = String.valueOf(letter);
        if (vowels.contains(s)){
            count++;
        }
    }
    return count;
}
```
countVowels2 in Java

- Writing the same method in Java using a for each loop

```java
public static int countVowels2(String word){
    int count = 0;
    String vowels = "aeiou";
    int len = word.length();

    for (char letter : word.toCharArray()) {
        String s = String.valueOf(letter);
        if (vowels.contains(s)){
            count++;
        }
    }
    return count;
}
```
public class Vowels {

    public static int countVowels(String word){
        int count = 0;
        String vowels = "aeiou";
        int len = word.length();

        for (int i = 0; i < len; i++) {
            char letter = word.charAt(i);
            String s = String.valueOf(letter).toLowerCase();
            if (vowels.contains(s)){
                count++;
            }
        }
        return count;
    }

    public static void main (String args[]) {
        String word = "Williams";
        System.out.println(countVowels(word));
    }
}