CS 2: Introduction to Programming Methods

Python-Java Comparison

Introduction to Java

You may have heard that "Java is verbose". This is true, but *there's a reason for much of the verbosity*. Every word of a Java program means something, and it's important to *eventually* understand all of it. We will demystify the magic incantations later, but, for now, we will just write them.

How To Use This Guide

- Read this page.
- Read the language constructs section.
- Try to write Java code, and when you get stuck, search this PDF for examples or keywords

Types

In Java, there are two kinds of types: primitive types and Object types.

The primitive types that we care about are `boolean`, `int`, `char`, and `double`. All of these have an object type version (`Boolean`, `Integer`, `Character`, and `Double`, respectively). And pretty much all the other types you will ever see will be object types. If the type starts with a capital letter, it is likely an Object type. The distinction doesn't usually matter except for two circumstances:

- If you are comparing two values for equality, you need to use == for primitive types and `.equals()` for object types.
- If you are using "generics" to declare a data structure (e.g., a `List` or a `Map`), you *must* use the object type version.

Some common object types you will see are:

- `Scanner`: This is used for reading from the console and files.
- `Random`: This is used to generate sequences of random numbers.
- `List`: This is the Java equivalent of Python's `list` type.
- `Map`: This is the Java equivalent of Python's `dict` type.

Language Constructs

Most of the language constructs (e.g., loops, if statements) you are familiar with from Python have Java equivalents. This section will show a direct comparison of the syntax.

Variable Declaration

Python	Java	Commentary
`i = 0`	`int i = 0;`	*The Java version has a type declaration as well as a semi-colon.*

Comments

Python	Java	Commentary
`# This is a single line comment`	`// This is a single line comment`	*Java also has multi-line comments which are started by "/*" and ended by "*/".*

While Loop

Python	Java	Commentary

`i = 0`
`while i < 10:`
`    print(i)`
 	
`    i += 1`
	`int i = 0;`
	`while (i < 10) {`
	`    System.out.println(i);`
	
	`    i++;`
	
	`}`
		Java has a "++" operator which we often use in place of "+= 1".

For Loop (counts up)

Python	Java	Commentary

`for i in range(10):`
`    print(i)`
	`for (int i = 0; i < 10; i++) {`
	`    System.out.println(i);`
	`}`
		Both of these loops go "until" $i = 10$.

For Loop (counts down)

Python	Java	Commentary

`for i in reversed(range(10)):`
`    print(i)`
	`for (int i = 9; i >= 0; i--) {`
	`    System.out.println(i);`
	`}`
		Both of these loops go "until" $i < 0$.

Foreach Loop

Python	Java	Commentary

`L = []`
`L.append(1)`
`L.append(2)`
`for i in L:`
`    print(i)`
	`List<Integer> list = new ArrayList<Integer>();`
	`list.add(1);`
	`list.add(2);`
	`for (Integer i : list) {`
	`    System.out.println(i);`
	`}`
		Notice the type declaration as well as the : instead of the “in”.

If-ElseIf-Else Statement

Python	Java	Commentary

`if i % 3 == 0 and i % 5 == 0:`
`    print("FizzBuzz")`
`elif i % 3 == 0:`
`    print("Fizz")`
`elif i % 5 == 0:`
`    print("Buzz")`
`else:`
`    print(i)`
	`if (i % 3 == 0 && i % 5 == 0) {`
	`    System.out.println("FizzBuzz");`
	`}`
	`else if (i % 3 == 0) {`
	`    System.out.println("Fizz");`
	`}`
	`else if (i % 5 == 0) {`
	`    System.out.println("Buzz");`
	`}`
	`else {`
	`    System.out.println(i);`
	`}`
		Similarly, in Java, or is “`||`” and not is “`!`”
		Note the difference between `elif` and `else if`

Function Declaration and Usage

Python	Java	Commentary

`def greeting(name):`
`    return "Hello, " + name`
`print(greeting("Adam"))`
	`public static String greeting(String name) {`
	`    return "Hello, " + name;`
	`}`
	`System.out.println(greeting("Adam"));`
		`static` roughly means “this declaration is a function, not a method”
		When calling a function, *Python syntax is basically the same as Java syntax.*

Data Structures

Java and Python have similar built-in data structures, but the syntax to use them is slightly different. Additionally, Java has the extra concept of a "fixed-size list" called an array.

Strings

Python	Java	Commentary
`s = "hello"`	`String s = "hello";`	
`s += " world"`	`s += " world";`	
`print(len(s))`	`System.out.println(s.length());`	
`print(s[1:5])`	`System.out.println(s.substring(1, 5));`	
`print(s[2])`	`System.out.println(s.charAt(2));`	
`if "hello" in s: ...`	`if (s.indexOf("hello") !=-1) { ... }`	

Arrays (fixed-size lists)

Python	Java	Commentary
`zeroedL = [0, 0, 0]`	`int[] zeroedArray = new int[3];`	
`L = [1, 2, 3]`	`int[] array = {1, 2, 3};`	
`L[0] = 5`	`array[0] = 5;`	
`print(L[0])`	`System.out.println(array[0]);`	
`print(L)`	`System.out.println(Arrays.toString(array));`	
`print(len(L))`	`System.out.println(array.length);`	

Lists (resizable lists)

Python	Java	Commentary
`L = []`	`List<Integer> list = new ArrayList<Integer>();`	
`L.append(0)`	`list.add(0);`	
`L[0] = 1`	`list.set(0, 1);`	
`print(L[0])`	`System.out.println(list.get(0));`	
`print(L, len(L))`	`System.out.println(list + " " + list.size());`	

Dictionaries/Maps

Python	Java	Commentary
`d = {}`	`Map<String, String> map = new HashMap<String, String>();`	
`d["hello"] = "goodbye"`	`map.put("hello", "goodbye");`	
`print(d["hello"])`	`System.out.println(map.get("hello"));`	
`print(d, d.keys())`	`System.out.println(map + " " + map.keySet());`	

Programs

It can be really useful to see Java in action. Here are some simple programs that you might find yourself referring to if you forget how to do something.

Hello World

Python	Java	Commentary

print("Hello World")	public class HelloWorld {
	    public static void main(String[] args) {
	        System.out.println("Hello World");
	    }
	}	

User Input

Python	Java	Commentary

line = input("How many dogs you got?")	import java.util.Scanner;
	public class UserInput {
	    public static void main(String[] args) {
	        Scanner s = new Scanner(System.in);
	        System.out.println("How many dogs you got?")
	        String line = s.nextLine();
	    }
	}	

Minimum of a List of Numbers

Python	Java	Commentary

def min(numbers):
    if len(numbers) < 1:
        raise ValueError()
 	
    m = numbers[0]
    for n in numbers:
        if n < m:
            m = n
 	
    return m
	public static int min(int[] numbers) {
	    if (numbers.length < 1) {
	        throw new IllegalArgumentException();
	    }
	    int m = numbers[0];
	    for (int i = 0; i < numbers.length; i++) {
	        if (numbers[i] < m) {
	            m = numbers[i];
	        }
	    }
	    return m;
	}	

Averaging Numbers from User Input

Python

```python
line = input("Type comma-sep ints: ")
grades = []
while "," in line:
    commaIndex = line.index(",")
    grade = int(
        line[0:commaIndex].strip()
    )
    grades.append(grade)
    line = line[:commaIndex+1]

grades.append(int(line))
s = 0
for x in grades:
    s += x

print("The average is " +
    str(float(s)/len(grades))
)
```

Java

```java
import java.util.Scanner;
import java.util.List;
import java.util.ArrayList;
public class Averager {
    public static void main(String[] args) {
        Scanner console = new Scanner(System.in);
        System.out.print("Type comma-separated ints: ");
        String line = console.nextLine();
        List<Integer> grades = new ArrayList<>();
        while (line.indexOf(",") != 0) {
            int commaIndex = line.indexOf(",");
            int grade = Integer.parseInt(
                line.substring(0, commaIndex).trim()
            );
            grades.add(grade);
            line = line.substring(commaIndex + 1);
        }
        grades.add(Integer.parseInt(line));
        int s = 0;
        for (int i = 0; i < grades.size(); i++) {
            s += grades.get(i);
        }
        System.out.println("The average is " +
            (s / (double) grades.size())
        );
    }
}
```

Commentary

Classes

Python	Java	Commentary
class Point(object):	public class Point {	
	private int x;	
	private int y;	
def __init__(self, x, y):	public Point(int x, int y) {	
self.x = x	this.x = x;	
self.y = y	this.y = y;	
	}	
	public Point() {	
	this(0, 0);	
	}	
def distanceTo(self, other):	public double distanceTo(Point other) {	
return math.sqrt(	return Math.sqrt(	
math.pow(self.x -...	Math.pow(this.x - other.x, 2) +	
math.pow(self.y -...	Math.pow(this.y - other.y, 2)	
)	);	
	}	
def translate(self, dx, dy):	public void translate(int dx, int dy) {	
self.x += dx	this.x += dx;	
self.y += dy	this.y += dy;	
	}	
def __str__(self):	public String toString() {	
return "("+ str(self.x) +...	return "(" + this.x + ", " + this.y + ")";	
	}	
	}	

Python	Java	Commentary
p1 = Point(5, 9)	Point p1 = new Point(5, 9);	
p2 = Point(-3, 3)	Point p2 = new Point(-3, 3);	
print("Point 1:", p1)	System.out.println("Point 1: " + p1);	
print("Distance:", p1.distanceTo(p2))	System.out.println("Distance: " + p1.distanceTo(p2));	
p1.translate(2, 2)	p1.translate(2, 2);	
print("Point 1:", p1)	System.out.println("Point 1: " + p1);	