Level 11: Sorting 1

Sorting algorithms are a set of instructions that take an array or list as an input and arrange the items into a particular order.

Sorts are most commonly in numerical or a form of alphabetical (called lexicographical) order, and can be in ascending (A-Z, 0-9) or descending (Z-A, 9-0) order.

Why Sorting Algorithms are Important

Since sorting can often reduce the complexity of a problem, it is an important algorithm in Computer Science. These algorithms have direct applications in searching algorithms, database algorithms, divide and conquer methods, data structure algorithms, and many more.

Trade-Offs of Algorithms

When using different algorithms some questions have to be asked. How big is the collection being sorted? How much memory is at disposal to be used? Does the collection need to grow?

The answers to these questions may determine what algorithm is going to work best for the situation. Some algorithms like merge sort may need a lot of space to run, while insertion sort is not always the fastest, but it doesn't require many resources to run.

You should determine what the requirements of the system are and its limitations before deciding what algorithm to use.

Some Common Sorting Algorithms

Some of the most common sorting algorithms are:

- Selection Sort
- Bubble Sort
- Insertion Sort
- Merge Sort
- Quick Sort
- Heap Sort
- Counting Sort
- Radix Sort
- Bucket Sort

(You can try to research these later on, they will help you in CP2 and CP3)

Problem to solve:

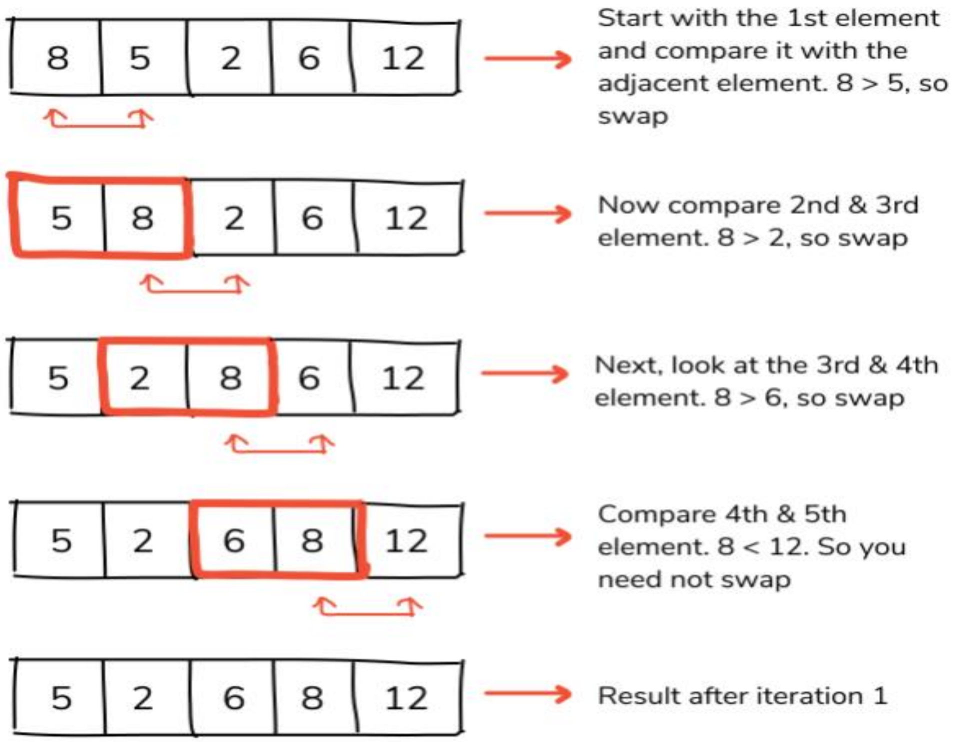
Write a Python program to sort a list of elements in a **decreasing** order using selection sort

Level 12: Sorting 2

Bubble sort works on the repeatedly swapping of adjacent elements until they are not in the intended order. It is called bubble sort because the movement of array elements is just like the movement of air bubbles in the water.

The average and worst-case complexity of Bubble sort is O(n^2), where n is the number of items.

Ex:



Start with the 1st element and compare it with the adjacent element. 8 > 5, so swap

Now compare 2nd & 3rd element. 8 > 2, so swap

Next, look at the 3rd & 4th element. 8 > 6, so swap

Compare 4th & 5th element. 8 < 12. So you need not swap

Result after iteration 1

Repeat for the second element and so on.

You can watch this video to understand more: https://www.youtube.com/watch?v=A6m-g0SPzt0

Problem to solve:

Write a Python program to sort a list of elements using the bubble sort algorithm.

Level 13: Searching

Searching Algorithms are designed to check for an element or retrieve an element from any data structure where it is stored. Based on the type of search operation, these algorithms are generally classified into two categories:

1. **Sequential Search**: In this, the list or array is traversed sequentially, and every element is checked. For example: Linear Search.
2. **Interval Search**: These algorithms are specifically designed for searching in sorted data-structures. These type of searching algorithms are much more efficient than Linear Search as they repeatedly target the center of the search structure and divide the search space in half. For Example: Binary Search.

Problem to solve:

Write a Python program to search through a list sorted in a decreasing order using **binary search.**

Level 13: Classes and Objects

A Class is like an object constructor, or a "blueprint" for creating objects.

Create a Class

To create a class, use the keyword class:

```
class MyClass:
  x = 5
```

Create Object

Now we can use the class named MyClass to create objects:

```
p1 = MyClass()
print(p1.x)
```

The __init__() Function

The examples above are classes and objects in their simplest form, and are not really useful in real life applications.

To understand the meaning of classes we have to understand the built-in __init__() function.

All classes have a function called __init__(), which is always executed when the class is being initiated.

Use the __init__() function to assign values to object properties, or other operations that are necessary to do when the object is being created:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

p1 = Person("John", 36)

print(p1.name) -> John
print(p1.age) -> 36
```

Object Methods

Objects can also contain methods. Methods in objects are functions that belong to the object.

Let us create a method in the Person class:

```
class Person:
  def __init__(self, name, age):
    self.name = name
    self.age = age

  def myfunc(self):
    print("Hello my name is " + self.name)
```

```
p1 = Person("John", 36)
p1.myfunc() -> Hello my name is John
```

Note: The self parameter is a reference to the current instance of the class and is used to access variables that belong to the class.

Problem to solve:

Create a class called Student that has the following attributes:

- Name
- ID
- Major
- GPA
- Credits taken

1-Create a method in this class that will display the information of a student as follows:

```
Student id: 201400000
Student Name: FirstName LastName
Student Major: Computer Science
Student GPA: 3.8
```

2-Create a method that will take the grade of a class and the number of credits, add the number of credits to the total credits taken by the students, and adjust the GPA based on the new grade.

How to calculate new gpa:

Ex: new grade=”A”, nbr_of_credits: 3, old_gpa=3.7, old_nbr_of_credits=30

“A” is 4 points so nbr_of_points=4

New_gpa= ((old_gpa* old_nbr_of_credits)+ nbr_of_credits*nbr_of_points)/(nbr_of_credits + old_nbr_of_credits)

New_gpa=((3.7*30)+3*4)/33

New_gpa= 3.72

Create an object of this class

fill it with your information

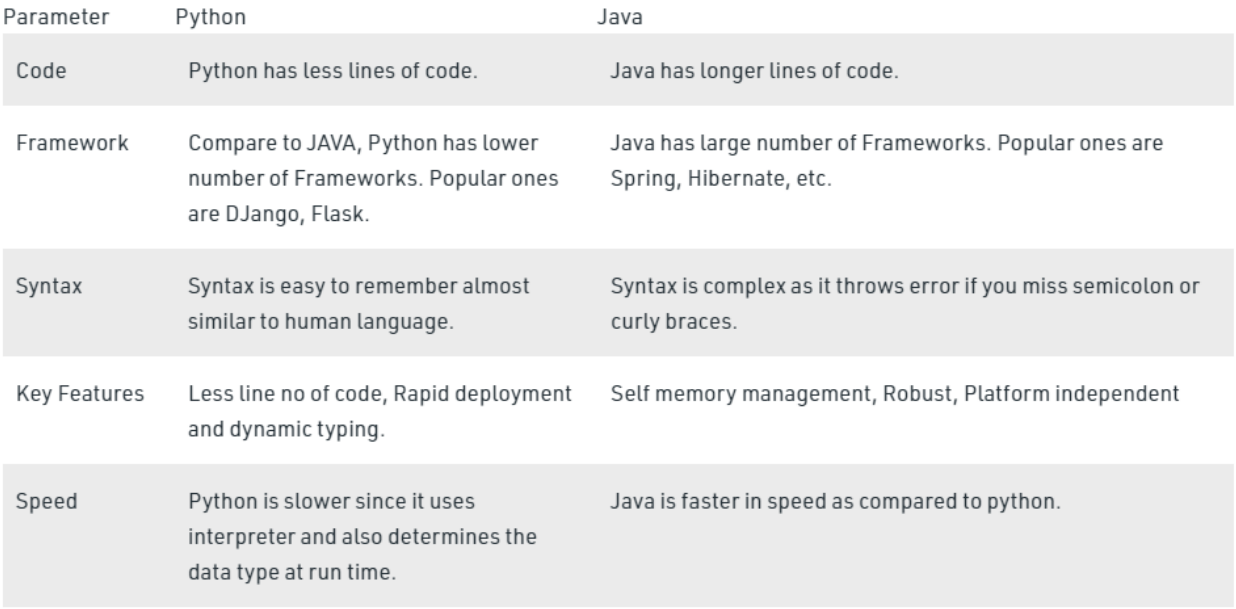
use the method in the class to display the information.

Add a new grade “A” of 3 credits.

display the information again

Level 13: From Python to Java (Simple)

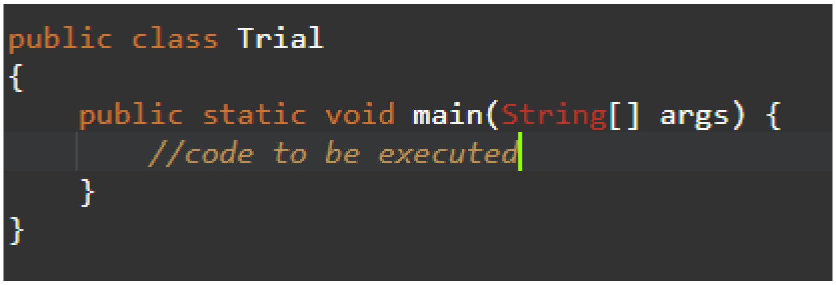
Java: Java is a high-level, object-oriented programming language which was originally developed by James Gosling at Sun Microsystems in 1995. Java has a syntax similar to C and C++ but with low-level difficulties. Java is platform-independent (WORA – Write Once Run Anywhere) meaning compiled java code can run on different platforms without recompilation.



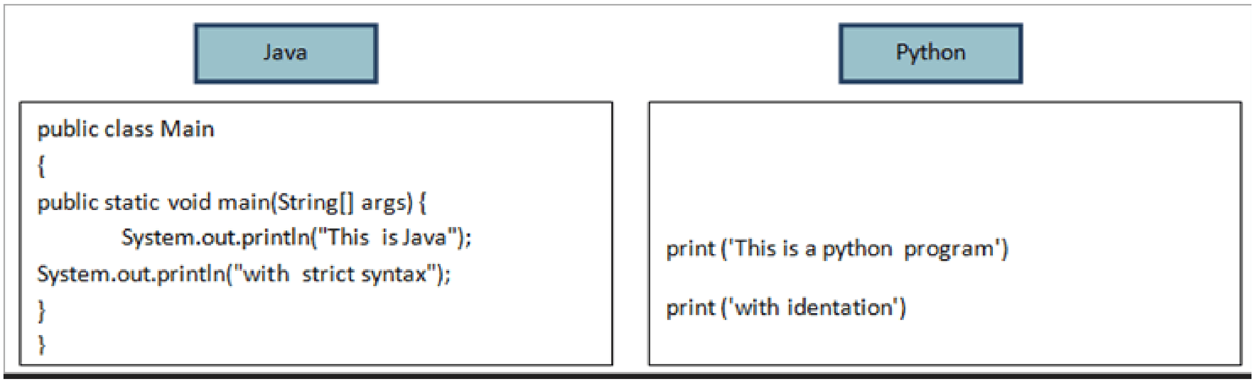
Parameter	Python	Java
Code	Python has less lines of code.	Java has longer lines of code.
Framework	Compare to JAVA, Python has lower number of Frameworks. Popular ones are DJango, Flask.	Java has large number of Frameworks. Popular ones are Spring, Hibernate, etc.
Syntax	Syntax is easy to remember almost similar to human language.	Syntax is complex as it throws error if you miss semicolon or curly braces.
Key Features	Less line no of code, Rapid deployment and dynamic typing.	Self memory management, Robust, Platform independent
Speed	Python is slower since it uses interpreter and also determines the data type at run time.	Java is faster in speed as compared to python.

Syntax:

In JAVA we must always write our code inside a class, and the code that needs to be executed should always be in the **main method**:



```
public class Trial
{
    public static void main(String[] args) {
        //code to be executed
    }
}
```



```
public class Main
{
public static void main(String[] args) {
          System.out.println("This  is Java");
System.out.println("with  strict syntax");
}
}
```

```
print ('This is a python  program')

print ('with identation')
```

To print in JAVA, we use System.out.println() as shown above

Variables:

In Java we must always specify the type of the variable before the variable name. Ex:

int var1 = 23;

Semicolon:

In Java, at the end of every syntax we put a semi-colon “;”.

For more info, you can check the file “A comparison of the Syntax of Python and JAVA” that is inside the folder where you found this challenge.

Problem to solve:

Turn this program from python to java:

```
var1=78
var2=23
var3="Hello World"
var4=8.9
print(var1+var2)
print(var2*var4)
print(var3)
```

Try this in python:

```
myvar=23
print("hello"+myvar)
```

Try this in Java:

```
int myvar=23;
System.out.println("Hello"+myvar);
```

What do you notice?

You can use this online compiler to compile your java code:
https://www.onlinegdb.com/online_java_compiler

Next Step:
Organize all your codes for all levels in a proper way, put them in a folder and send them to all 3 of the lab instructors by email.

References:

https://www.freecodecamp.org/news/sorting-algorithms-explained-with-examples-in-python-java-and-c/

https://www.geeksforgeeks.org/searching-algorithms/#:~:text=Searching%20Algorithms%20are%20designed%20to%20check%20for%20an,is%20traversed%20sequentially%20and%20every%20element%20is%20checked.

https://www.javatpoint.com/bubble-sort

https://www.w3schools.com/python/python_classes.asp