ICS 365 Organization of Programming Languages

Fall 2022

Dr. Jigang Liu

Comparative study of three programming languages:

Java, Python, and Solidity

Pong Shoua Lee

Metropolitan State University

Nov. 22, 2022

There are many types of areas that many computer science students need to know, such as how to first run a simple output displaying “Hello World” in their IDLE, how to compute simple algebraic equations like 1 + 2, or even knowing what computer programming language to start out with. There are a multitude of programming languages to choose from, for example python. The goal of this topic is to compare three computer programming languages: Java, Python, and Solidity.

Python is a high-level object-oriented programming language. Python was designed to be readable and easier to use for everyone regardless of their knowledge of the programming language. With Python being so beginner friendly it has the ability to be portable, meaning it can run on multiple hardware platforms making it useful and very versatile. Python being both easy to use and portable means it can also reduce the cost of maintenance, low cost of maintenance means faster production and development. Python can also be integrated easily with other computer programming languages such as Java.

Java, while compared to Python, is more of the standard of how computer programming languages are written. While it lacks the readability of Python it instead has better performance and distribution across multiple platforms compared to Python. Java emphasizes object-oriented programming. Knowing object-oriented programming helps lower the cost of development and the time it needs to be integrated. Combining all the strengths of Java is one of the reasons why it’s one of the most popular and most compatible computer programming languages out there. While Java is mainly used for development there are other types of computer programming languages such as Solidity that can be used as a blockchain.

Solidity is a contract-oriented programming language. Meaning it executes actions to terms of contracts without using a third-party. It cuts out the middleman and goes straight to the buyer and seller. Even though Solidity is a contract-oriented programming language, it is still a high-level language. It takes influence from C++, Python and JavaScript. Solidity is designed around the Ethereum Virtual Machine or EVM[5]. EVM is a runtime environment for smart contacts in Ethereum. It protects contacts from unwanted onlookers and even cuts out the third party that is usually required for most contacts dealing with money being moved around. While the history of Solidity is quite new compared to the other two programming languages it still has an impact on the software industry.

Solidity was first proposed by Gavin Wood, a computer scientist in August 2014[3]. The purpose of Solidity was to run the Ethereum blockchain. It was to go hand in hand with Bitcoin, the first ever cryptocurrency. However, as Solidity was only proposed in August of 2014 it was not adopted until the following fall of 2014 by a rival platform, Monax. Then after a few more months Solidity was officially released in August of 2015. Taking only a whole year from an idea to an official computer programming language is quite an achievement. While Solidity took a year to be official, other languages would not have the same type of development.

Python was first worked on in the 1980’s by Guido Van Rossum [1]. For Guido, Python was a side project and was something for him to work on to pass the time. Python was officially released in 1991, and at that time other big computer programming languages such as Java, C++, and C used more codes to express concepts, while Python was doing the opposite.

The start of Java was originally designed for interactive television. Since the idea was to advance at the time it was scrap and transited elsewhere. Java was then developed for internet programming. Starting in the early 1990’s it was developed by James Gosling and his team members. He and his team wanted Java programming to be "Simple, Robust, Portable, Platform-independent, Secured, High Performance, Multithreaded, Architecture Neutral, Object-Oriented, Interpreted, and Dynamic" [2]. Java was released in 1995. Java was also called a few different names before Gosling settled with the name Java. It was first called Greentalk after his team’s name Green Team. After that name it was called “Oak”[4], named after the oak tree, but had to change the name due to a trademark by a company named “Oak Technologies”.

Each of the three computer programming languages all have different beginnings and very different ways of how each of them is implemented in code. Python, a simple and powerful language, Java a very robust and reliable programming language, and finally Solidity the newest of the groups used for blockchains.

Here is a code comparison among the three computer programming languages. Starting with Python, a simple output of “Hello World” can be written as: print("Hello World."). Next is the Java computer programming language.

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

Here is the code for the output of “Hello World” using Java. Uses more code and needs a class for it to run the code. Finally, up next is Solidity.

```solidity
// SPDX-License-Identifier: MIT
pragma solidity 0.8.13;
contract HelloWorld {
    function sayHelloWorld() public pure returns (string memory) {
        return "Hello World";
    }
}
```

Image taken from (https://blog.chain.link/how-to-create-a-hello-world-smart-contract-with-solidity/)

Compared to Java it may look the same but upon closer inspection the syntax of the code is very different. Solidity needs a function called sayHelloWorld() that returns a string memory, rather than a print method like Java or Python.

Next comparison is the if-statements.

```python
a = 200
b = 33
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
else:
  print("a is greater than b")
```

Image taken from (https://www.w3schools.com/python/python_conditions.asp)

As an example, from above Python does not require any objects to be cast as an integer or as a string. The language already knows the object is being assigned as. Next is Java.

```java
int time = 22;
if (time < 10) {
  System.out.println("Good morning.");
} else if (time < 20) {
  System.out.println("Good day.");
} else {
  System.out.println("Good evening.");
}
```

Image taken from (https://www.w3schools.com/java/java_conditions.asp)

Now compared to Python, Java here needs to have its object be cast first. Even though it's a small difference between the two languages, both are still following the same format for the if-else statements. Both print statements are indented below the respective if body field. Finally, Solidity if-else statement.

```solidity
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract IfElse {
    function foo(uint x) public pure returns (uint) {
        if (x < 10) {
            return 0;
        } else if (x < 20) {
            return 1;
        } else {
            return 2;
        }
    }

    function ternary(uint _x) public pure returns (uint) {
        // if (_x < 10) {
        //     return 1;
        // }
        // return 2;

        // shorthand way to write if / else statement
        // the "?" operator is called the ternary operator
        return _x < 10 ? 1 : 2;
    }
}
```

Image taken from (https://solidity-by-example.org/if-else/)

Solidity also retains the same structure as both Python and Java, as indicated by the indentation for the output or in this case the return statement. The biggest difference here is that Solidity needs another method to use the if-else method. In turns of coding inheritance using the same if-else method isn't a bad thing but it depends on what the if-else method is being used for.

Next comparison is the while loop.

```python
i = 1
while i < 6:
  print(i)
  i += 1
```

Image taken from (https://www.w3schools.com/python/python_while_loops.asp)

The example above is a Python while loop. Starts at 1 and ends until it hits 6. The output prints out the numbers from 1 to 5 as the less than symbol does not include the 6 number.

```java
int i = 0;
while (i < 5) {
  System.out.println(i);
  i++;
}
```

Image taken from (https://www.w3schools.com/java/java_while_loop.asp)

Above the image is a Java for loop. Looks exactly like Python in terms of format with small details Java needs for it to run the program.

```
// SPDX-License-Identifier: MIT
pragma solidity ^0.8.13;

contract Loop {
    function loop() public {
        // for loop
        for (uint i = 0; i < 10; i++) {
            if (i == 3) {
                // Skip to next iteration with continue
                continue;
            }
            if (i == 5) {
                // Exit loop with break
                break;
            }
        }

        // while loop
        uint j;
        while (j < 10) {
            j++;
        }
    }
}
```

Image taken from (https://solidity-by-example.org/loop/)

Now for Solidity, the biggest take away from the code provided above is that it was strongly advised not to use a while loop with writing it in Solidity. The link above states that if the loop is unbounded, it can hit the gas limit, thus causing a transaction to fail. So, while it can do a while loop it is better to do a for loop instead.

Last comparison for all three computer programming languages is input and output.

```
# Taking input from the user
name = input("Enter your name: ")

# Output
print("Hello, " + name)
print(type(name))
```

Image taken from (https://www.geeksforgeeks.org/input-and-output-in-python/)

The code written above using Python asks for the user's name for the input and then outputs the user's name, first with "Hello, User" and then second with "User".

```
import java.util.Scanner;

class Input {
    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter an integer: ");
        int number = input.nextInt();
        System.out.println("You entered " + number);

        // closing the scanner object
        input.close();
    }
}
```

Image taken from (https://www.programiz.com/java-programming/basic-input-output)

The Java code written above needs to import the scanner for it to be able to get a user's input. Once Java has imported the scanner it must then create a scanner object which for this example can be named as "input". For the example above it's asking for a integer so the scanner needs to make sure that the object called "input" needs to have

nextInt() to catch the integer. After the scanner is done catching the user's input the object "input" then needs to be closed. Finally next is Solidity.

```
function singleIncomingParameter(int _data) returns (int
_output) {
    return _data * 2;
}
```

Image taken from (https://subscription.packtpub.com/book/application-development/9781788831383/7/ch07lvl1sec79/function-input-and-output)

The small code written above in Solidity almost acts like Java, asking for an integer and returns the user's input to an output that is then multiplied by 2. What makes Solidity interesting is that it needs functions for most of the code it needs to run.

When comparing all three computer programming languages, Java, and Solidity both look the same in terms of how they both look and return certain data, but for Solidity "return" is used as its output. One thing that they all share is how they indent certain key syntax like the for loop and even the if-statements. In terms of which is the best is based on preference and what the programmer is trying to accomplish.

Here is a table that summarizes the three computer programming languages:

Programming language	Where to download	Latest version	Supporting sites	Common IDE	Main Strengths	Key weakness	Learning curve	Influences	Popular Usage
Java	https://www.java.com/en/download/	Version 8	https://docs.oracle.com/javase/tutorial/index.html	Eclipse	*Simple *Is Object-oriented *Secure *Easy to maintain	*Complex codes *Requires significant memory space *Can be Slow and perform poorly	Medium	C++ C	*Data Structures *Enterprise applications *Sever-side applications
Python	https://www.python.org/downloads/	3.11.0	https://www.python.org/doc/	IDLE	*Easy to learn *Object-Oriented *Does not require a complier	*Issues with database layers *Not strong in mobile development	Easy	ABC programming language	*Websites *Task automation *Website and software
Solidity	https://docs.soliditylang.org/en/v0.8.17/installing-solidity.html#installing-solidity	0.8.17	https://docs.soliditylang.org/en/v0.8.17/	Remix	*Contract-oriented programming *Inheritance support	*Very new *Small library	Medium	Mainly C++ but also borrowed concepts from Python and JavaScript	*Block chain *Smart contact

Comparing all three of the computer programming languages, Java, Python, and Solidity there is subtle difference on how they code, what they do and how they interpret code. All three have their own purposes in the software development world. Python being great at making automation applications, Java works wonders in data structures,

and finally Solidity, the newest of the three, is showing the world what it can do with blockchain and smart contacts. A simple data table that shows how they compare to one another doesn't really show make they are capable of. It's the surface level of what they can do and what weakness they have. All three of the computer programming languages are more than capable of showing more than their weaknesses. It all comes down to the software developer and what they want to accomplish with the given tools handed to them, whether it be making a blockchain using Solidity or even using Java to find the most data in an enterprise environment.

[1] Vasilisa Sheromova “Guido is the main creator of the language” https://exyte.com/blog/a-brief-history-of-python

[2] JavaTpoint https://www.javatpoint.com/history-of-java

[3] Matt Hussey “Solidity language proposed by Gavin Wood” https://decrypt.co/resources/solidity

[4] freejavaguide “Java was started as a project called "Oak" by James Gosling in June 1991” https://www.freejavaguide.com/history.html

[5] Vedant Chainani ”The Ethereum Virtual Machine (or) EVM” https://dev.to/envoy_/history-and-origin-of-solidity-2mhl

Other resources

https://pythoninstitute.org/about-python

https://www.tutorialspoint.com/java/java_overview.htm

https://www.w3schools.com/java/java_intro.asp

https://blog.chain.link/how-to-create-a-hello-world-smart-contract-with-solidity/

https://techvidvan.com/tutorials/pros-and-cons-of-java/

https://kcpelearning.com/showcourse/python3/strengths-and-weaknesses

https://www.qualium-systems.com/blog/blockchain/solidity-ethereums-programming-language-for-smart-contraction/

https://www.google.com/books/edition/Python_Programming_Deep_Learning/lEtVEAAAQBAJ?hl=en&gbpv=1&dq=python++strengths+and+weaknesses&pg=PT91&printsec=frontcover