

# Class02

## Pseudocode

# Homework:

<https://github.com/prateek27/java-nov-21>

# Warm-up Puzzle!

## Problem Statement:

There is a circular jail with 100 cells numbered 1-100. Each cell has an inmate and the door is locked. One night the jailor gets drunk and starts running around the jail in circles. In his first round he opens each door. In his second round he visits every 2nd door (2,4,6---) and shuts the door. In the 3rd round he visits every 3rd door (3,6,9---) and if the door is shut he opens it, if it is open he shuts it. This continues for 100 rounds (i.e. 4,8,12 ---; 5,10,15 ---; ---; 49,98 etc.) and exhausted the jailor falls down. How many prisoners found their doors open after 100 rounds?

## Homework

Take input N followed by N more numbers, find the sum of those numbers.

eg N = 4

10, 20, 30 , 50

Output : 110

## Homework

Find LCM of two numbers A and B example -

A = 10

B = 20

Output:

20

## Homework (Optional)

Print all Primes upto N.

$N = 10$

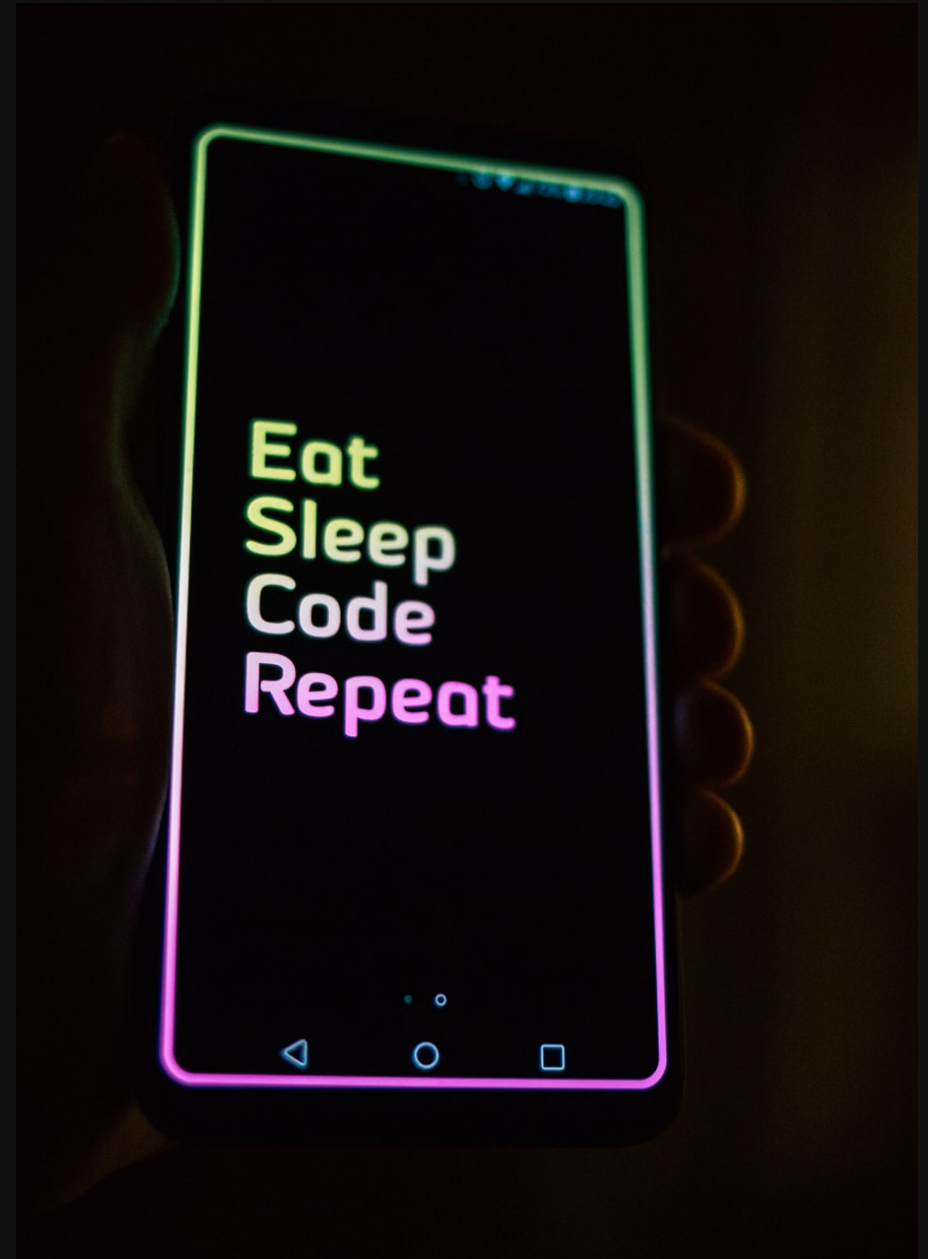
Output:

2, 3, 5, 7

# Lecture - 2

# Pseudocode

Human readable description  
of an algorithm





# It's helpful!

- Language Independent.
- Structure your code before writing it.
- Fastest way to verify / get a review.



```
2 const fetch = require('node-fetch');
3 const log = require('log');
4 let embed;
5
6 function transformIt, ...transformations {
7   // Promise.resolve to promise:
8   return transformations.reduce((prev, transform) => {
9     return transform(prev);
10   });
11
12 function removeWhiteSpace(prev) {
13   return prev.then(() => {
14     const children = $(header).children();
15     if (children.length) {
16       $(header).html(children.html());
17     }
18     return header;
19   });
20 }
21
22 return Promise.resolve();
23 });
```



# Pseudocode

Let's define our own instruction set

- **Input** [ read N ]
- **Assignment** [ Sum = 0 ]
- **Output** [ print Sum]
- **If Else**
  - [ if I < N then ... end else then ... end ]
- **While Loop**
  - [ while I < N do ... end ]
- **Exit** [exit]



# Prime Number

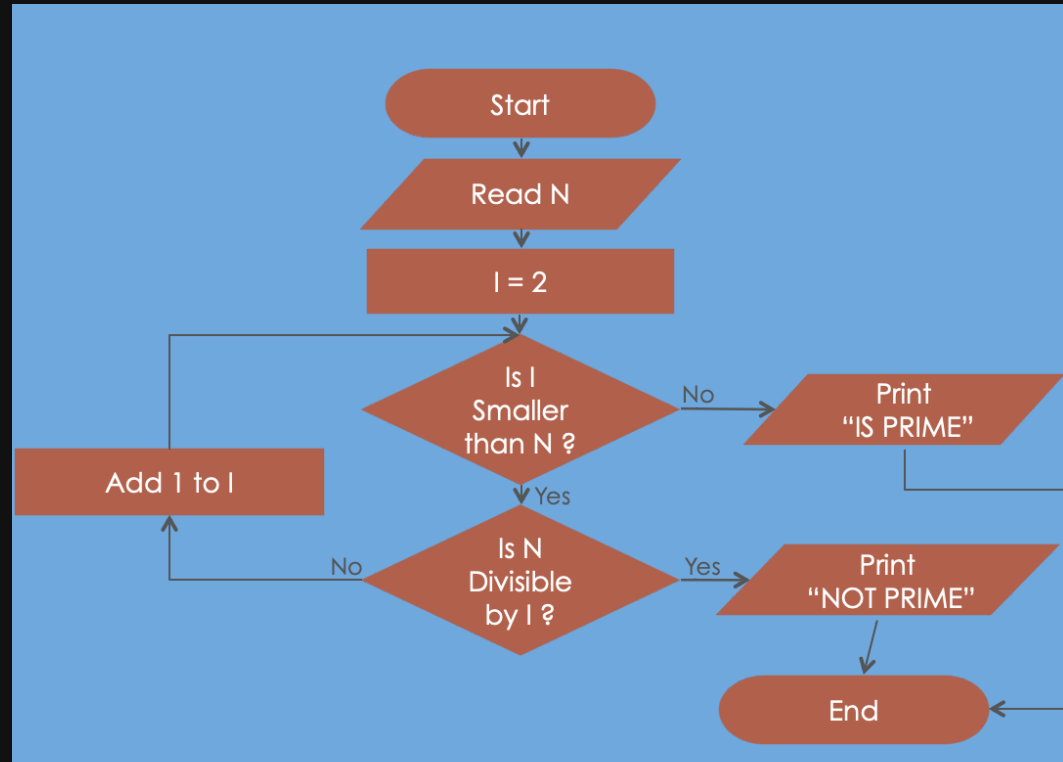
Pseudocode to check if number is prime or not.

## Input

11

## Output

Yes



```
read N
```

```
i = 2
```

```
while i < N do
```

```
    if N is divisible by i then
```

```
        print "NOT PRIME"
```

```
    exit
```

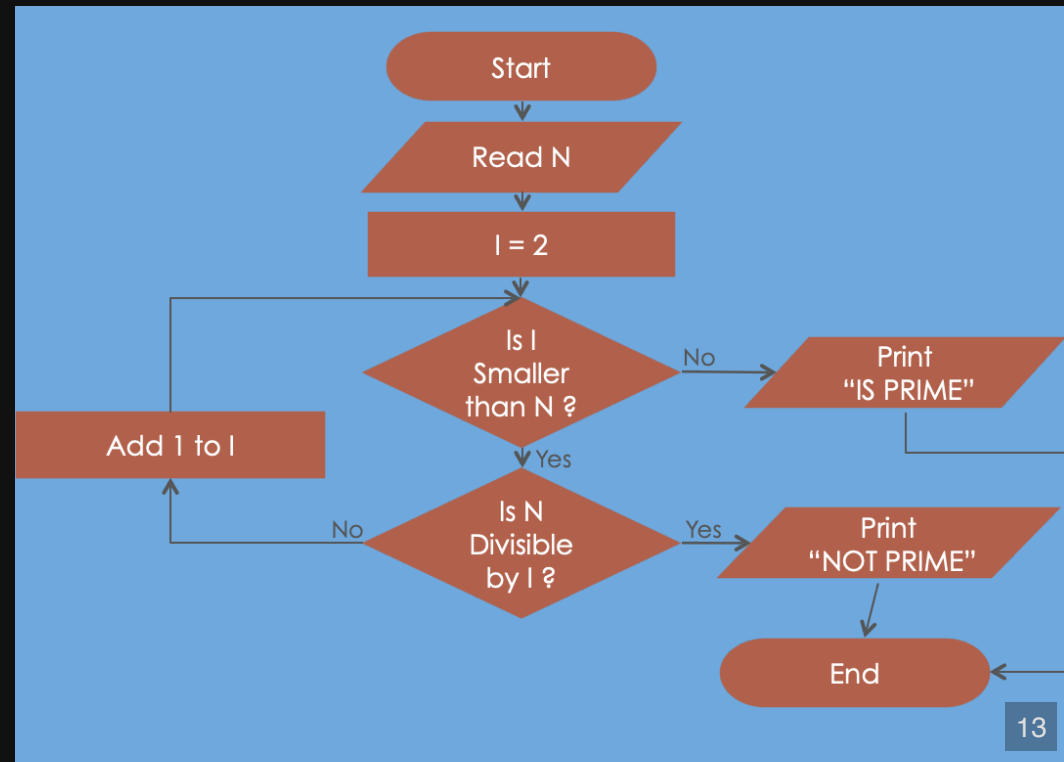
```
end
```

```
    i = i + 1
```

```
end
```

```
print "IS PRIME"
```

```
exit
```





# Number Pattern

Print the following pattern for a given N.

N = 4

1

2 3

4 5 6

7 8 9 10

1

2 3

4 5 6

7 8 9 10

```
read N
Row = 1
Value = 1

while Row <= N do
    Col = 1
    while Col <= Row do
        print Value
        Value = Value + 1
        Col = Col + 1
    end

    print "\n"
    Row = Row + 1
end
exit
```





# Number Pattern - II

Write pseudocode to **print the following pattern!**

1

232

34543

4567654

567898765



# Time to Try!

- Sum of Digits of a Number
- Swap two numbers
- Half Diamond Pattern

N = 3

\*

\*\*\*

\*\*\*\*\*

# Next Lecture 3

## Programming in

### JAVA



# Binary Number System



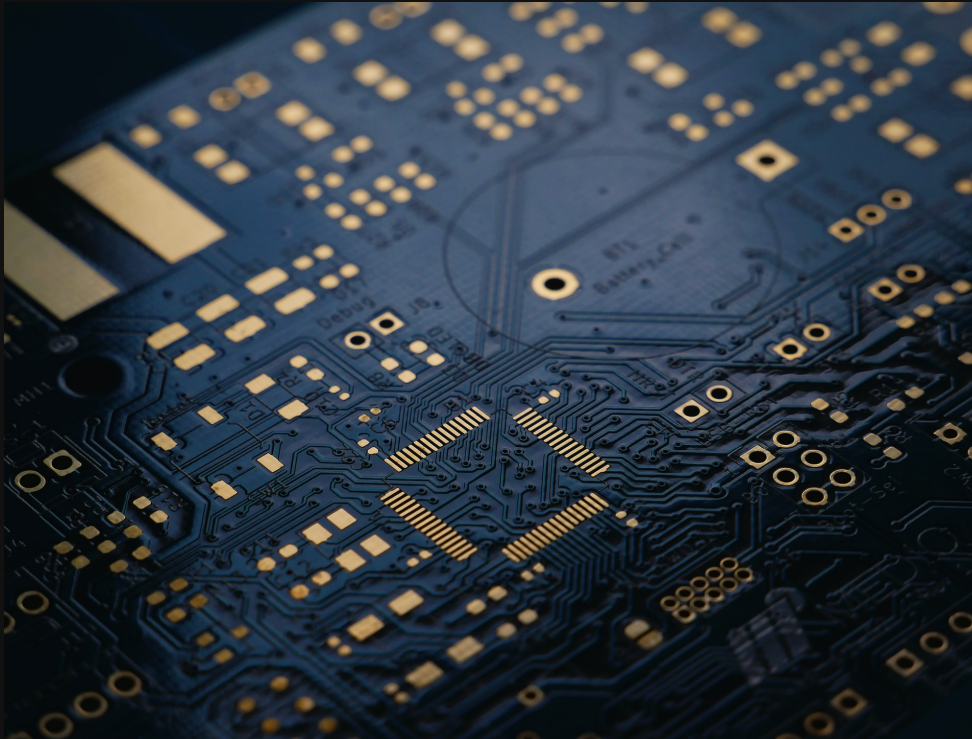
## Story behind

We humans use a decimal, or base-10, numbering system, presumably because people have 10 fingers

Early computers were designed around the decimal numbering system. This approach made the creation of computer logic capabilities unnecessarily complex and did not make efficient use of resources. (For example, 10 vacuum tubes were needed to represent one decimal digit.)



To deal with the basic electronic states of on and off, Von Neumann suggested using the binary numbering system



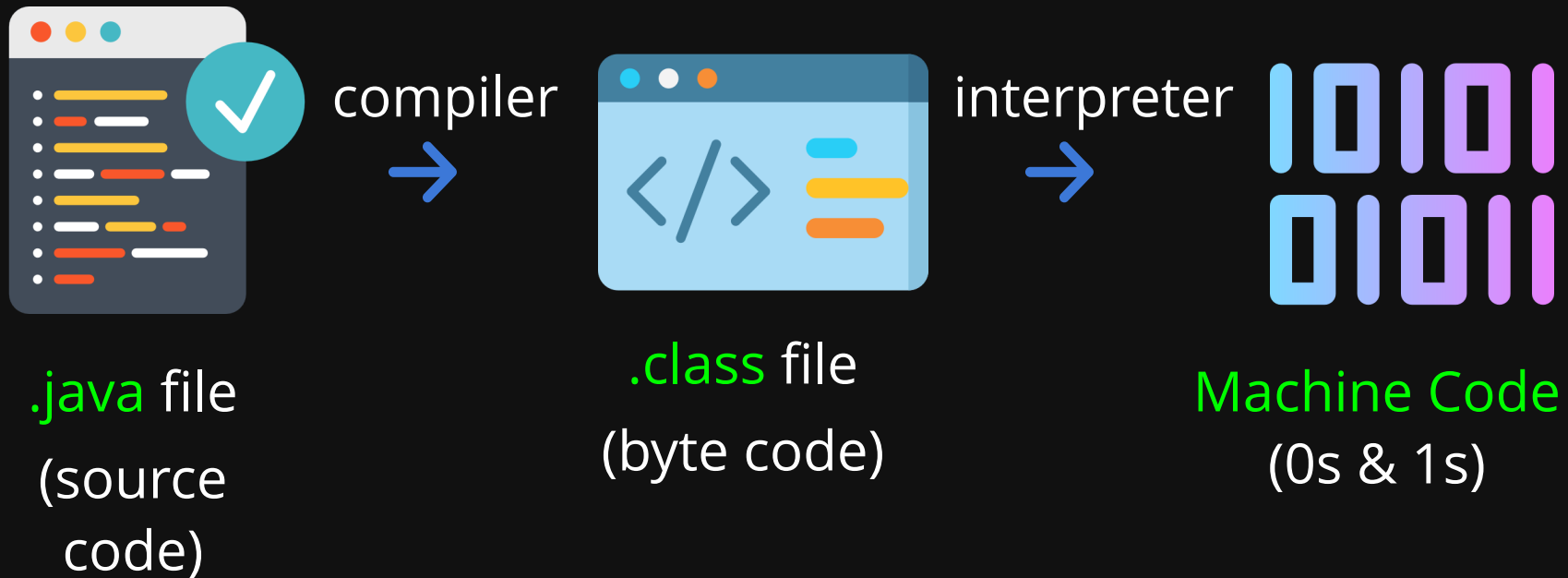
# Binary Number System



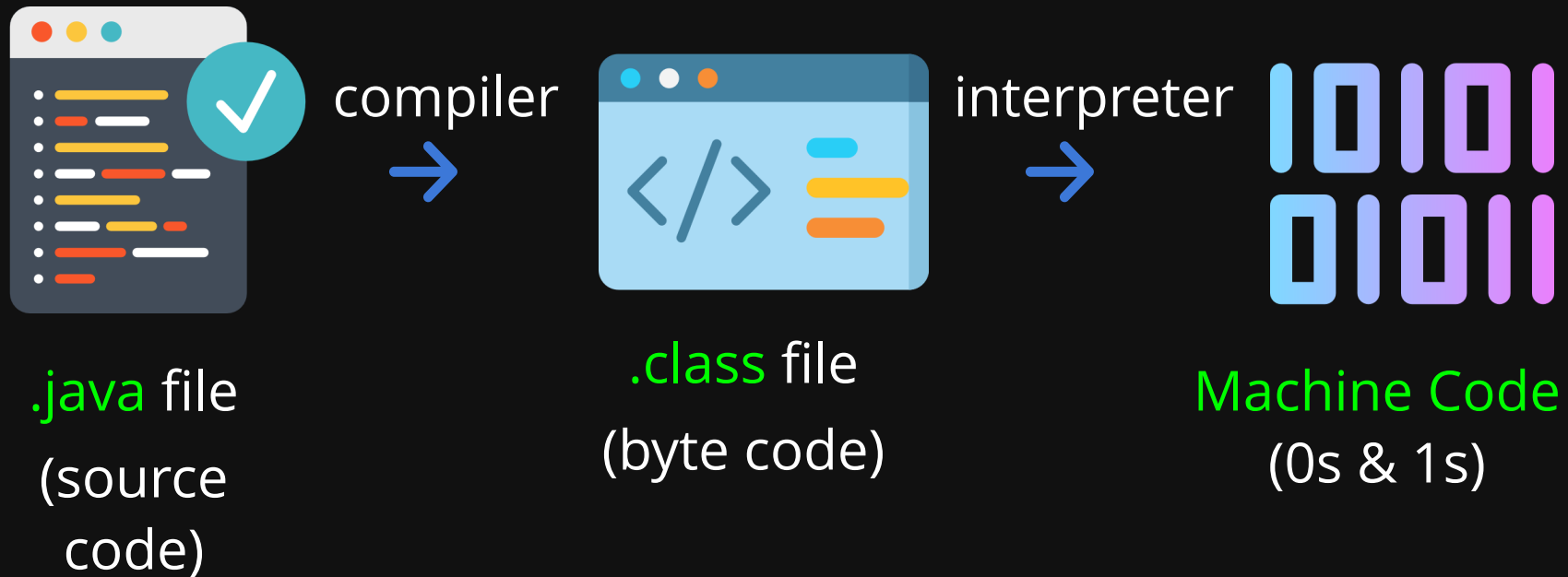


# Programming in JAVA

What happens behind the scenes?  
Why Java is Platform independent.



What happens behind the scenes?  
Why Java is Platform independent.



Byte Code doesn't run directly, we need a JVM (Java Virtual Machine) to run this code

# Platform Independence?

Byte-Code can run all  
operating systems

# C++

C/C++ Compiler generates a .exe file which is platform dependent.

# Python

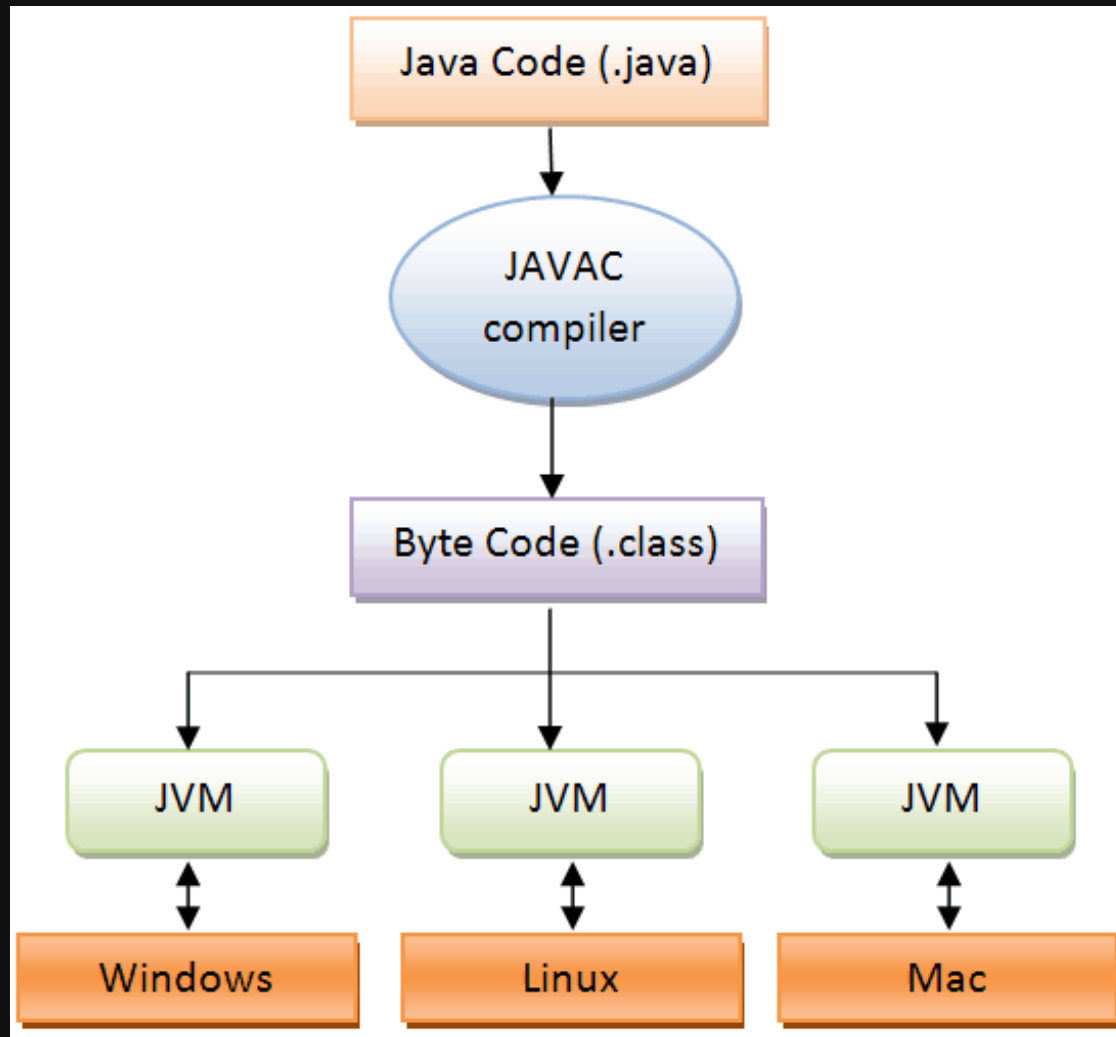
Python is interpreted language, instructions are executed line by line.

# Java

In Java we get bytecode, JVM converts this bytecode into machine code.

To run byte-code, we need a JVM installed on a machine.

Java is platform independent but JVM is platform dependent.



# Architecture

JDK = JRE + Development Tools

JRE = JVM + Library Classes

Java Virtual Machine (JVM)

JIT Compiler  
(Just In Time)





Consists of Development tools & environment to run the Java program.

1. Development Tools
2. Compiler (javac)
3. Archiver (jar)
4. docs generator (javadoc)
5. interpreter / loader



Provides environment to only **run** the program.

JRE = JVM + Additional technologies & features

1. Deployment solutions
2. Development toolkits & libraries
3. Integration libraries
4. Language and utility libraries  
(such as collections framework)

JVM 🤔

Java Virtual Machine



# JVM

Java Virtual Machine, or JVM, loads, verifies and executes Java bytecode. It is known as the interpreter or the core of Java programming language because it executes Java code.

- **Class Loader Subsystem** is responsible for loading, linking and initializing a Java class file (i.e., “Java file”), otherwise known as dynamic class loading.
- **Runtime Data Areas** contain method areas, PC registers, stack areas and threads.
- **Execution Engine** contains an interpreter, compiler and garbage collection area.



# Summary

The JRE combines Java code created using the JDK with the necessary libraries required to run it on a JVM and then creates an instance of the JVM that executes the resulting program.

JVMs are available for multiple operating systems, and programs created with the JRE will run on all of them. In this way, the Java Runtime Environment is what enables a Java program to run in any operating system without modification.

# Software Tools [OPTIONAL] 🤔

IntelliJ Idea (IDE)

<https://www.jetbrains.com/idea/download>

(Free Community Edition)

**JDK**

<https://www.oracle.com/java/technologies/downloads/#java16>

Recommended for beginners

## Coding Minutes IDE

[ide.codingminutes.com](https://ide.codingminutes.com)



**No Installation Required!**

Directly Run | Save | Share Codes!

# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6
7     }
8 }
```



# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6         //Your Code goes here
7     }
8 }
```

# First Java Program

## Boilerplate Code

```
1 //Boilerplate Code
2 class Main
3 {
4     public static void main (String[] args)
5     {
6         //Your Code goes here
7         System.out.println("Hello World");
8     }
9 }
```

Let's Solve Problems & learn **Java syntax!**

1. Hello Java
2. Variables
3. Loops
4. Typecasting
5. Problems - Prime Numbers, Patterns, Simple Interest

# Ternary Operator

```
variable = (condition) ? expressionTrue : expressionFalse;
```