

CHANDAN MUKHERJEE

**MTech (IT), BE (Computer Science)
SCJP (Java), Oracle (SQL) GLOBAL CERTIFIED
Microsoft Certified Innovative Educator
Corporate Trainer
chan.muk@gmail.com**

JAVA





A Brief History of JAVA

- ❖ Its original name was **Oak**,
- ❖ was developed as a part of the Green project at Sun Microsystem.
- ❖ It was started in **December '90** by **Patrick Naughton, Mike Sheridan** and **James Gosling** and was chartered to spend time trying to figure out what would be the "next wave" of computing and how we might catch it.
- ❖ They came to the **conclusion** that at least one of the waves was going to be the convergence of **digitally controlled consumer devices and computers**.



JAVA Compiler

- JAVA 8 - March 2014 (LTS – Long Term Support Version)
- Extended Support till December 2030
- <https://www.oracle.com/java/technologies/java-se-support-roadmap.html>

✓ JDK – Java Development Kit - compiler

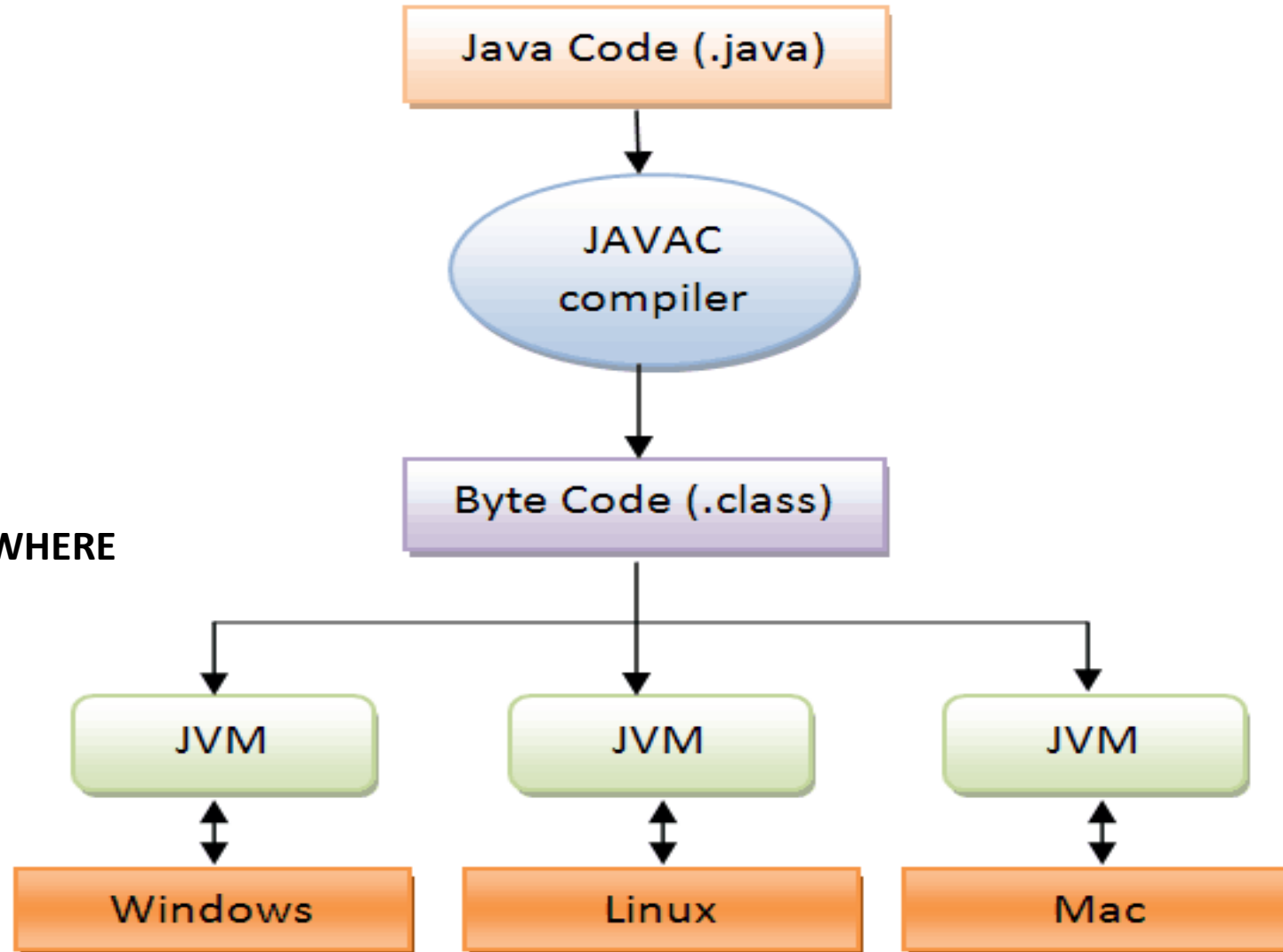
✓ JRE – Java Runtime Environment

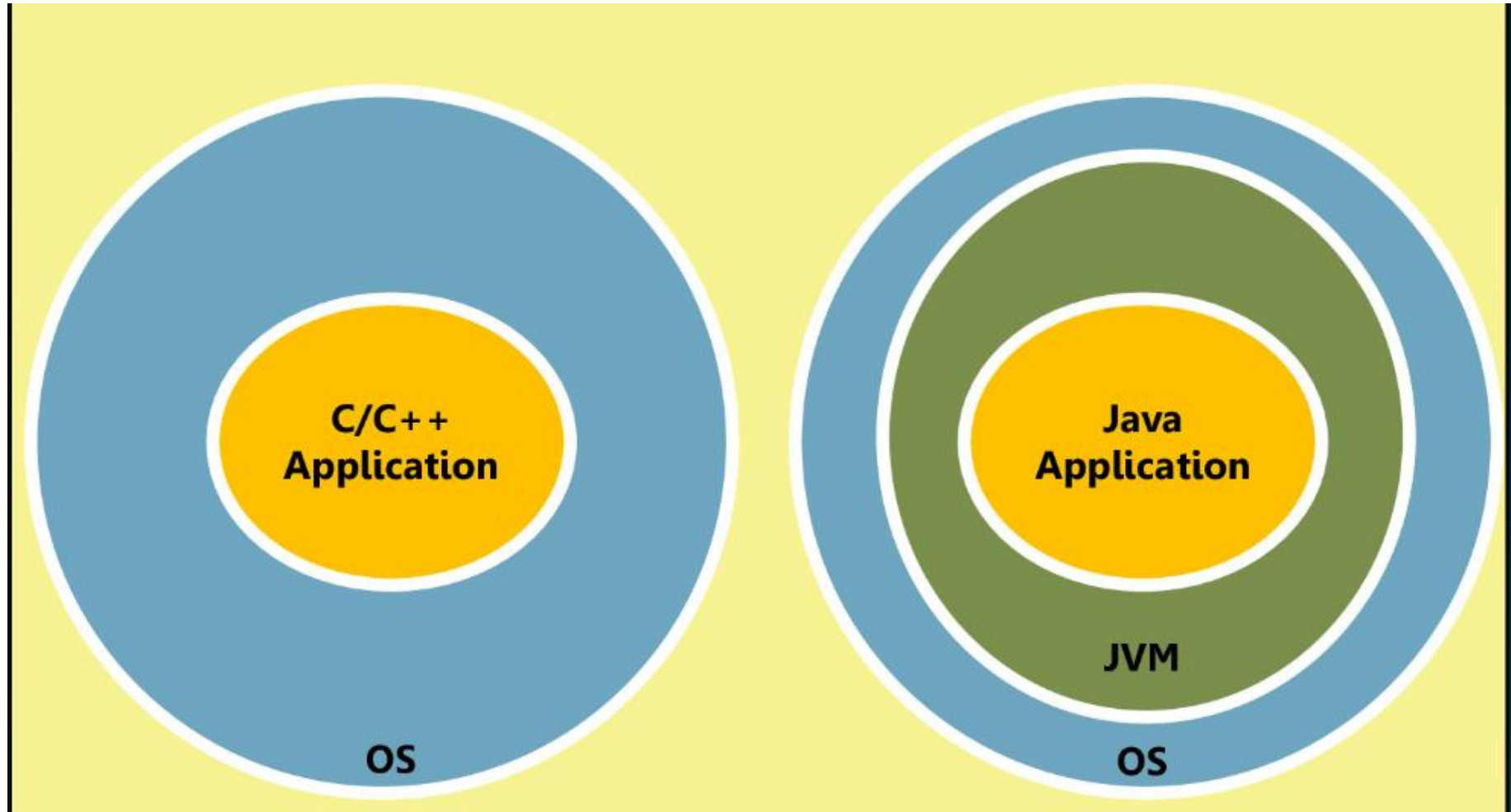
<https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html>



JAVA Program Execution

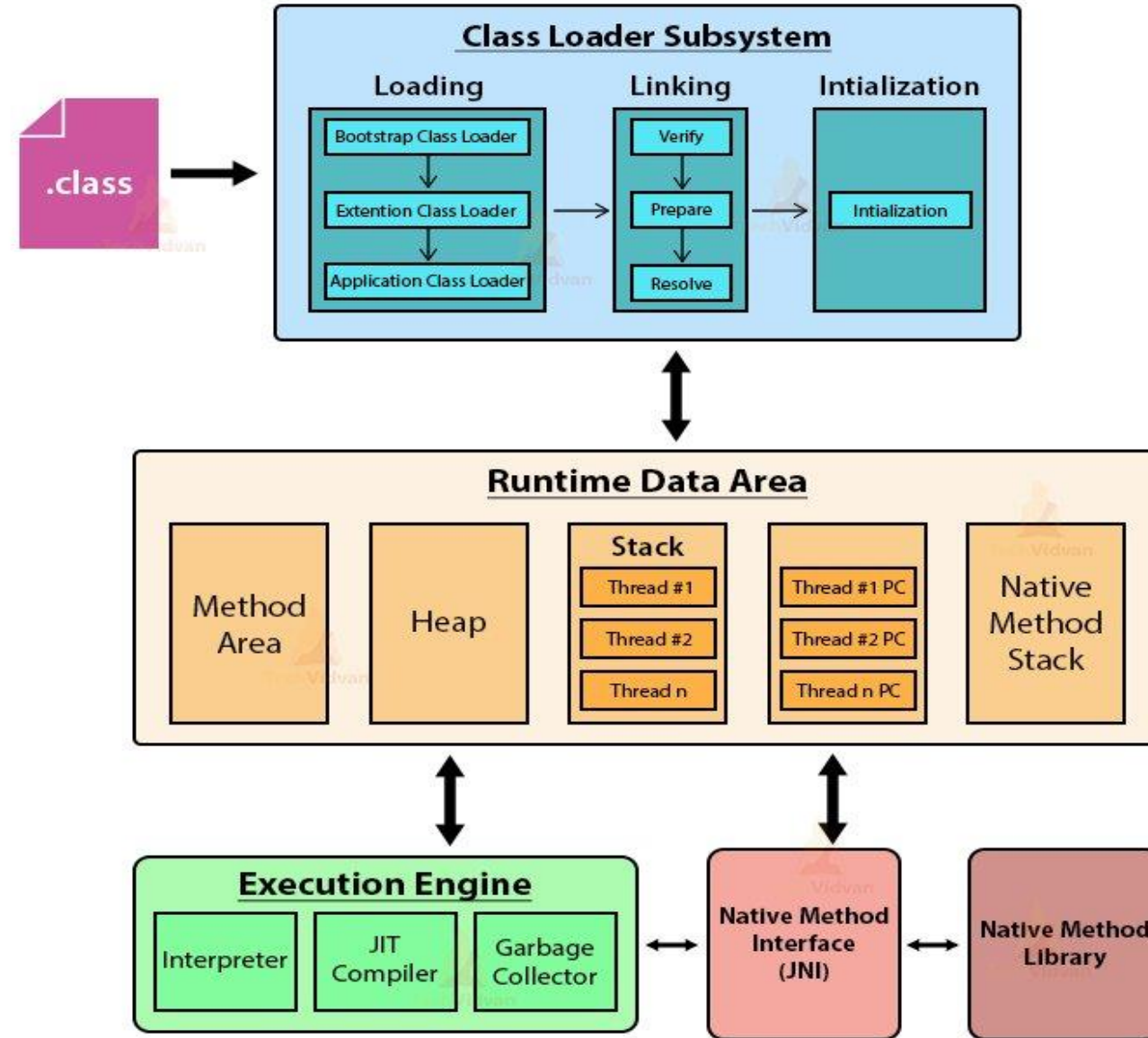
WORA
WRITE ONCE RUN ANYWHERE







JVM Model





JAVA IDE

(Integrated Development Environment)



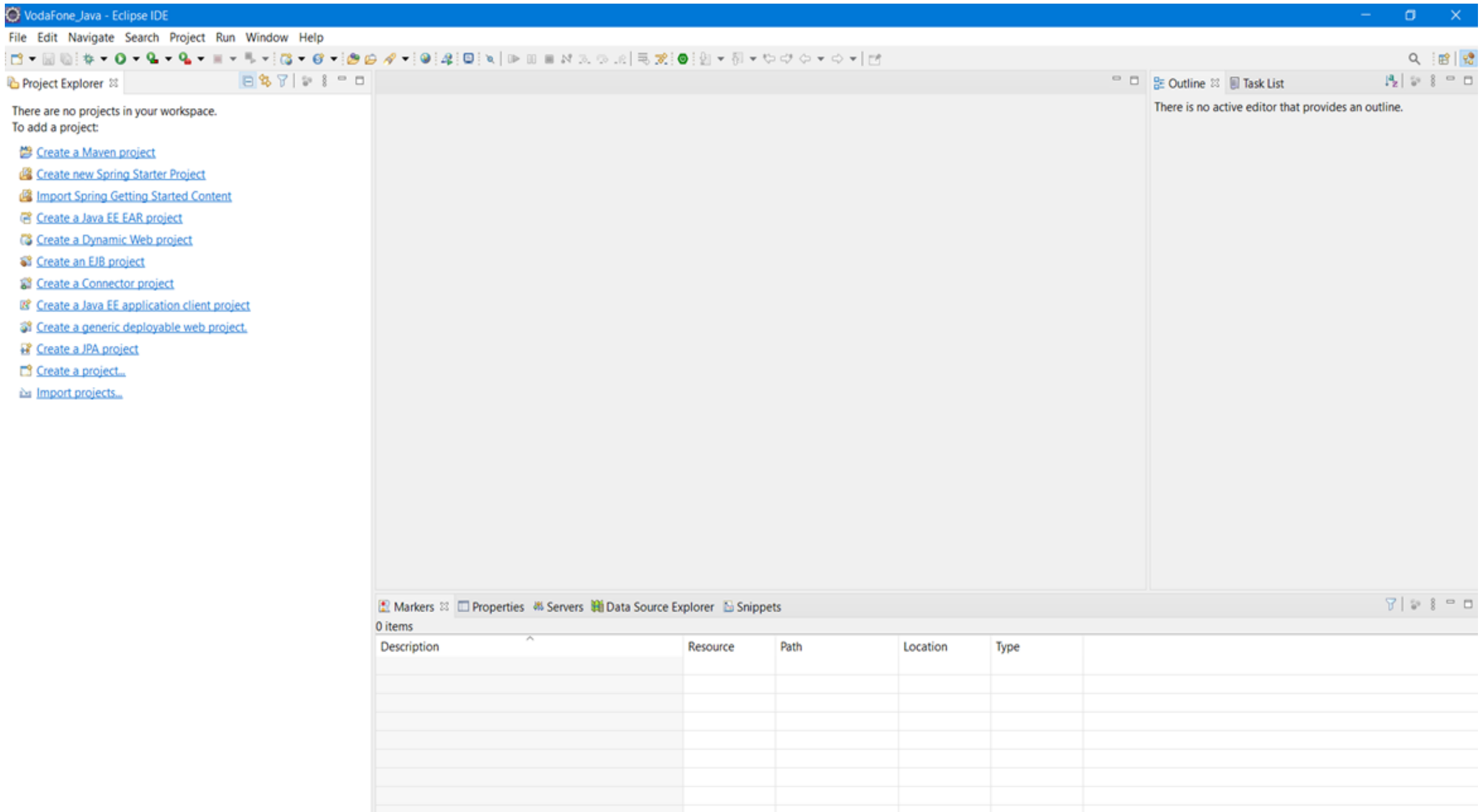
<https://www.eclipse.org/downloads/packages/>

<https://www.jetbrains.com/idea/download/#section=windows>



JAVA IS CASE SENSITIVE

CREATE WORKSPACE / JAVA PROJECT IN ECLIPSE



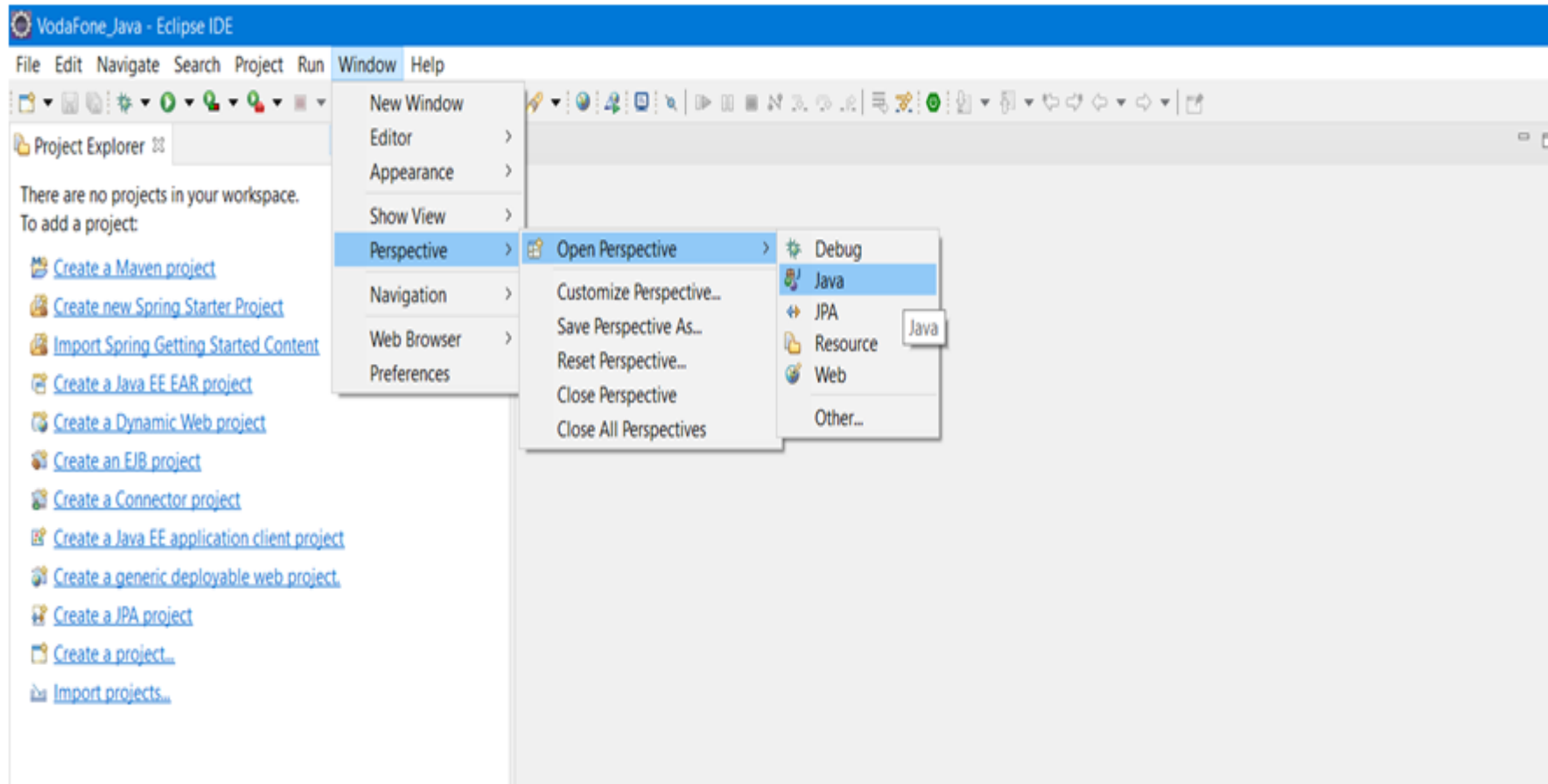
Description

Resource

Path

Location

Type





JAVA NAMING CONVENTION

Variable Name - Variable name should be in lowerCamelCase.

Method Name - Methods should be verbs and in lowerCamelCase.

Class Name - Class names should be nouns in UpperCamelCase.



WRITE AND RUN JAVA PROGRAM

A screenshot of the Eclipse IDE interface. The Package Explorer on the left shows a project named 'VodaFoneJava' with a source folder 'src' containing a file 'First_Prog.java'. The main editor window displays the code for 'First_Prog.java'. The code defines a public class 'First_Prog' with a public static void main method that prints 'WELCOME ALL' to the console. The console output at the bottom shows the program has terminated and printed 'WELCOME ALL'.

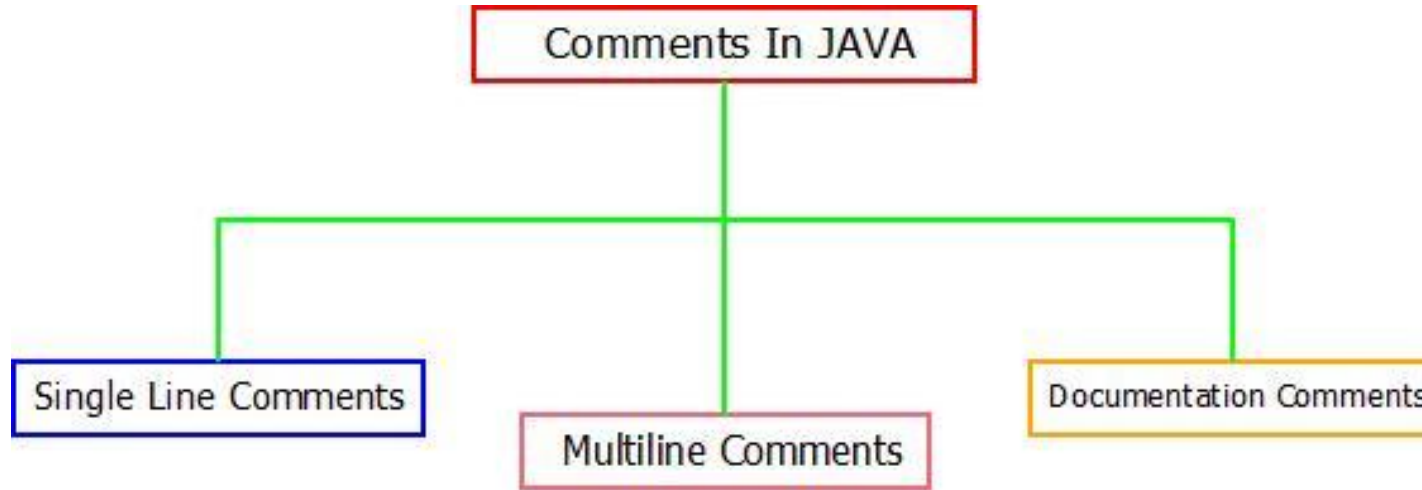
```
1 public class First_Prog {  
2  
3     public static void main(String[] args) {  
4  
5         System.out.println("WELCOME ALL");  
6  
7     }  
8  
9 }  
10
```

<terminated> First_Prog [Java Application] D:\eclipse_March21\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.wi

WELCOME ALL



Comments In Java

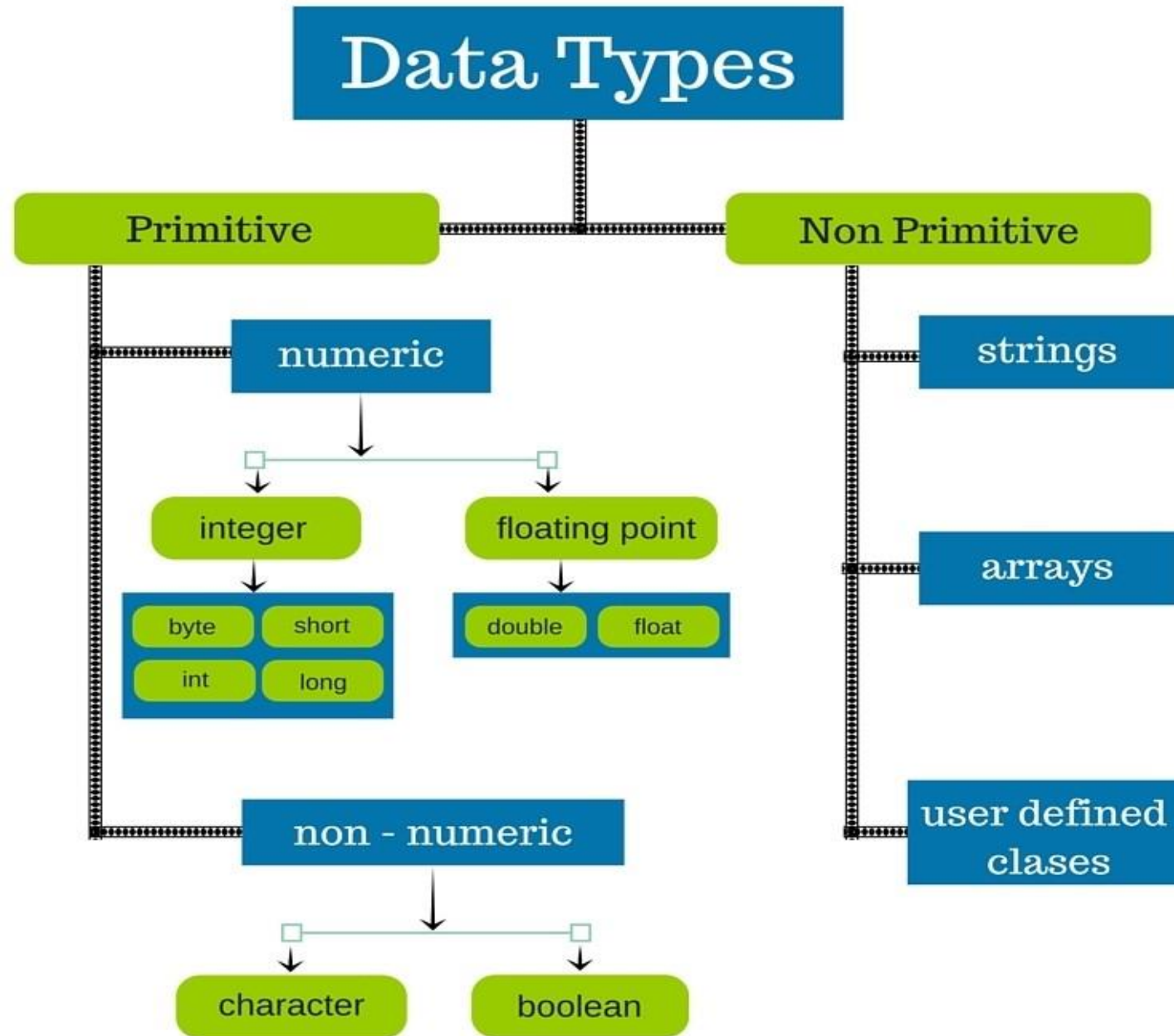


- **Comments in java are the statements that are ignore by java compiler and hence are not executed.**
- **They are used to describe the program, variable and method used to the other person who is reading the source code.**
- **Also, generation of documentation of a class using documentation comment is one of the feature of comments in java.**



*First_Prog.java

```
1 //SINGLE LINE COMMENT
2 //THIS IS MY FIRST PROGRAM
3 public class First_Prog {
4
5     public static void main(String[] args) {
6
7         /*
8          THIS IS
9          MULTILINE COMMENT
10        */
11
12        /**
13         This is Documentation
14         Comment
15        */
16        System.out.println("WELCOME ALL");
17    }
18 }
```



Java Primitive Data Types

Data Type	Characteristics	Range
<code>byte</code>	8 bit signed integer	-128 to 127
<code>short</code>	16 bit signed integer	-32768 to 32767
<code>int</code>	32 bit signed integer	-2,147,483,648 to 2,147,483,647
<code>long</code>	64 bit signed integer	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<code>float</code>	32 bit floating point number	$\pm 1.4\text{E-}45$ to $\pm 3.4028235\text{E+}38$
<code>double</code>	64 bit floating point number	$\pm 4.9\text{E-}324$ to $\pm 1.7976931348623157\text{E+}308$
<code>boolean</code>	<code>true</code> or <code>false</code>	NA, note Java booleans cannot be converted to or from other types
<code>char</code>	16 bit, Unicode	Unicode character, <code>\u0000</code> to <code>\uFFFF</code> Can mix with integer types

VodaFone_Java - VodaFoneJava/src/Second_AddTwoNo.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

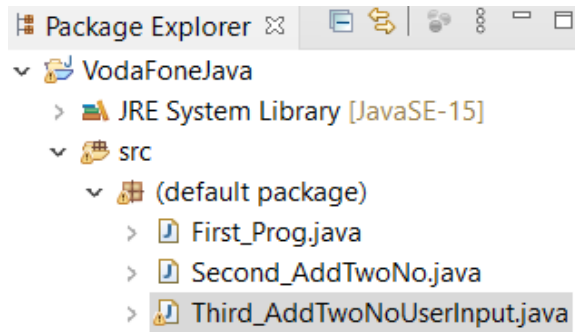
- VodaFoneJava
 - JRE System Library [JavaSE-15]
 - src
 - (default package)
 - First_Prog.java
 - Second_AddTwoNo.java
 - Third_AddTwoNoUserInput.java

```
1 public class Second_AddTwoNo {  
2  
3     public static void main(String[] args) {  
4  
5         double fno = 50.25;  
6  
7         int sno = 60;  
8  
9         System.out.println(fno+sno);  
10    }  
11 }  
12
```

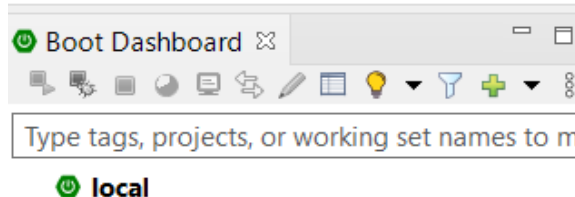
Problems @ Javadoc Declaration Console

<terminated> Second_AddTwoNo [Java Application] D:\eclipse_March21\plugins\org.eclipse.justj.openjdk.hotsp

110.25

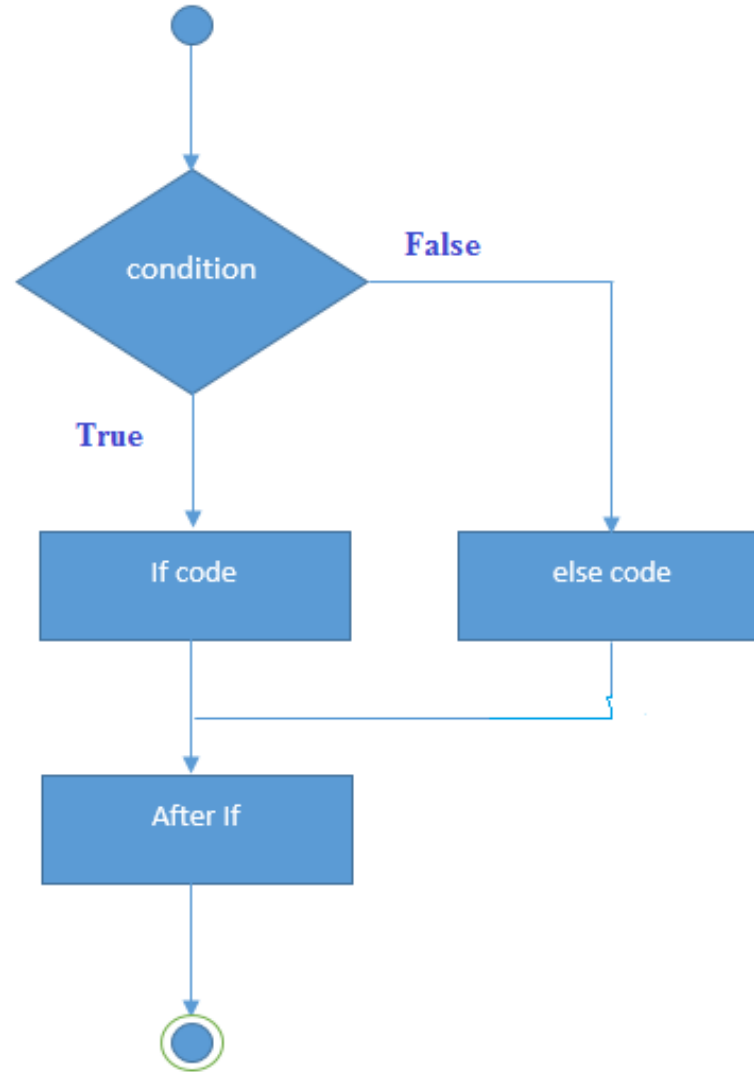


```
1 import java.util.Scanner;
2
3 public class Third_AddTwoNoUserInput {
4
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.println("Enter 1st no ");
9         double fno = sc.nextDouble();
10
11        System.out.println("Enter 2nd no ");
12        int sno = sc.nextInt();
13
14        System.out.println(fno+sno);
15    }
16 }
```



```
<terminated> Third_AddTwoNoUserInput [Java Application] D:\eclipse_March21\plugins\org.eclipse.justj.openj
Enter 1st no
12.35
Enter 2nd no
15
27.35
```

IF ELSE





*Four_If_Else_ValidVoter.java

```
1 import java.util.Scanner;
2
3 public class Four_If_Else_ValidVoter {
4
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         System.out.println("Enter Age ");
9         int age = sc.nextInt();
10
11         if (age >= 18)
12             System.out.println("Eligible To Give Vote");
13         else
14             System.out.println("Not Eligible To Give Vote");
15
16     }
17 }
```

Problems Javadoc Declaration Console

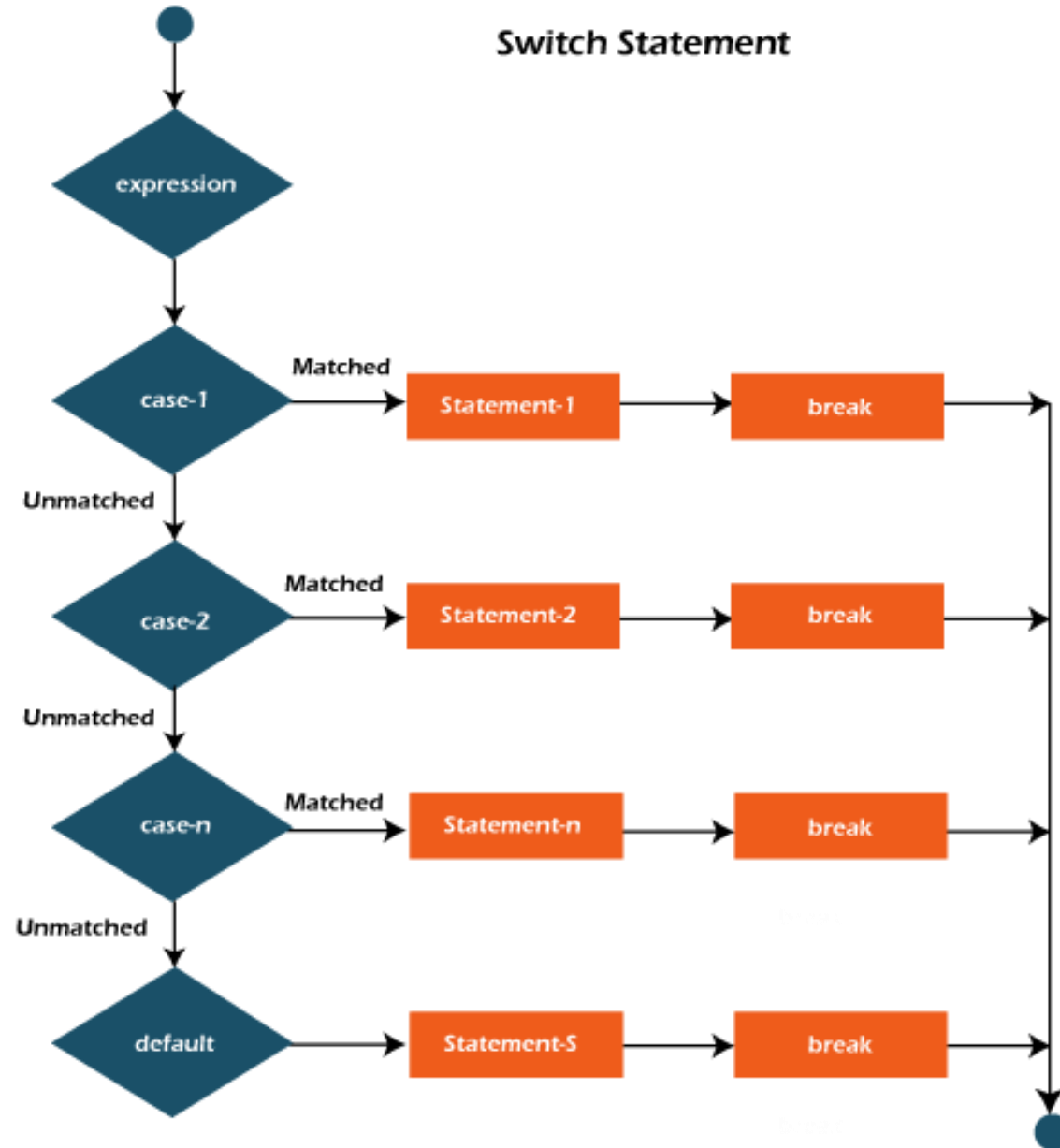
<terminated> Four_If_Else_ValidVoter [Java Application] D:\eclipse_March21\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v

Enter Age

12

Not Eligible To Give Vote

Switch Statement

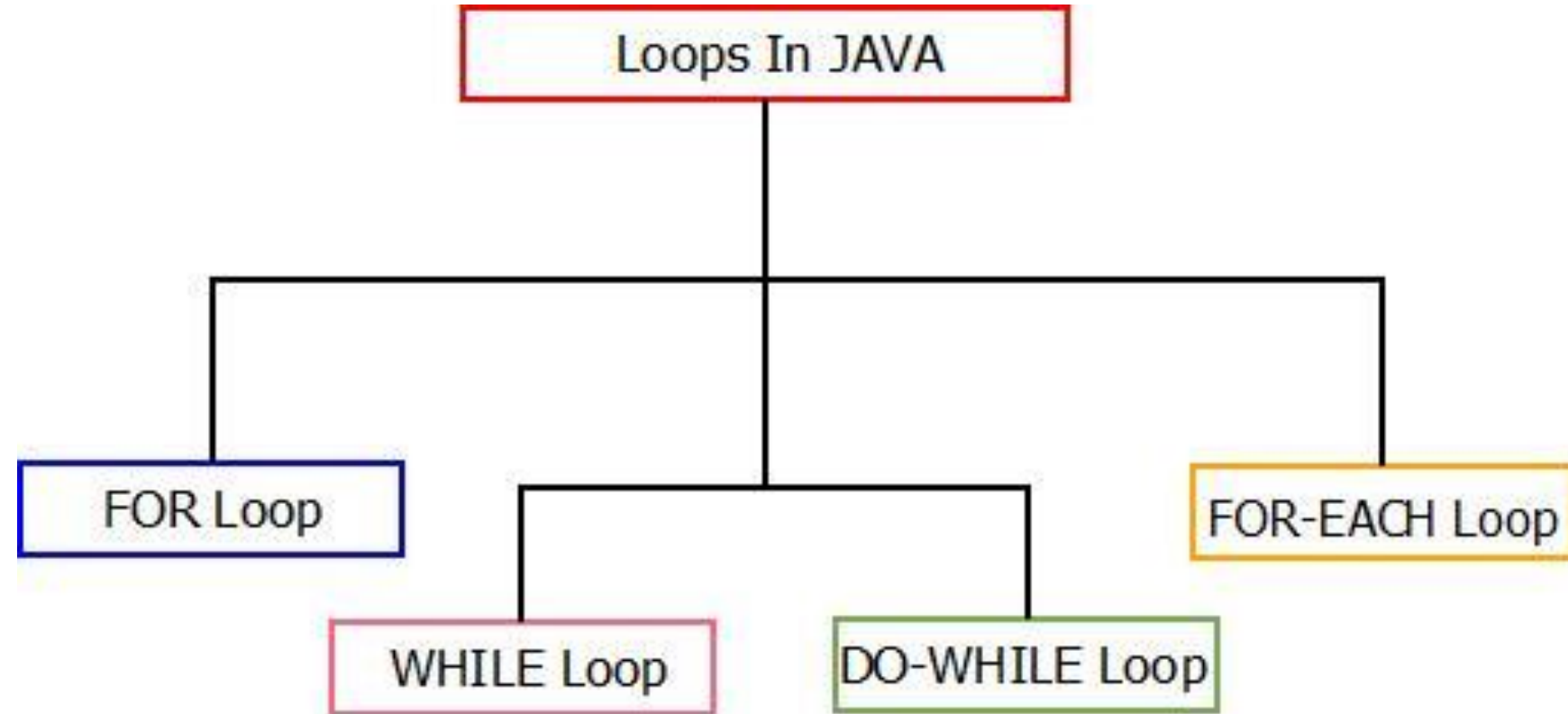


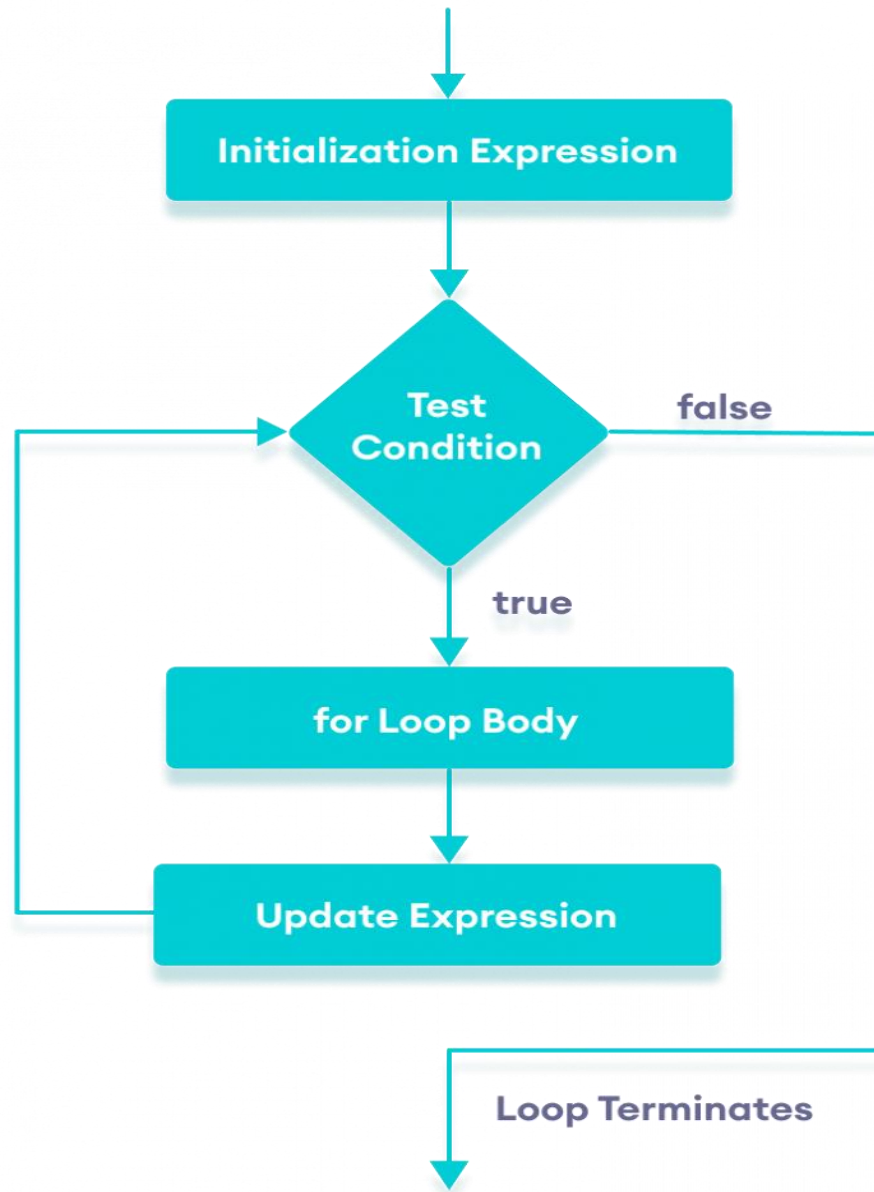
SWITCH CASE PROGRAM



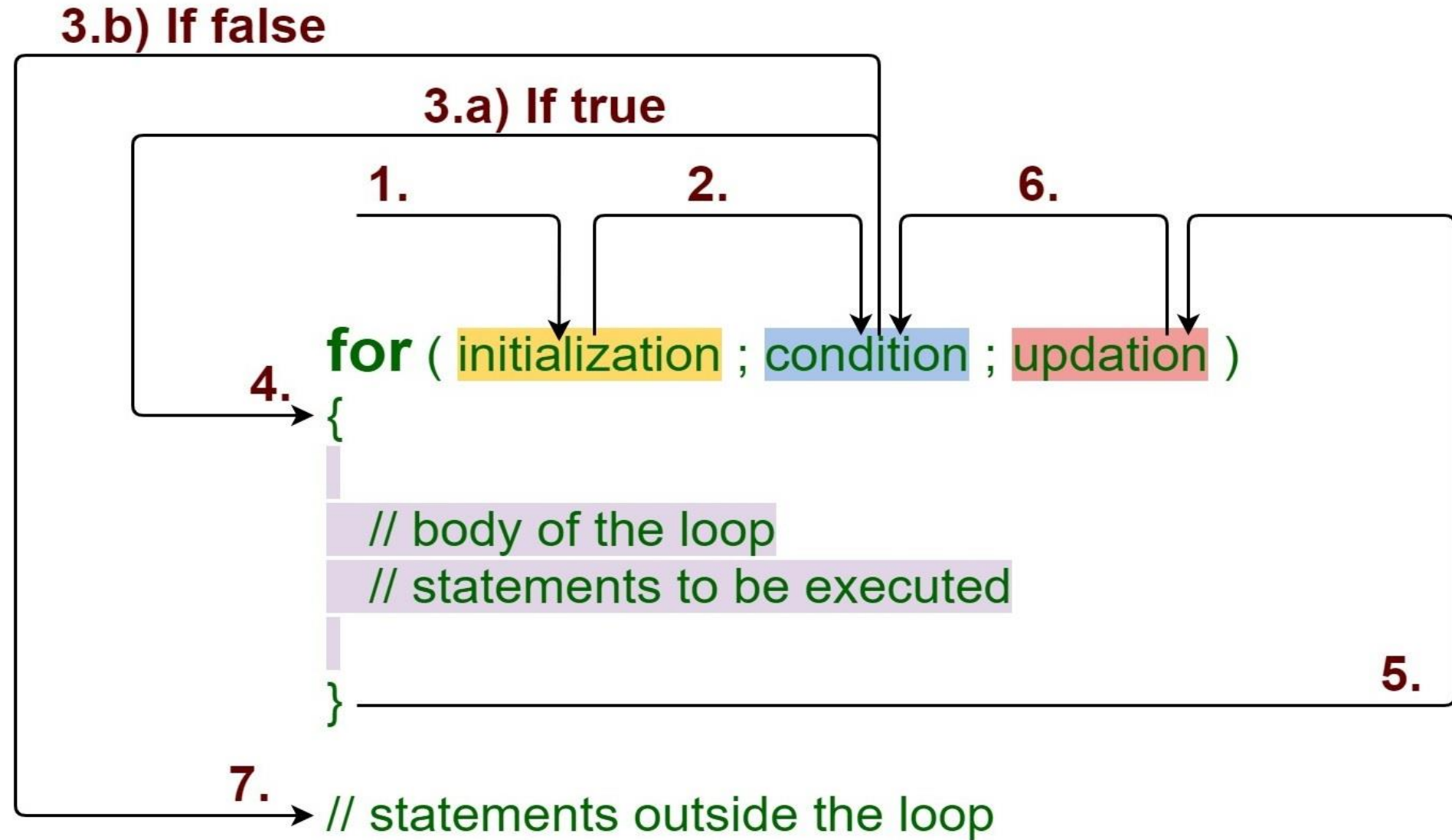
Operators	Notation	Precedence/Priority
Postfix	expr++ , expr--	1
Unary	++expr , --expr , +expr -expr , ~ , !	2
Multiplicative	* , / , %	3
Additive	+ , -	4
Shift	<< , >> , >>>	5
Relational	< , > , <= , >= , instanceof	6
Equality	== , !=	7
Bitwise AND	&	8
Bitwise Exclusive OR	^	9
Bitwise Inclusive OR		10
Logical AND	&&	11
Logical OR		12
Ternary	? :	13
Assignment	= , += , -= , *= , /= , %= , &= , ^= , = , <<= , >>= , >>>=	14

- Leap Year Problem
- Find greatest no among 3 no. No will give by the user.

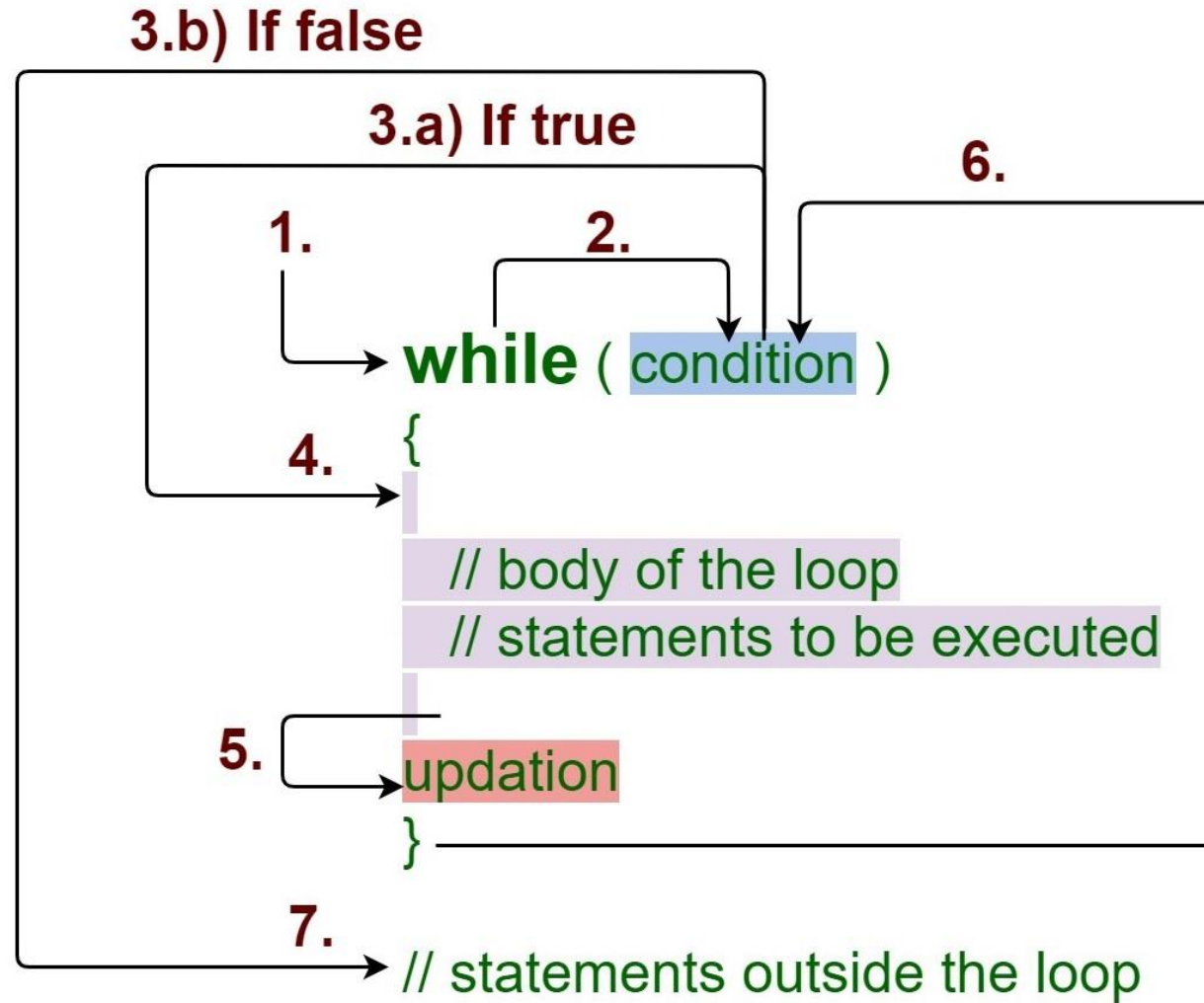




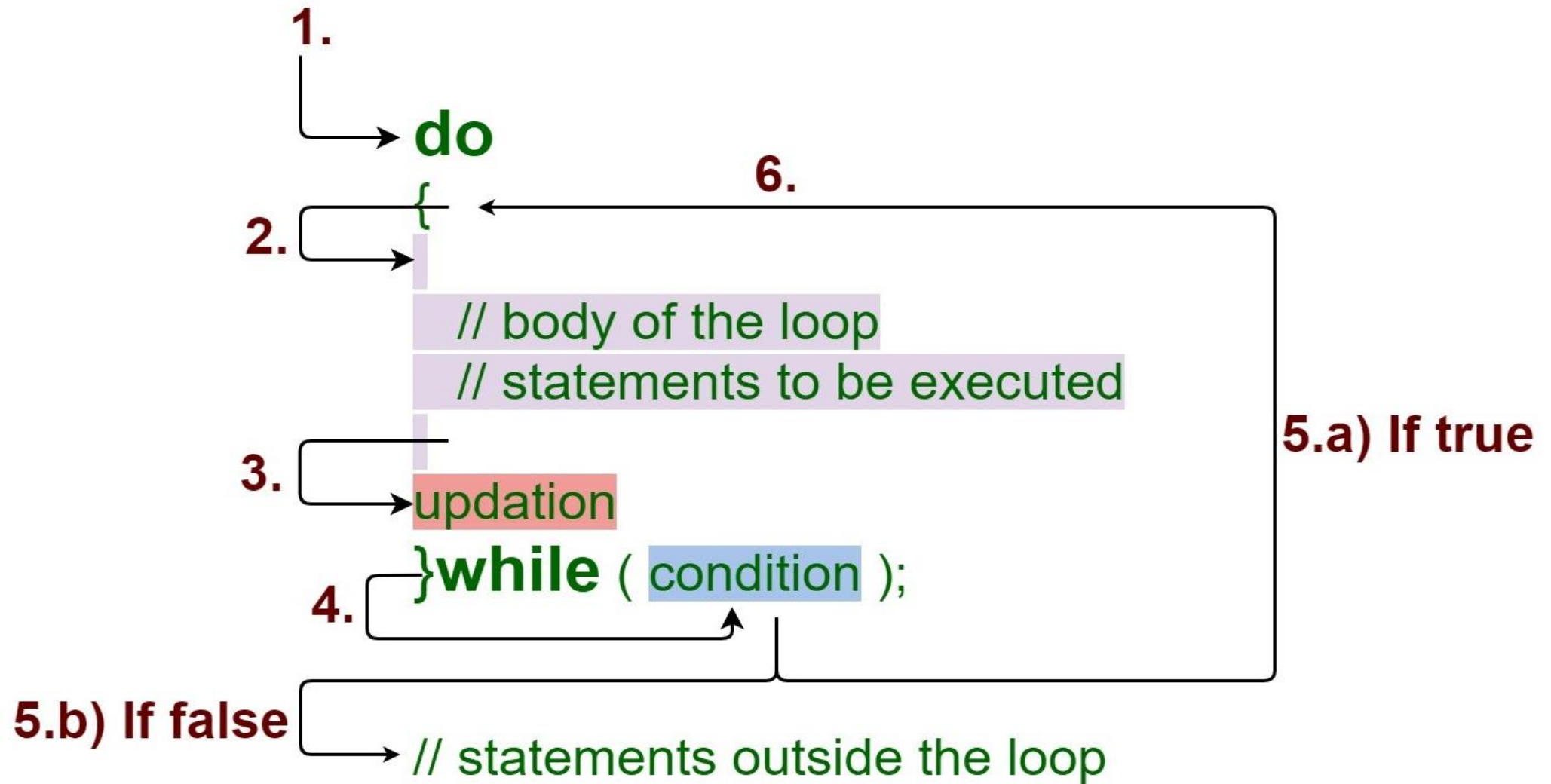
FOR LOOP – Entry Control Loop



WHILE LOOP - Entry Control Loop

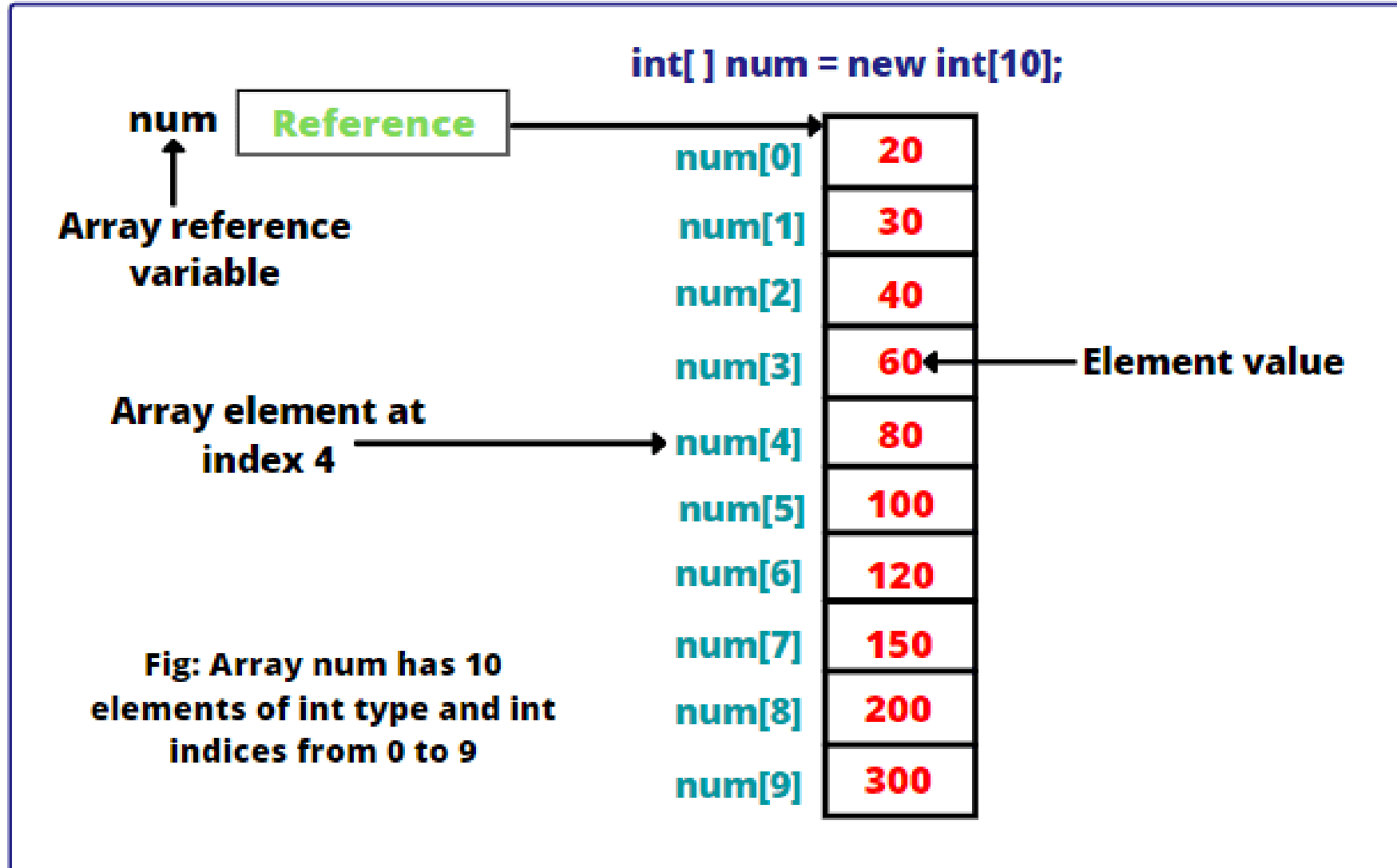


DO WHILE LOOP - Exit Control Loop



- PROGRAM
- Display All Odd No up to 20
- Check No is Prime Or Not

ARRAY (Single Dimensional Array)



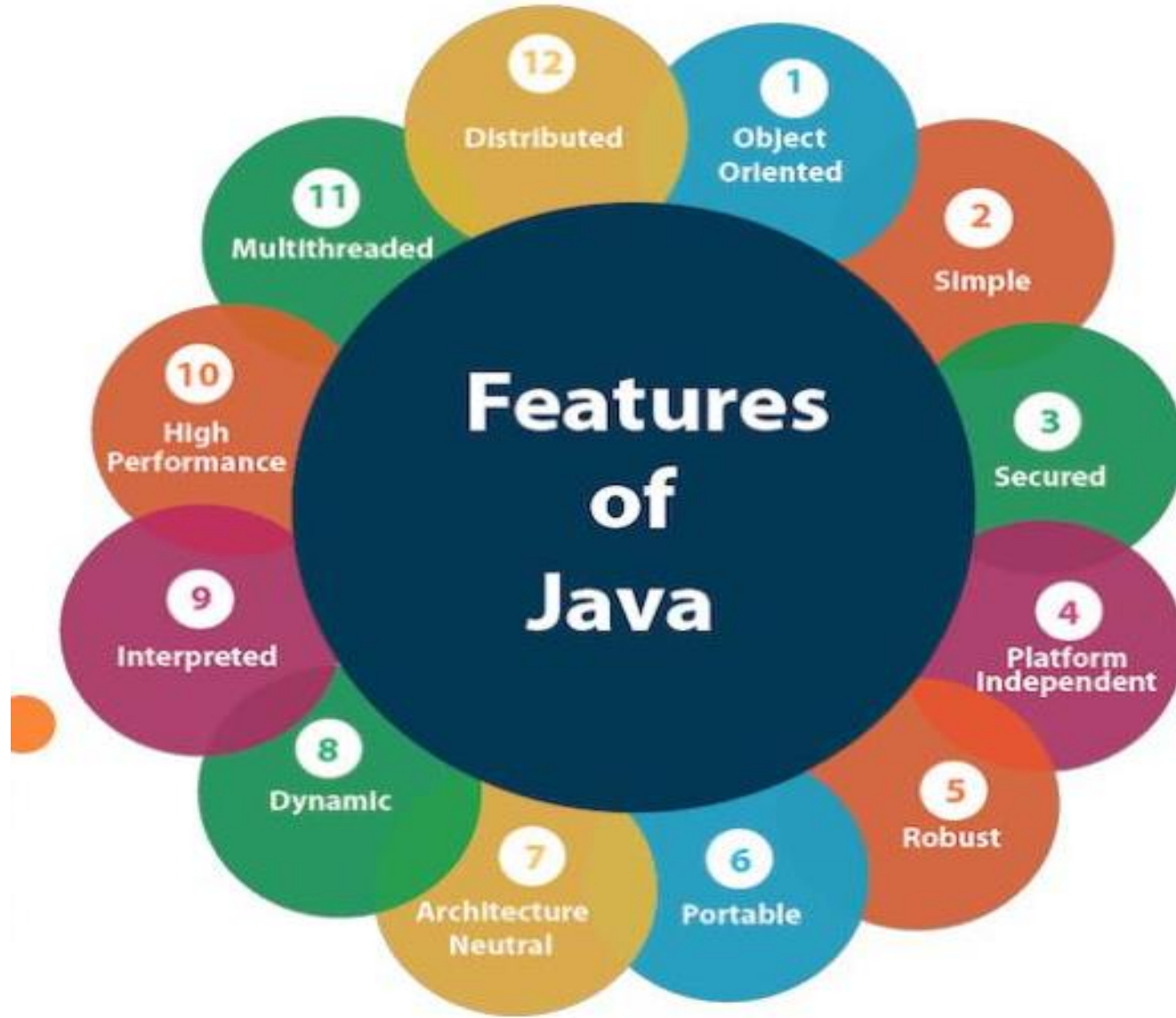
ARRAY (Multi-Dimensional Array)

	0	1	2
0	arr[0][0]	arr[0][1]	arr[0][2]
1	arr[1][0]	arr[1][1]	arr[1][2]
2	arr[2][0]	arr[2][1]	arr[2][2]

arr[3][3]

Multidimensional Array : Structure

- ARRAY PROGRAM INPUT/OUTPUT
- FIND GREATEST NO IN AN ARRAY
- SEARCH A NO IN AN ARRAY





Simple

- Java is easy to learn and its Syntax is quite simple, clean and easy to understand.
- The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

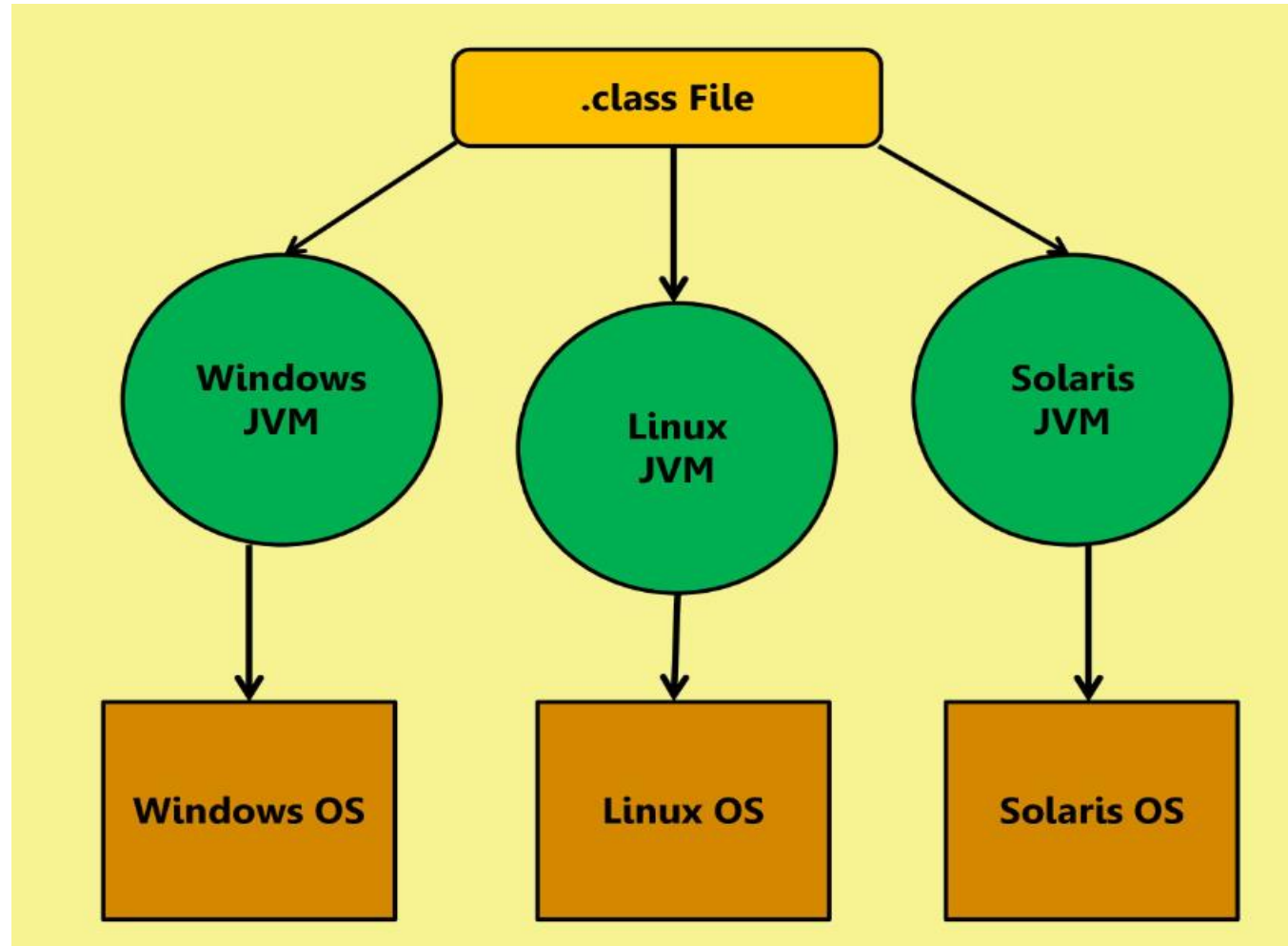


Secure

- When it comes to security, Java is always the first choice.
- With java secure features it enable us to develop virus free, temper free system.
- With the help of Byte Code Verifier Java achieve it
- Java program always runs in Java Runtime Environment(JRE) with almost null interaction with system OS, hence it is more secure.



Platform Independent





Robust

- Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking.
- But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling



Portable

- Java Byte code can be carried to any platform.
- No implementation dependent features.
- Everything Related to storage is predefined.
- For Example : size of primitive data types



Architectural Neutral

- Compiler generates bytecodes , which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.



Interpreted

- Bytecode can be interpreted on any platform by JVM.



High Performance

- Java is an interpreted language, so it will never be as fast as a compiled language like C or C++.
- But, Java achieve high performance with the use of just-in-time compiler (JIT)



Multi Threading

- Java multithreading feature makes it possible to write program that can do many tasks simultaneously.
- Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time.
- For Example - While we are typing, grammatical errors are checked.



Object Oriented Features

- Data Abstraction & Encapsulation
- Polymorphism
- Inheritance
- Modularity
- Dynamic Binding
- Message Passing
- Class
- Object