WEB DEVELOPER

@COMPUTER__PROGRAMMER

JAVA

CHEATSHEET 2024

1. BASIC SYNTAX

```java
// Single line comment
/* Multi-line comment */

public class Main {
  public static void main(String[] args) {
    System.out.println("Hello, World!"); // Print statement
  }
}
```

2. DATA TYPES

```java
int num = 100;            // Integer
double pi = 3.14;         // Double
char letter = 'A';        // Character
boolean isTrue = true;    // Boolean
String text = "Hello";    // String
```

3. VARIABLES AND CONSTANTS

```java
int x = 10;                  // Variable
final int MAX_VALUE = 100; // Constant
```

4. OPERATORS

```java
int sum = 5 + 10;          // Arithmetic
boolean isEqual = 5 == 5; // Equality
boolean isAnd = true && false; // Logical AND
int increment = ++x;       // Increment
```

5. CONDITIONAL STATEMENTS

```java
if (x > 10) {
  System.out.println("x is greater than 10");
} else if (x == 10) {
  System.out.println("x is 10");
} else {
  System.out.println("x is less than 10");
}
```

6. SWITCH STATEMENT

```
switch (day) {
  case 1:
    System.out.println("Monday");
    break;
  case 2:
    System.out.println("Tuesday");
    break;
  default:
    System.out.println("Another day");
}
```

7. LOOPS

```java
// For loop
for (int i = 0; i < 5; i++) {
  System.out.println(i);
}

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

8. ARRAYS

```java
int[] numbers = {1, 2, 3, 4, 5};
System.out.println(numbers[0]); // Access element

numbers[1] = 10; // Modify element
```

9. METHODS

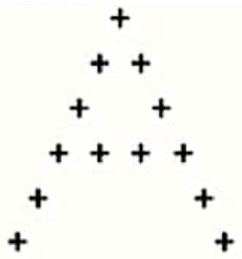
```java
public static int addNumbers(int a, int b) {
  return a + b;
}

public static void main(String[] args) {
  int result = addNumbers(5, 3);
  System.out.println(result);
}
```

10. CLASSES AND OBJECTS

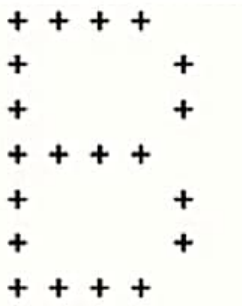
```java
public class Car {
  String model = "Tesla";

  public void displayModel() {
    System.out.println(model);
  }
}

public class Main {
  public static void main(String[] args) {
    Car myCar = new Car();  // Create an object
    myCar.displayModel();   // Call method
  }
}
```

Alphabet Patterns programs in java

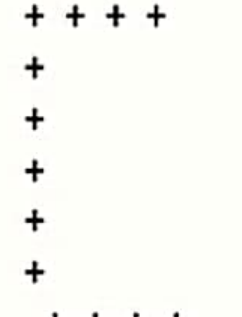
Pattern	Logic

	for (int i=1 ; i<=5 ; i++){
	for (int j=1 ; j<=9 ; j++){
	if((j==6-i)||(i==3&&j>=3&&j<=7)||(j==4+i))
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	}

Pattern	Logic

	for (int i=1 ; i<=7 ; i++){
	for (int j=1 ; j<=7 ; j++){
	if((j==1)||(i==1&&j<=6)||(i==4&&j<=6)||(i==7&&j<=6
	)||(j==7&&i>1&&i<4)||(j==7&&i>4&&i<7))
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	},

Pattern	Logic

	for (int i=1 ; i<=7 ; i++){
	for (int j=1 ; j<=7 ; j++){
	if((i==1&&j>1)||(j==1&&i>1&&i<7)||(i==7&&j>1))
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	}

Pattern	Logic

+ + + +
+ +
+ +
+ +
+ +
+ + + +
	for (int i=1 ; i<=7 ; i++){
	for (int j=1 ; j<=7 ; j++){
	if(j==1||(i==1&&j<7)||(i==7&&j<7)||(j==7&&i>1&&i<7))
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	}

Pattern	Logic

+ + + + +
+
+
+ + + + +
+
+
+ + + + +
	for (int i=1 ; i<=7 ; i++){
	for (int j=1 ; j<=7 ; j++){
	if(j==1||i==1||i==4||i==7)
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	}

Pattern	Logic

+ + + + +
+
+
+ + + + +
+
+
+
	for (int i=1 ; i<=7 ; i++){
	for (int j=1 ; j<=7 ; j++){
	if(j==1||i==1||i==4)
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();
	}

```
+ + + + +
+
+
+       + +
+         +
+         +
+ + + + +
```

```
for (int i=1 ; i<=7 ; i++){
        for (int j=1 ; j<=7 ; j++){
        if(j==1||i==1||i==7||(j==7&&i>=4)||(i==4&&j>=4))
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
}
```

Pattern	Logic

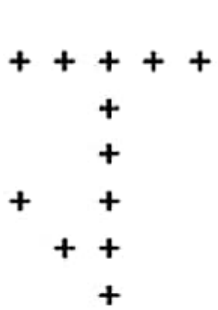
```
+         +
+         +
+         +
+ + + + +
+         +
+         +
+         +
```

```
for (int i=1 ; i<=7 ; i++){
        for (int j=1 ; j<=7 ; j++){
                if(j==1||i==4||j==7)
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
}
```

Pattern	Logic

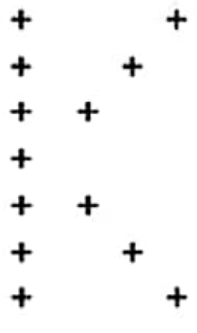
```
+ + + + +
    +
    +
    +
    +
    +
```

```
for (int i=1 ; i<=7 ; i++){
        for (int j=1 ; j<=7 ; j++){
                if(i==1||j==4||i==7)
                        System.out.print("+");
                else
                        System.out.print(" ");
```



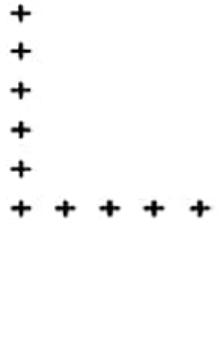
```
for (int i=1 ; i<=7 ; i++){
        for (int j=1 ; j<=7 ; j++){
                if(i==1||j==4||i==j+3)
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
}
```

Pattern	Logic



```
for (int i=1 ; i<=7 ; i++){
        for (int j=1 ; j<=7 ; j++){
                if(j==1||(j==6-i)||(i==2+j))
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
}
```

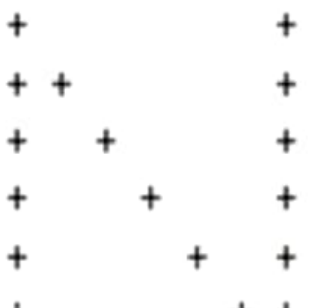
Pattern	Logic



```
for (int i=1 ; i<=7 ; i++)
{
        for (int j=1 ; j<=7 ; j++)
        {
                if(j==1||i==7)
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
```

```
for (int i=1 ; i<=7 ; i++)
{
        for (int j=1 ; j<=7 ; j++)
        {
                if(j==1||j==7||(i==j&&j<=4)||(j==8-i&&j>4))
                        System.out.print("+");
                else
                        System.out.print(" ");
        }
        System.out.println();
}
```

Pattern	Logic

	for (int i=1 ; i<=7 ; i++)
	{
	for (int j=1 ; j<=7 ; j++)
	{
	if(j==1||j==7||(i==j))
	System.out.print("+");
	else
	System.out.print(" ");
	}
	System.out.println();

WEB DEBVELOEPR
@COMPUTER__PROGRAMMER

OOPS in Real Life

@COMPUTER__PROGRAMMER

Polymorphism

A Girl

She can be many things

→ **Coder**

→ **Daughter**

→ **Mother**

Encapsulation

Company

It can have several departments

- → **Finance**
- → **Marketing**
- → **Production**

These Departments from the company

Inheritance

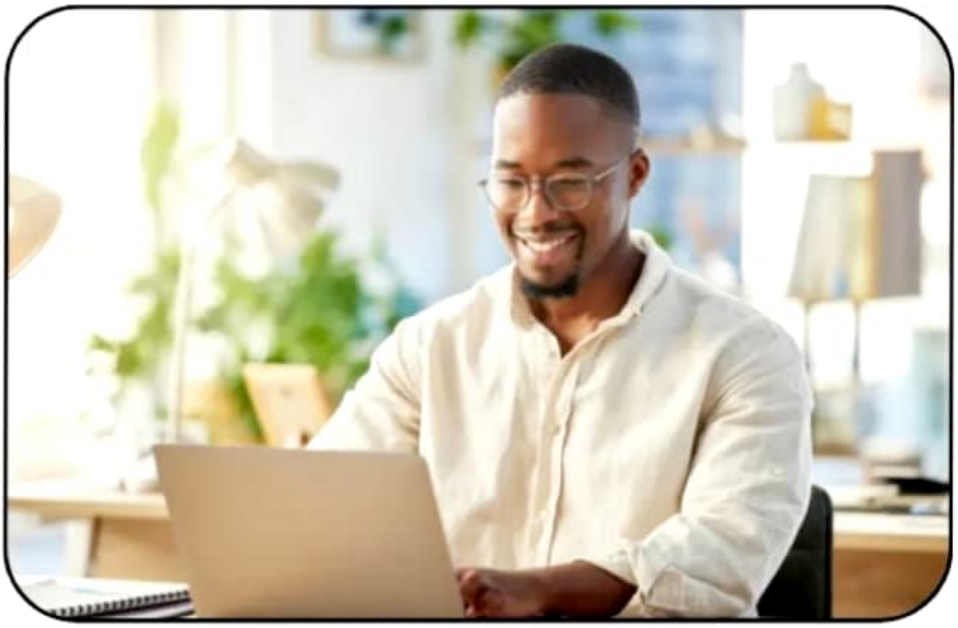
Dogs

They can have same

- → **Name**
- → **Size**
- → **Colour**

But, they are distinct dogs

Abstraction



Laptop

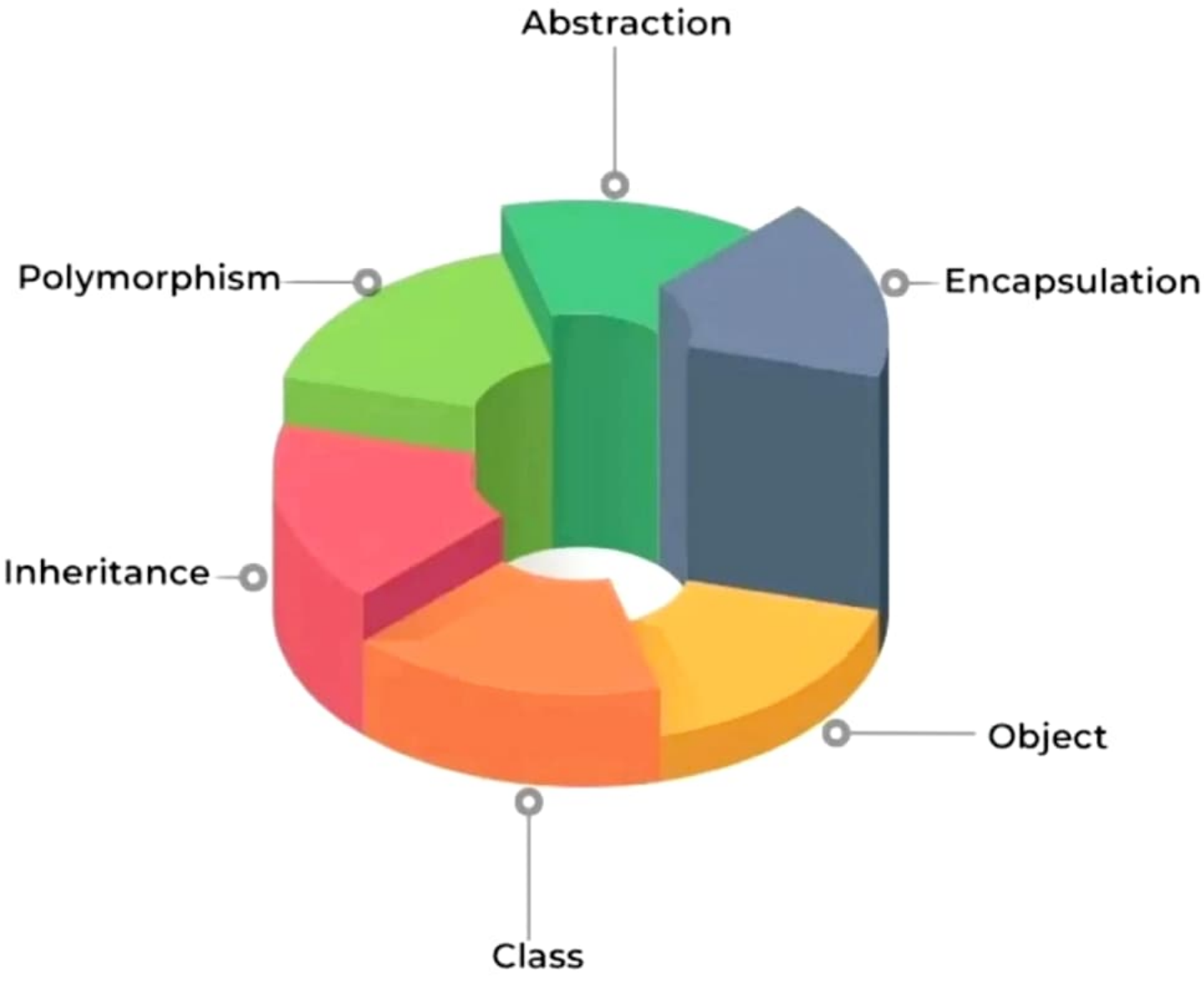
You can do make thing like

- → **Play Games**
- → **Coding**
- → **Designing**

It dosen’t revel internal process of how it functions

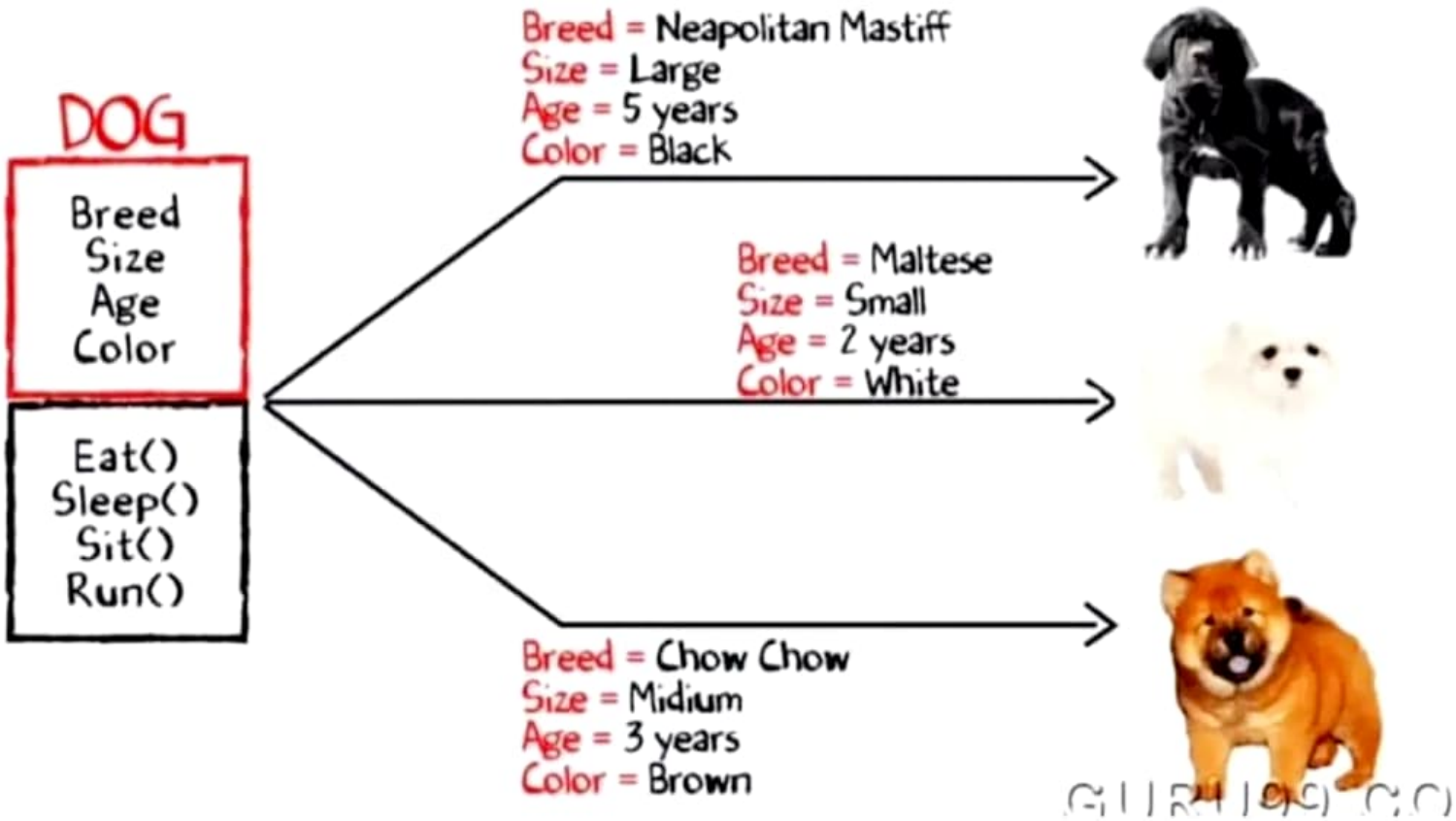
OOPS

in real life



OBJECT

An object is a class instance that allows programmers to use variables and methods from inside the class.



There are 4 OOP concepts. They are:

1. **Polymorphism**
2. **Inheritance**
3. **Encapsulation**
4. **Abstraction**

POLYMORPHISM

Polymorphism is the ability to exist in many forms.

A Player

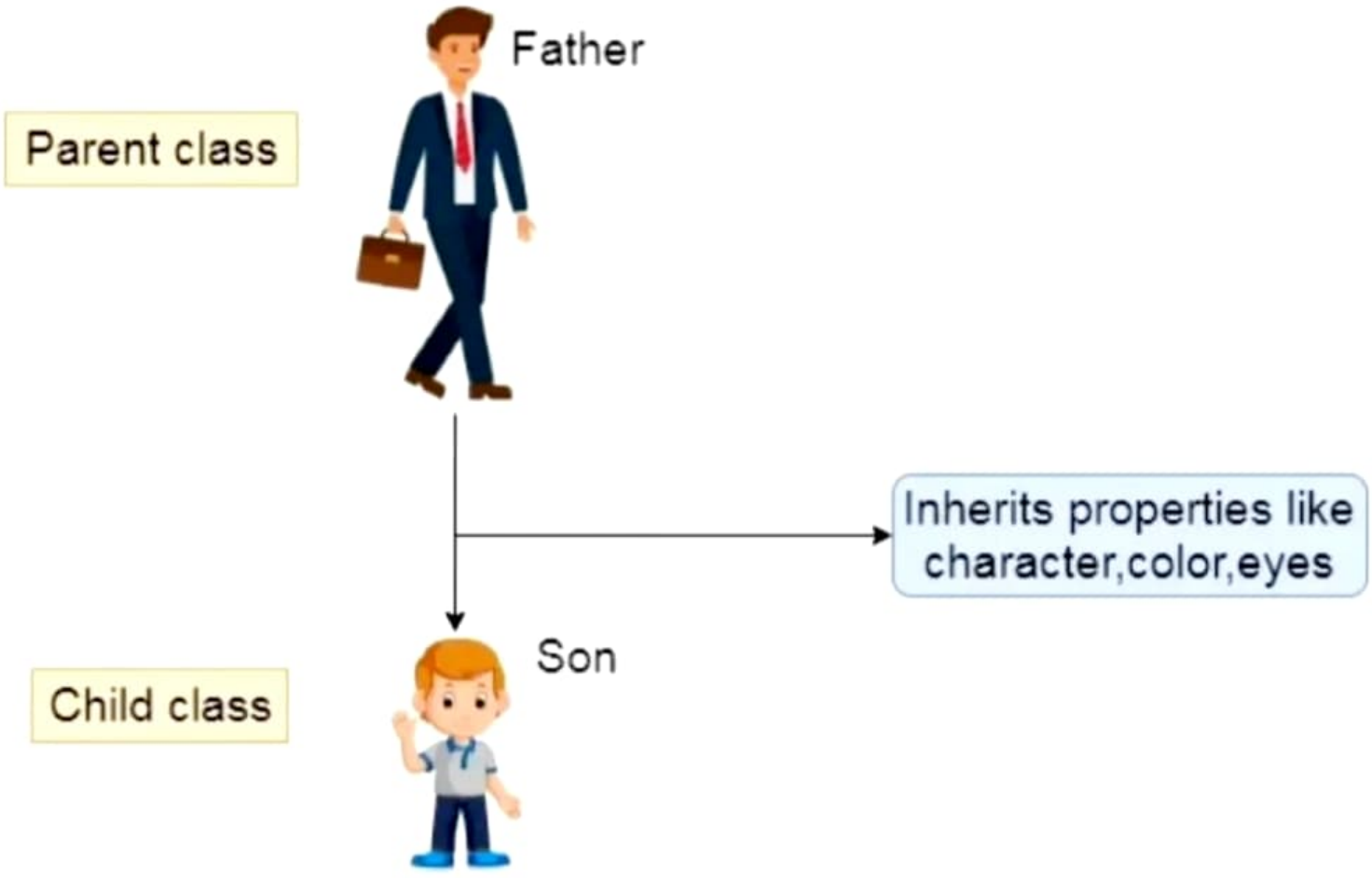
A Writer

A Student

A boy

INHERITANCE

Inheritance means it allows classes to inherit common properties from the parent class.



ENCAPSULATION

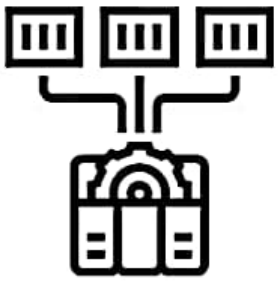
Encapsulation means it binds data and code together into one unit.

ABSTRACTION

In abstraction, it displays only the important information by hiding the implementation part.

DSA

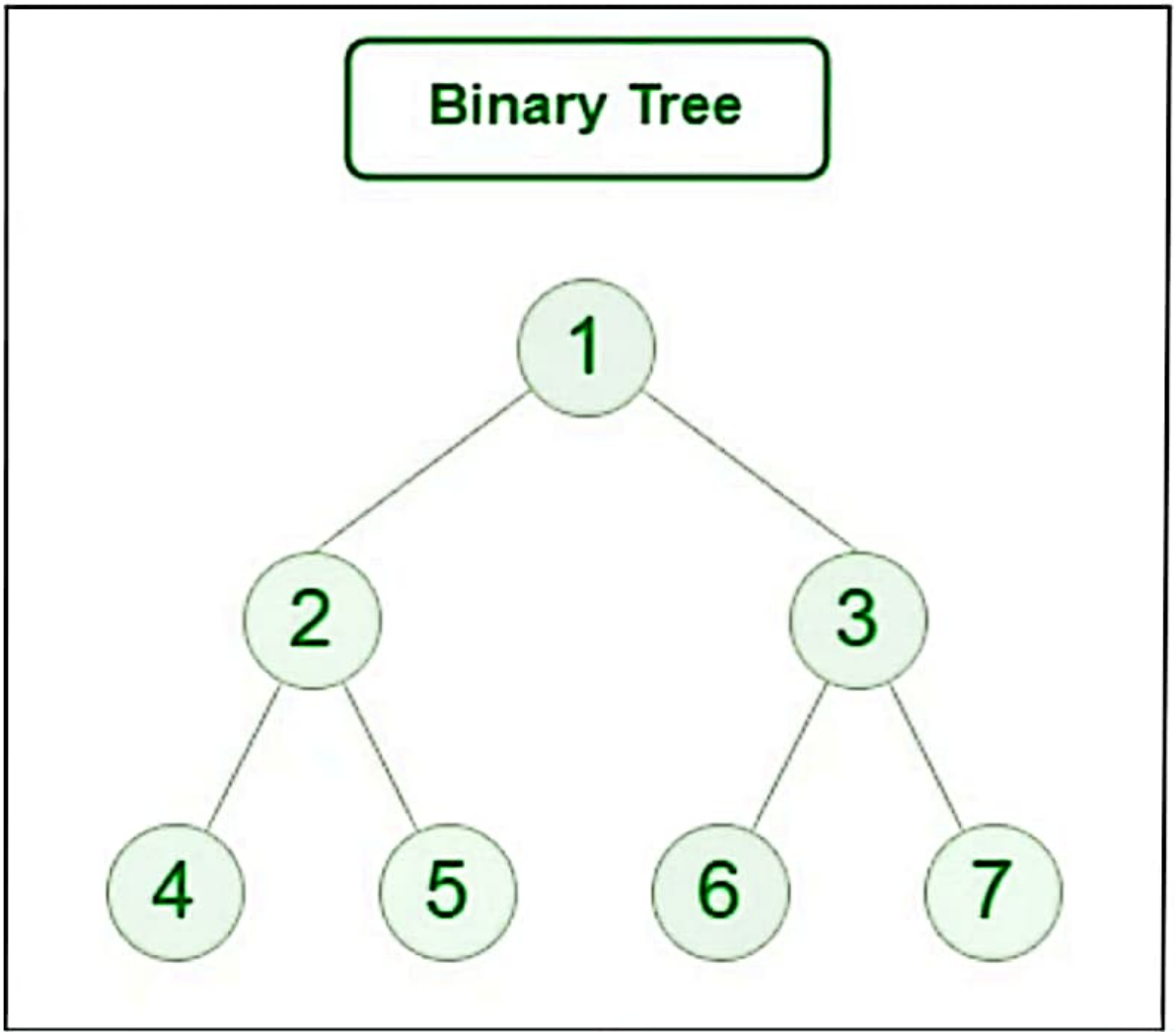
IN JAVA



@codi.vation

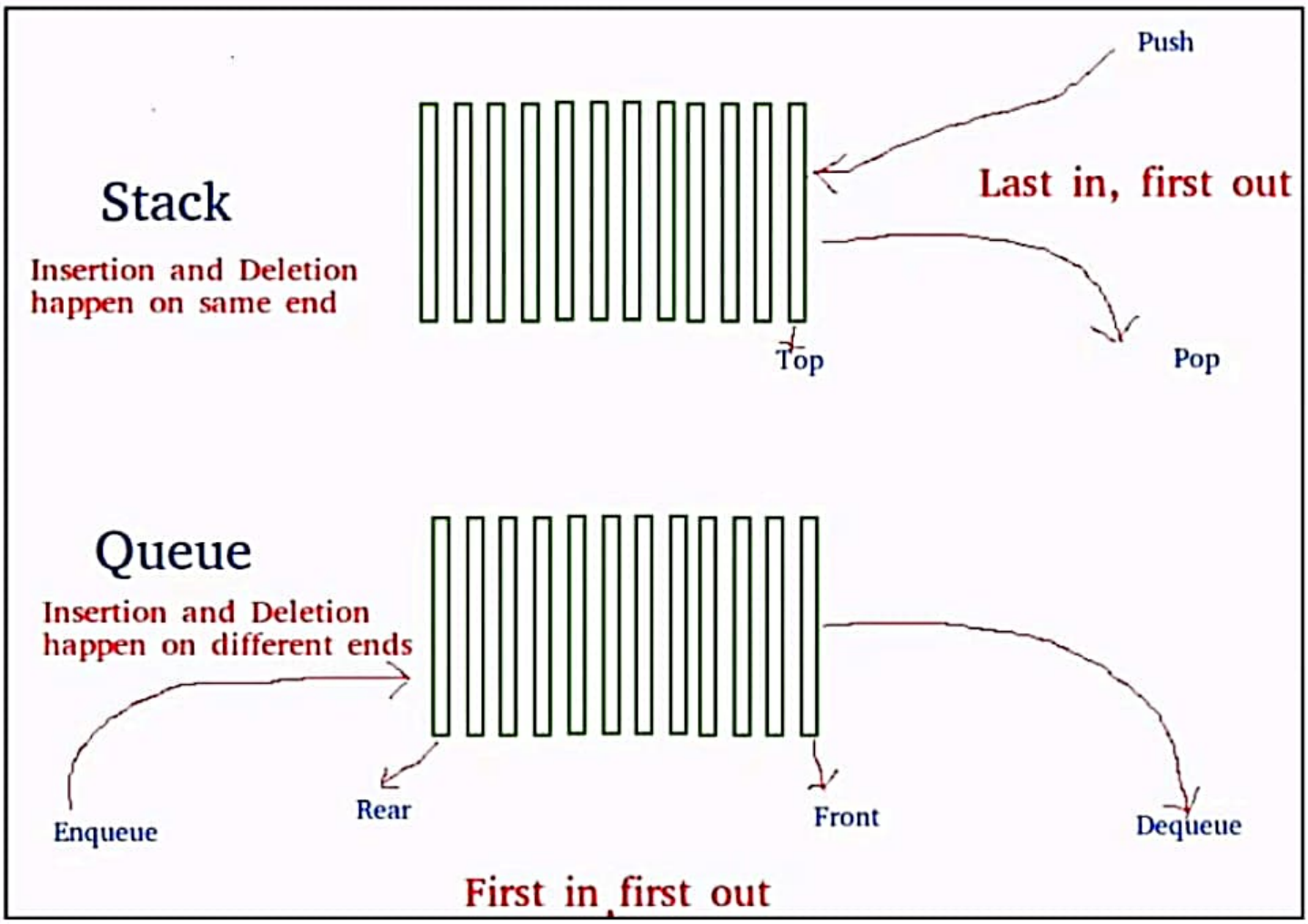
4. Trees:

- Binary trees, binary search trees (BST), AVL trees, and other tree data structures.
- Tree traversal algorithms (inorder, preorder, postorder).



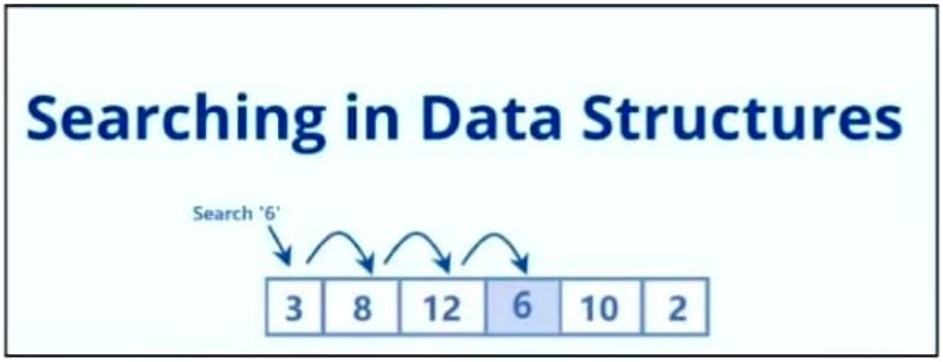
3. Stacks and Queues:

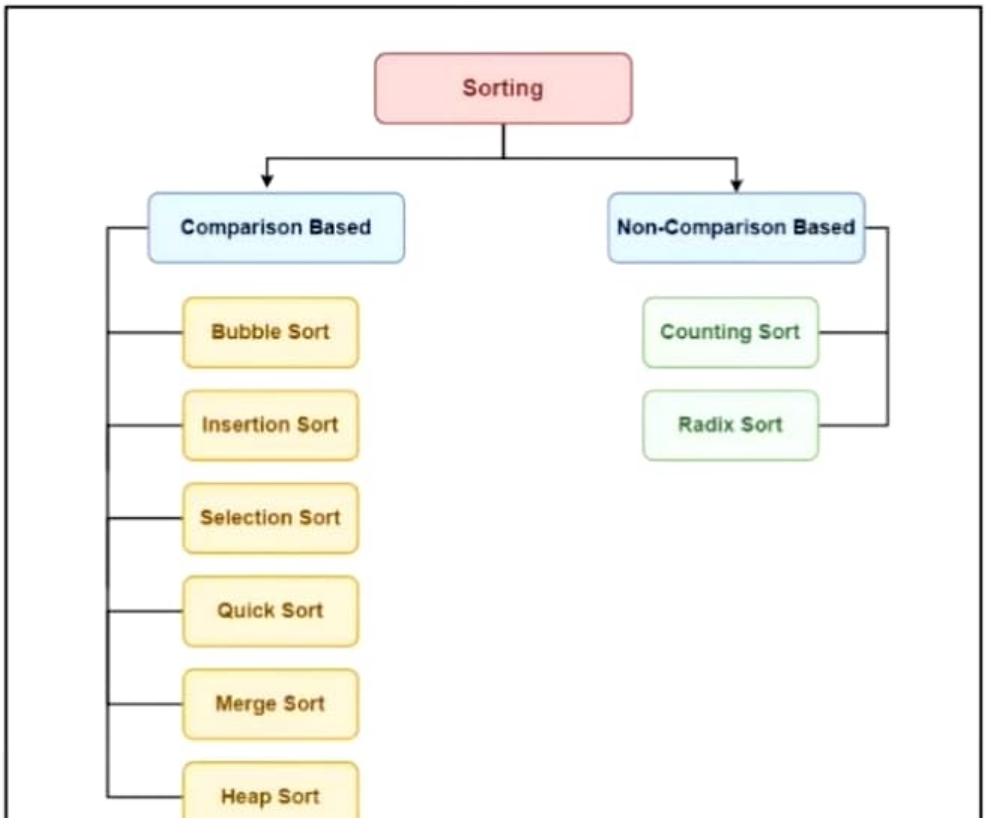
- Stack and queue implementations using arrays or linked lists.
- Operations like push, pop, enqueue, and dequeue.



6. Sorting and Searching:

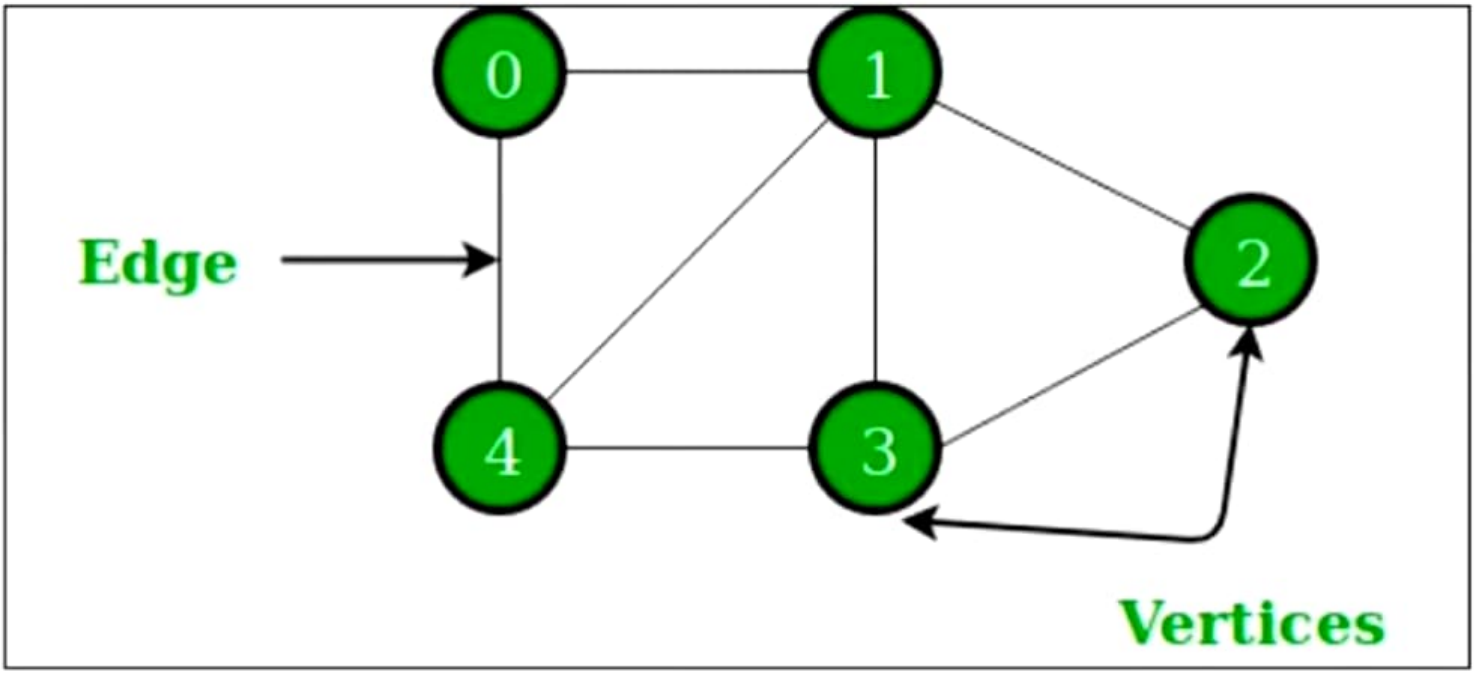
- Implementation of various sorting algorithms (e.g., quick sort, merge sort) and searching algorithms (e.g., binary search).





5. Graphs:

- Graph representation using adjacency matrix, adjacency list, or edge list.
- Graph traversal algorithms (DFS, BFS) and shortest path algorithms (Dijkstra's, Bellman-Ford).



Java was released in 1995

Python was released in 1991

Java programming

What is Java?

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by Sun Microsystems (which is now the subsidiary of Oracle) in the year 1995. James Gosling is known as the father of Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

Application

1. Desktop Applications such as acrobat reader, media player, antivirus, etc.
2. Web Applications such as irctc.co.in, javatpoint.com, etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games, etc.

Features of Java

Simple

Java is very easy to learn, and its syntax is simple, clean and easy to understand. According to Sun Microsystem, Java language is a simple programming language because:

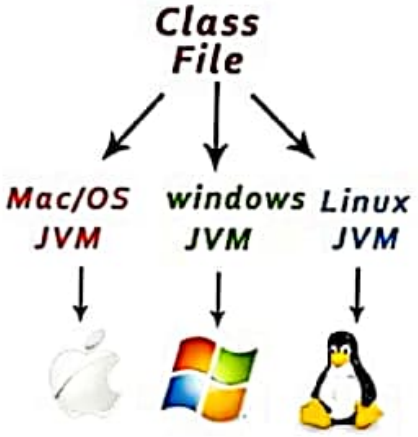
- Java syntax is based on C++ (so easier for programmers to learn it after C++).

- Java has removed many complicated and rarely-used features, for example, explicit pointers, operator overloading, etc.
- There is no need to remove unreferenced objects because there is an Automatic Garbage Collection in Java.

Object-oriented

Java is an object-oriented programming language. Everything in Java is an object. Object-oriented means we organize our software as a combination of different types of objects that incorporate both data and behaviour.

Platform Independent



Java is platform independent because it is different from other languages like C, C++, etc. which are compiled into platform specific machines while Java is a write once, run anywhere language. A platform is the hardware or software environment in which a program runs.

Secured

Java is best known for its security. With Java, we can develop virus-free systems. Java is secured because:

- **No explicit pointer**
- **Java Programs run inside a virtual machine sandbox**

Robust

The English mining of Robust is strong. Java is robust because:

- It uses strong memory management.
- There is a lack of pointers that avoids security problems.
- Java provides automatic garbage collection which runs on the Java Virtual Machine to get rid of objects which are not being used by a Java application anymore.

Architecture-natural

Java is architecture natural because there are no implementation dependent features, for example, the size of primitive types is fixed.

Portable

Java is portable because it facilitates you to carry the Java bytecode to any platform. It doesn't require any implementation.

High-performance

Java is faster than other traditional interpreted programming languages because Java bytecode is "close" to native code.

Distributed

Java is distributed because it facilitates users to create distributed applications in Java.

Multi-threaded

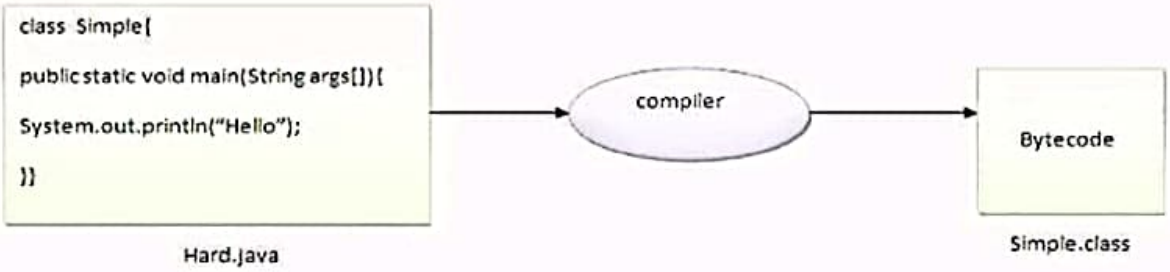
A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads.

Dynamic

Java is a dynamic language. It supports the dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.

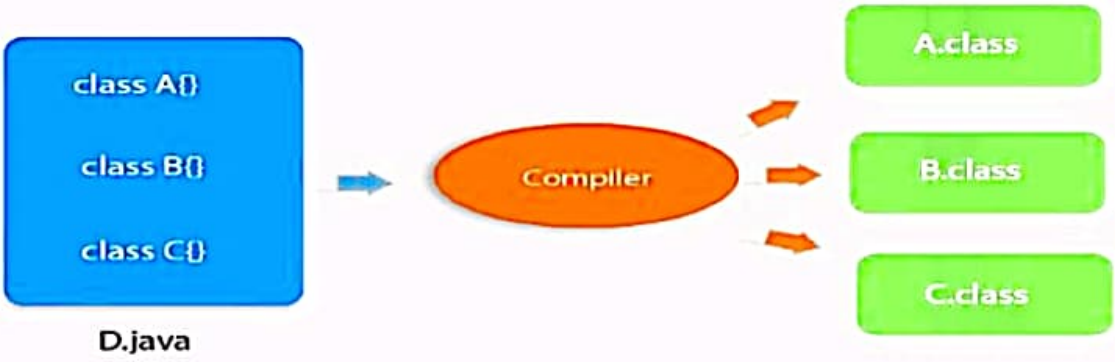
Q) Can you save a Java source file by another name than the class name?

Yes, if the class is not public. It is explained in the figure given below:



Q) Can you have multiple classes in a java source file?

Yes, like the figure given below illustrates:



Difference between JDK, JRE, and JVM

JVM

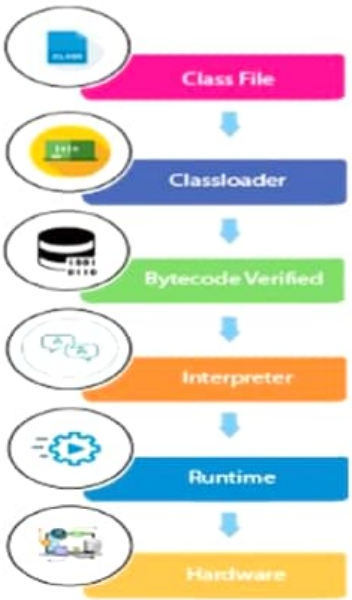
JVM (Java Virtual Machine) is an abstract machine. It is called a virtual machine because it doesn't physically exist. It is a specification that provides a runtime environment

First Java Program

- **class** keyword is used to declare a class in Java.
- **public** keyword is an access modifier that represents visibility. It means it is visible to all.
- **static** is a keyword. If we declare any method as static, it is known as the static method. The core advantage of the static method is that there is no need to create an object to invoke the static method. The main() method is executed by the JVM, so it doesn't require creating an object to invoke the main() method. So, it saves memory.
- **void** is the return type of the method. It means it doesn't return any value.
- **main** represents the starting point of the program.
- **String[] args** or **String args[]** is used for command line argument.
- **System.out.println()** is used to print statement. Here, System is a class, out is an object of the PrintStream class, println() is a method of the PrintStream class.

What happens at runtime?

At runtime, the following steps are performed:



Data Types in Java

Data types specify the different sizes and values that can be stored in the variable. There are two types of data types in Java:

1. **Primitive data types:** The primitive data types include boolean, char, byte, short, int, long, float and double.
2. **Non-primitive data types:** The non-primitive data types include Classes, Interfaces, and Arrays.

Unicode System

Unicode is a universal international standard character encoding that is capable of representing most of the world's written languages.

Why java uses Unicode System?

Before Unicode, there were many language standards:

- **ASCII** (American Standard Code for Information Interchange) for the United States.
- **ISO 8859-1** for Western European Language.
- **KOI-8** for Russian.
- **GB18030 and BIG-5** for Chinese, and so on.

Problem

This caused two problems:

1. A particular code value corresponds to different letters in the various language standards.
2. The encodings for languages with large character sets have variable length.
3. Some common characters are encoded as single bytes, other require two or more byte.

JRE

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment.

JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

Java Variables

A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type.

Variable is a name of memory location. There are three types of variables in java: local, instance and static.

1) Local Variable

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

A local variable cannot be defined with "static" keyword.

2) Instance Variable

A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.

3) Static variable

A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class.

Solution

To solve these problems, a new language standard was developed i.e. Unicode System.

In unicode, character holds 2 byte, so java also uses 2 byte for characters.

lowest value:\u0000

highest value:\uFFFF

Operators in Java

Operator in Java is a symbol that is used to perform operations. For example: +, -, *, / etc.

There are many types of operators in Java which are given below:

- Unary Operator,
- Arithmetic Operator,
- Shift Operator,
- Relational Operator,
- Bitwise Operator,
- Logical Operator,
- Ternary Operator and
- Assignment Operator.

Java Keywords

Java keywords are also known as reserved words. Keywords are particular words that act as a key to a code. These are predefined words by Java so they cannot be used as a variable or object name or class name.