# Java Fullstack

## What is Full stack

- Frontend
- Backend
- Database

End to end application development process is called as Full stack Development.

## What is Software Project?

=> Collection of programs is called as Software Project.

=> Software projects are used to reduce human efforts

=> Software projects are simplifying human's life

Ex: online banking, tickets booking, shopping etc...

## Software Companies

1) Product Based
2) Service Based
3) Outsourcing

=> Product Based Companies will develop projects/products and will sell those projects in market.

Ex: Microsoft, amazon, flipkart, samsung, sony, ibm, oracle., google.

=> Service Based Companies will develop the projects based on client requirement.

Ex: TCS, Infy, Capgemni, CTS, Accenture, Wipro, TechM, Deloitte....

=> Outsourcing companies will provide employees to other companies on contract basis.

Got Job in HCL company

HCl provided 7 lakhs per year (7 LPA)

After 2 months, HCL sent that person to Microsoft

## Product Based Company

Package : your.exp * 5 to 7 lakhs

1) Data Structures & Algorithms

2) Problem solving

3) System design

4) Design patterns

## Service Based Companies

Package : years.of.exp * 3 to 4 lakhs

1) Coding

2) Fullstack development

- Frontend Technologies (HTML, CSS, JS, BS, Ng or React)
- Backend Technologies(Java/Python/Dot Net/ Node JS)
- Database ( Oracle / MySQL )

## Fullstack course

Course Duration : 6 to 7 Months

Daily Timings: 9 AM to 6 PM

Laptop is mandatory

Monday to friday regular classes will happen

every saturday exam will be there

Sunday is holiday

## Course Content

Module-1 : Backend Development (Programming Language)

Java : Core java + Adv Java + 100 Logical Programs

Python : Core Python + Adv Python + 100 Logical Programs

Module-2 : Web Development

Ui : HTML + CSS + JS + BS + Angular

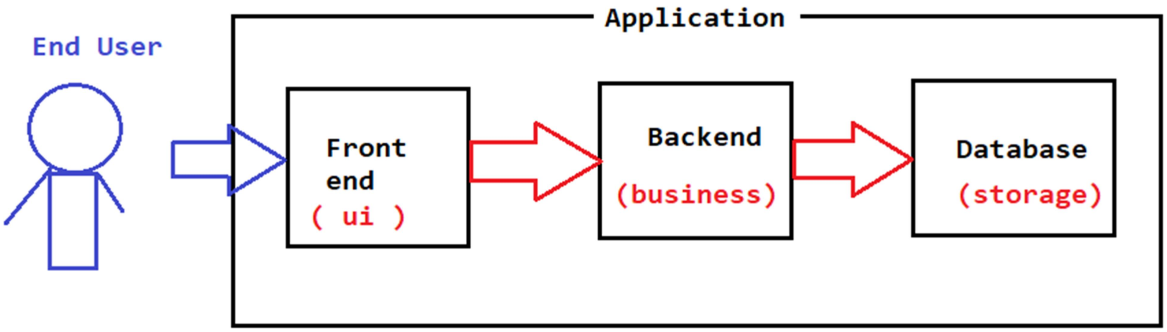
Module-3 : Database

DB : SQL & PL-SQL

Module-4 : Frameworks

Java : Spring Boot & Microservices + Realtime Project

Python : Django + Realtime Project



**What is Database?**

=> Database is used to store the data.

=> We will perform CRUD Operations in Database

C : Create

R : Retrieve

U : Update

D : Delete

Ex: Oracle DB, MySQL DB, SQL Server, Mongo DB etc...

**What is Backend?**

=> Backend contains business logic

Ex:

- Verify your login credentials are valid or not
- User account creation
- Sending emails
- Sending OTPs
- Calculations

**Backend Technologies:**

- Java
- .Net
- Python
- PHP
- Node JS

**What is Frontend?**

=> User Interface

=> It contains presentation logic

=> End Users will communicate with application using frontend only

**Frontend Technologies:**

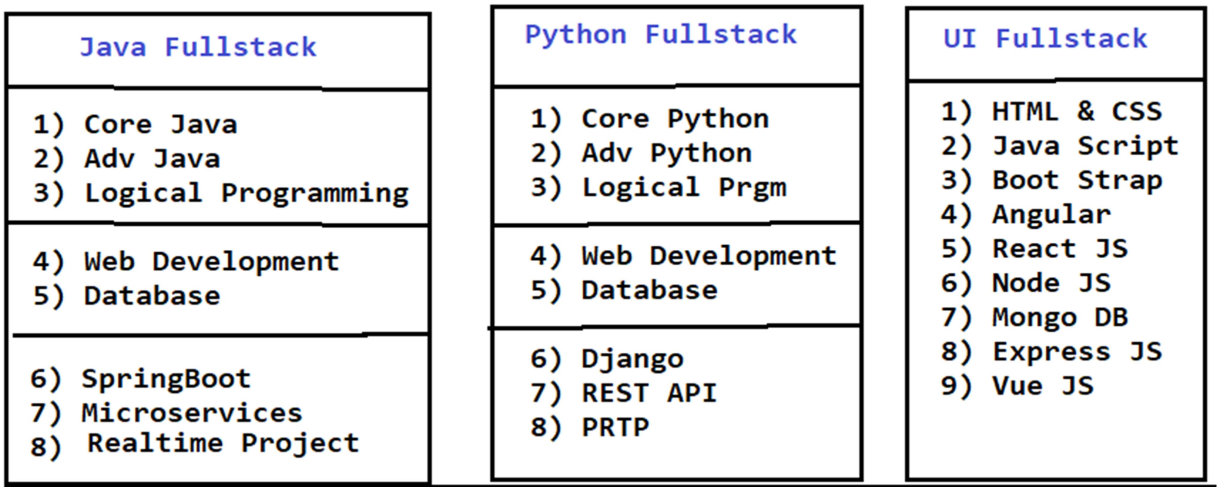
1) **HTML:** Structure of web page

2) **CSS:** To apply styles for web page

3) **Java Script:** To add dynamic behaviour

4) **Bootstrap:** Responsive design

5) **Angular / React JS:** To develop enterprise apps



Java Fullstack

1) Core Java
2) Adv Java
3) Logical Programming

4) Web Development
5) Database

6) SpringBoot
7) Microservices
8) Realtime Project

Python Fullstack

1) Core Python
2) Adv Python
3) Logical Prgm

4) Web Development
5) Database

6) Django
7) REST API
8) PRTP

UI Fullstack

1) HTML & CSS
2) Java Script
3) Boot Strap
4) Angular
5) React JS
6) Node JS
7) Mongo DB
8) Express JS
9) Vue JS

## Software Development Life Cycle (SDLC)

=> SDLC Represents end to end application development process

=> We will have several phases in SDLC like below

1) **Requirements Gathering ->** **(Functional Team / SME) -> FDD / SRS**
2) **Analysis ->** **Understand requirements & ask questions**
3) **Planning ->** **Prepare plan & prototype**
4) **Development ->** **Coding (frontend & backend & database)**
5) **Testing ->** **Verification and validation**
6) **Deployment ->** **Run application in the server**
7) **Delivery ->** **Handover to client (DevOps)**
8) **Maintenance ->** **Based on SLA ( 6 Months / 1 Year)**

## Waterfall Model

-> Sequence / Linear Model

-> Step by Step process

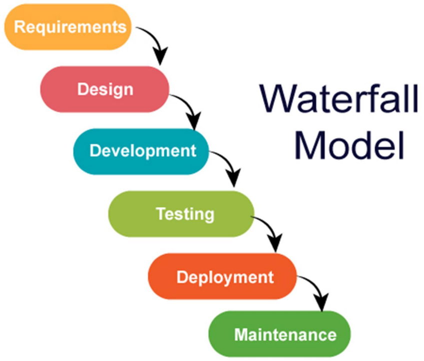
-> We will move in forward direction

-> Requirements are fixed

-> Budget is fixed

-> Client involvement is very less

-> Client will see project only at the end



## What is Agile?

-> Agile is iterative approach / Cyclic process

-> In Agile methodology "Planning + Development + Testing + Deployment + Delivery" is continuous process

-> Requirements are not fixed

-> Budget is not fixed

-> In Agile, we will deliver project to client in multiple releases (Sprints).

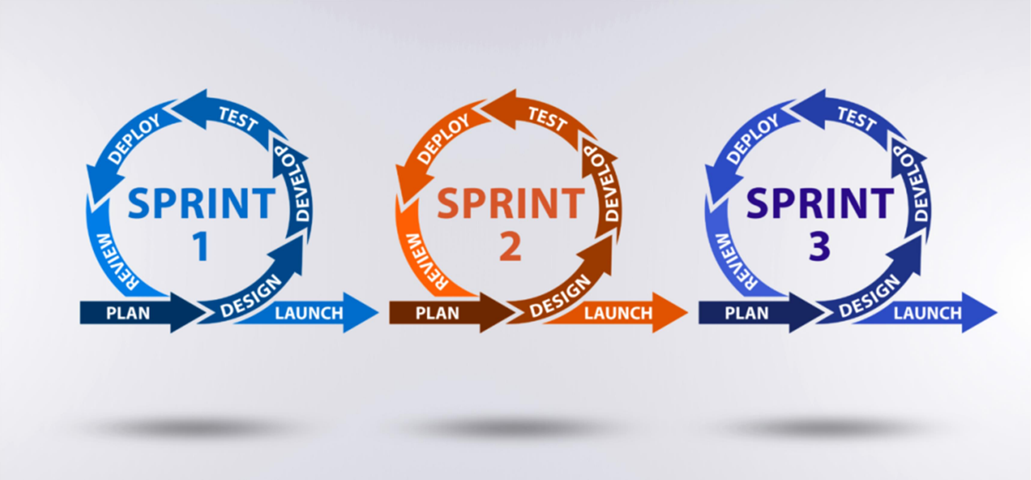
-> Client involvement is very high in Agile

-> Client is feedback is most important in Agile.

-> We will follow Scrum process

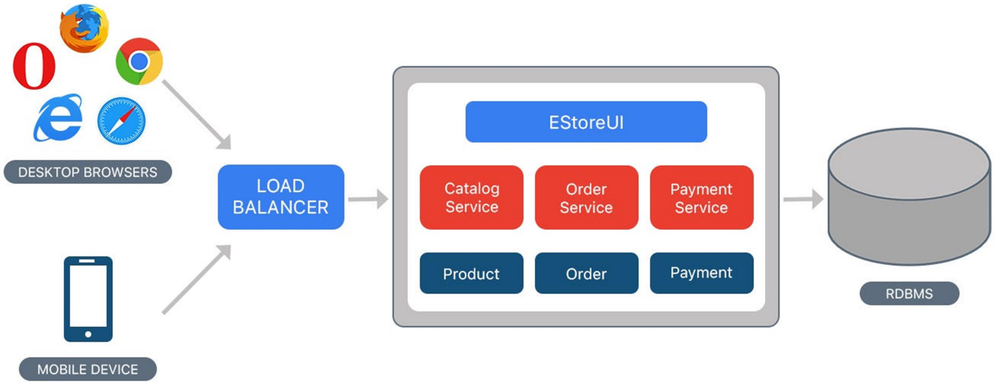
-> Scrum means Daily meeting for work updates.

- What we are doing
- When it will complete
- Any challenges in work
- Do we need any help



## What is Monolith Architecture?

If we develop all the functionalities in single project then it is called as Monolithic Architecture Based project.



**1) Single Point of failure**

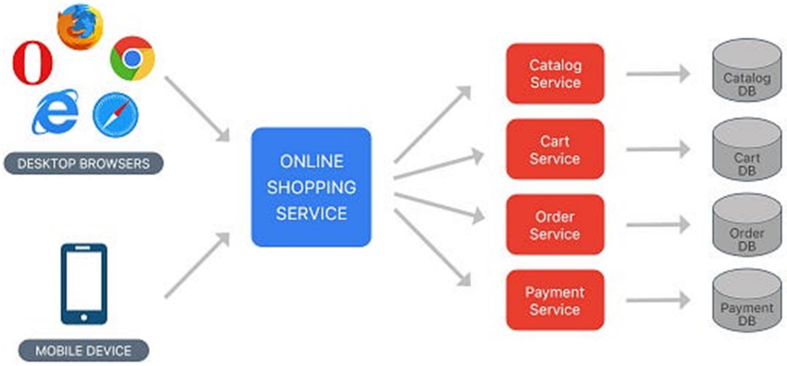
**2) Whole Project Re-Deploy**

**3) Burden on Server**

**4) Maintenance**

## Microservices Architecture

If we develop our application by using individual services then that is called as Microservices Architecture.



## What is Java?

=> Java is a computer programming language

=> Java Language developed by Sun Microsystem Company in 1991

=> The initial name of Java was 'OAK' programming language

-> In 1995 they renamed OAK to Java

-> Java Languages developed by a Team in Sun Microsystem. The teal lead is James Gosling.

-> Sun Microsystem sold out to Oracle Company in 2010. Now Java is under license of Oracle Corporation.

-> Java is free & open source.

**Sun Microsystem Divided Java into 3 parts**

1) J2SE : Java Standard Edition (JSE)
2) J2EE : Java Enterprise Edition (JEE)
3) J2ME : Java Micro Edition (JME)

-> JSE is used to develop standalone applications.

-> Standalone application means which is specific to only one system

ex: Calculator, Notepad

-> JEE is used to develop web application

-> Web application can be accessed by multiple users at a time

Ex: Gmail, Facebook, LinkedIn, Google, YouTube

-> JME is used to develop mobile applications

Ex: Whatsapp, fb messenger, instagram app....

## Java Installation

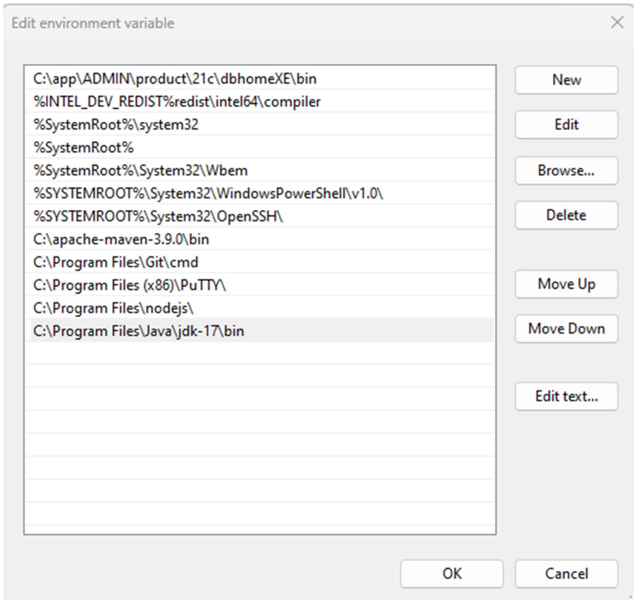
1) Download Java software from Oracle website

*Link: https://download.oracle.com/java/17/latest/jdk-17_windows-x64_bin.exe*

2) Install Java software (Double click on .exe file -> Next -> Next -> Finish)

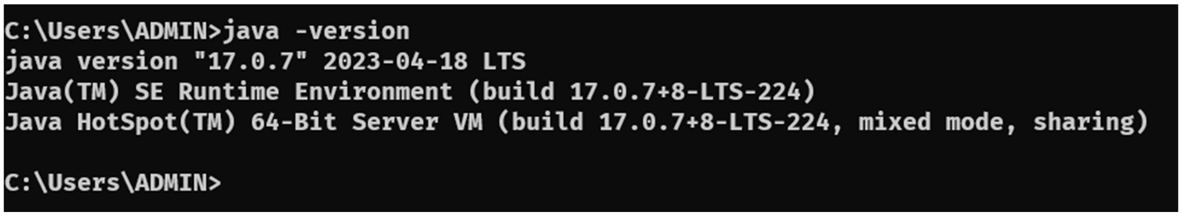
3) Set Path for Java up to JDK/bin directory (Open System Environment Variables)

Path = C:\Program Files\Java\jdk-17\bin



```
C:\app\ADMIN\product\21c\dbhomeXE\bin
%INTEL_DEV_REDIST%redist\intel64\compiler
%SystemRoot%\system32
%SystemRoot%
%SystemRoot%\System32\Wbem
%SYSTEMROOT%\System32\WindowsPowerShell\v1.0\
%SYSTEMROOT%\System32\OpenSSH\
C:\apache-maven-3.9.0\bin
C:\Program Files\Git\cmd
C:\Program Files (x86)\PuTTY\
C:\Program Files\nodejs\
C:\Program Files\Java\jdk-17\bin
```

4) Verify Java Installation (open command prompt and execute below command)



```
C:\Users\ADMIN>java -version
java version "17.0.7" 2023-04-18 LTS
Java(TM) SE Runtime Environment (build 17.0.7+8-LTS-224)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.7+8-LTS-224, mixed mode, sharing)

C:\Users\ADMIN>
```

== Java Installed Successfully ===

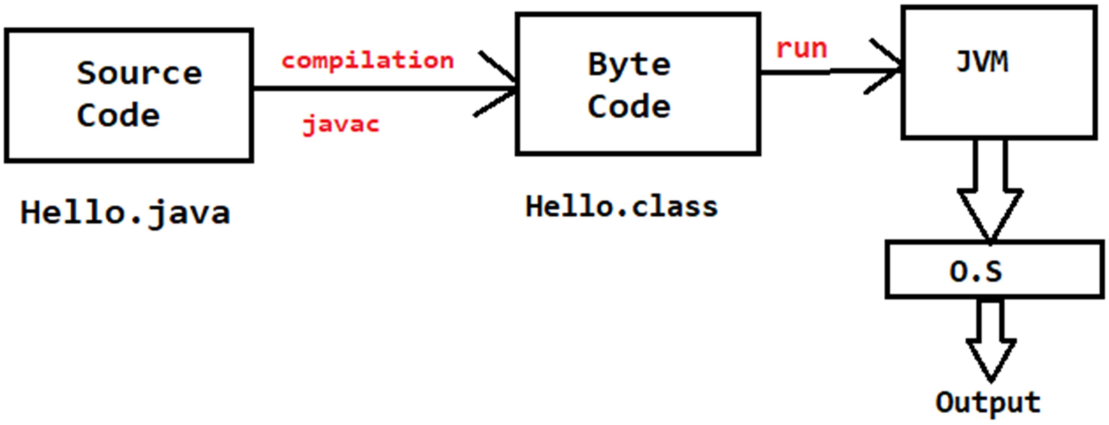
## Java Program Execution Flow

Step-1: Write Java Program and save it in a file with .java extension

Step-2: Compile Java Program using javac (It will generate .class file with byte code)

Step-3: Run .class file (JVM will start the program execution)

**Note: JVM will convert byte code into machine understandable code.**



## Java Program Structure

// package statement (To bind class to a package)

// import statements (To refer another package classes)

// class declaration (class is a plan/model)

// variables (To store the data)

// methods (To performs action)

## How to develop Java Program

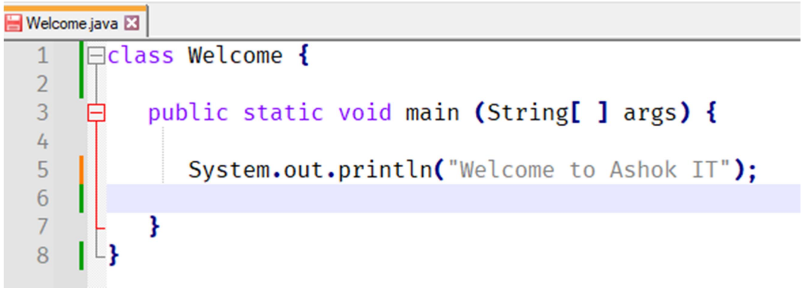
- We can use text editor to develop Java programs

  Ex: Notepad, Note Pad ++, Edit Plus etc..

- To compile and execute we will use Command Prompt (cmd)

## Developing First Java Program

1) **Create text file with .java extension (Ex : Welcome.java)**
2) **Write Java class in .java file and save it**
3) **Open command prompt**
4) **Compile and Run Java program**

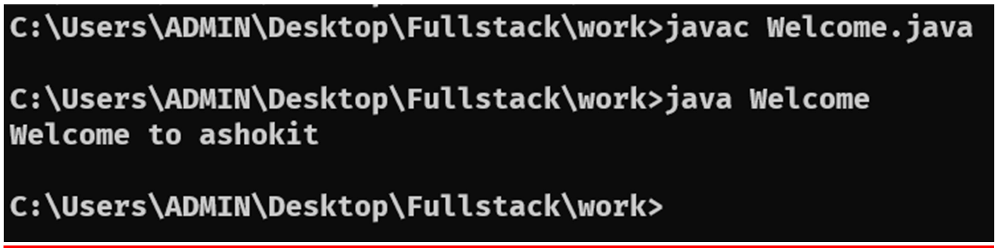


```
class Welcome {

   public static void main (String[ ] args) {

      System.out.println("Welcome to Ashok IT");

   }
}
```

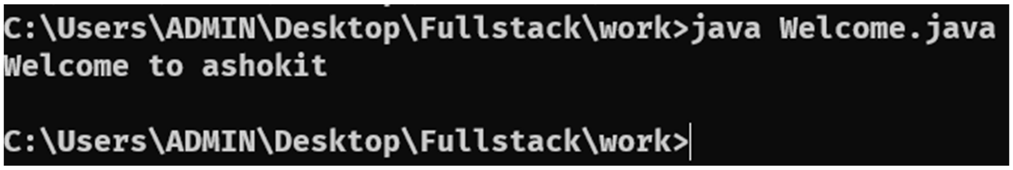


```
C:\Users\ADMIN\Desktop\Fullstack\work>javac Welcome.java

C:\Users\ADMIN\Desktop\Fullstack\work>java Welcome
Welcome to ashokit

C:\Users\ADMIN\Desktop\Fullstack\work>
```

Note:  From Java 11v onwards, we can execute java program directly without compilation. This is called as(Single Source File execution.



```
C:\Users\ADMIN\Desktop\Fullstack\work>java Welcome.java
Welcome to ashokit

C:\Users\ADMIN\Desktop\Fullstack\work>
```

## Comments:

We have the following comments in java:

- Single Line Comments
- Multi Line Comments
- Documentation Comments

Note: Java compiler will ignore comments in compilation phase.

## Identifier Rules (Mandatory to follow)

-> The name we are using for classes, variables, methods is called as Identifier.

Rule - 1: It can contain upper case characters ( A - Z )

Rule - 2: It can contain lower case characters ( a - z )

Rule - 3: It can contain digits ( 0 - 9 )

Rule - 4: Name should not start with digit

Rule - 5: Special characters only underscore (_) and dollar ($) allowed

Rule - 6: We should not use spaces in middle

## Naming Conventions (Optional But Recommended)

Class Name: Every Word First Letter Upper Case

Hello, Welcome, Report, UserService , UserDao

Variable Name: start with lower case and from second word onwards every word first character as Upper case

Age, userAge, creditCardNumber

Method Name: Similar to variable naming convention with brackets ( )

main ( ), login ( ), register ( ), generateReport ( )

Constants: Every letter uppercase

NAME, UNIVERSITY_NAME, COMPANY_NAME

## Data Types & Variables

=> Variables are used to store the data

```
age = 20

price = 2000.50

name = ashok

gender = m

isEmp = true
```

=> Data Types are used to specify what type of data we can store into variable.

1) Primitive Data Types (we have 8 )

- 1.1) Integral Data Types (To store numbers +ve or -ve)
    - 1.1.1) byte --> 1 byte
    - 1.1.2) short --> 2 bytes
    - 1.1.3) int --> 4 bytes
    - 1.1.4) long --> 8 bytes
- 1.2) Decimal Data Types ( To store decimal value +ve or -ve )
    - 1.2.1) float --> 4 bytes
    - 1.2.2) double --> 8 bytes
- 1.3) Character Data Type (To store single character )
    - 1.3.1) char --> 2 bytes
- 1.4) Boolean Datatype (to store true or false)
    - 1.4.1) boolean --> 1 bit

2) Referenced Data Types

- 2.1) Arrays : To store group of values in single variable
- 2.2) String : To Store group of characters

Note: In Java, every class will be treated as Referenced Data Type.

## Java Program on Variables

```
class VarDemo {

        public static void main(String[] args){

                byte age = 20;

                int id = 200; // declaration + initialization

                System.out.println(age);

                System.out.println(id);

                char gender = 'M';
                System.out.println(gender);

                double price = -125.60;

                System.out.println(price);

                boolean status = false;

                System.out.println(status);

                String name = "hi";

                System.out.println(name);

        }

}
```

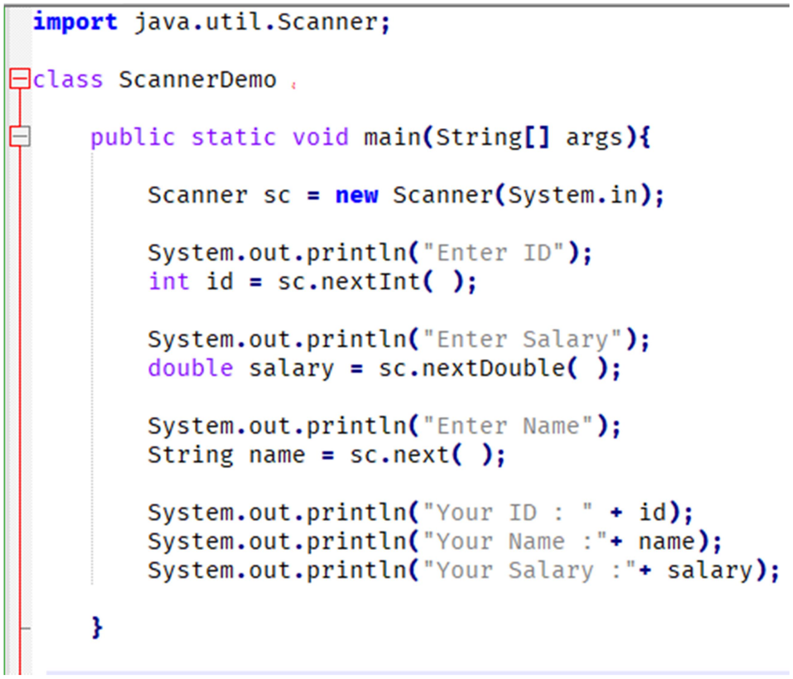
## Reading data from Keyboard

=> Scanner is a pre-defined class available in java

=> Scanner class is part of java.util package

=> To use Scanner class in our program first we need to import that class.

**import java.util.Scanner;**



```
import java.util.Scanner;

class ScannerDemo

    public static void main(String[] args){

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter ID");
        int id = sc.nextInt( );

        System.out.println("Enter Salary");
        double salary = sc.nextDouble( );

        System.out.println("Enter Name");
        String name = sc.next( );

        System.out.println("Your ID : " + id);
        System.out.println("Your Name :"+ name);
        System.out.println("Your Salary :"+ salary);

    }
```

## Programs to Practise On Variables and Data Types

1) How to read data from keyboard and print that on console

2) Sum of two numbers with dynamic input

3) Swap two numbers with third variable

4) Swap two numbers without using third variable

5) Age Calculation based on birth year

6) Write a java program to calculate Rectangle Area by reading width and height from keyboard.

7) Write a java program to calculate simple interest by taking data from keyboard.

8) Write a java program to print reminder and quotient by taking two numbers from keyboard

9) Write a java program to convert meters into centi-meters and milli-meters (take meter from keyboard).

10) Write a java program to calculate BMI by reading data from keyboard.

## Operators

=> Operator is a symbol which is used to perform some operation on data

Unary Operators: ++ (increment) , -- (decrement)

Arithmetic Operators: +, -, *, /, %

Relational Operators : >, >=, <, <=, ==, !=

Logical Operators: && (AND) , || (OR)

Assignment Operators: =

Ternary Operator: ? :

### Unary Operators

=> To perform operation on single variable we will use Unary Operators

=> Unary Operators are divided into 2 types

1) Increment ( ++ ) : Increment value by 1

    1.1) Pre Increment   ( ++ a ) : First Increment then use variable

    1.2) Post Increment  ( a ++ ) : First Use variable then increment

2) Decrement ( -- ) : Decrement value by 1

    2.1) Pre Decrement ( -- a ) : First Decrement then use variable

    2.2) Post Decrement ( a -- ) : First Use variable then decrement

```
class UnaryOperators {

    public static void main(String[] args){

        int a = 25;

        System.out.println(++ a); // 26

        System.out.println (a ++ ); // 26 => a = 27

        System.out.println(++ a); // 28

    }

}
```

```
class UnaryOperators {

    public static void main(String[] args){

        int a = 25;

        System.out.println(++ a + a ++ ); //52

        System.out.println(a);  // 27

    }

}
```

```
class UnaryOperators {

        public static void main(String[] args){

                int a = 25;

                System.out.println(++ a + a ++ + ++a ); // 80

                System.out.println( --a ); // 27

        }

}


class UnaryOperators {

        public static void main(String[] args){

                int a = 35;

                System.out.println( -- a ); // 34

                System.out.println( a -- ); // 34 => a = 33

                System.out.println(a) ; // 33

        }

}


class UnaryOperators {

        public static void main(String[] args){

                int a = 3;

                                                   2    + 2    - 1

                System.out.println( -- a + a -- - a++); // 3

                System.out.println(a) ; // 2

        }

}
```

## Arithmetic Operators

=> To perform calculations we will use arithmetic operators

1) Addition ( + )

2) Substraction ( - )

3) Multiplication ( * )

4) Division ( / ) (Quotient)

5) Modula's ( % ) (Reminder)

## Relational Operators

=> To check relations among the data / variables and return result as true/false

1) Greater Than & Equals ( > , >= )

2) Less than & equals ( < , <= )

3) Equality ( == )

4) Not Equals ( != )

## Logical Operators

=> To check more than one relation

1) AND ( && ) => If all conditions are true then only result will be true

2) OR ( || ) => If any one condition is true then result will be true

## Assignment Operators

1) Equal ( = )

```
Ex : int a = 10;
```

## Ternary Operator

=> It is called as Conditional Operator

```
int x = 15;

int y = 20;

int max = ( x > y ) ? x : y;

System.out.println("Max: "+ max);
```

```
}

class Operators {

        public static void main(String[] args){

                int a = 10;

                int b = 20;

                int c = 35;

                System.out.println(++a); // 11

                System.out.println(a++); // 11 => a = 12

                System.out.println(a--); //12 => a = 11

                System.out.println(--a); // 10

                System.out.println(a + b) ; // 30

                System.out.println(a >= b ); // false

                System.out.println(b >= a ); // true

                System.out.println(a == b ); // false

                System.out.println(a != b ); // true

                System.out.println(a == b && b==c ); // false

                System.out.println(a != b && b!=c ); // true

                System.out.println(a > b || b > c ) ; // false

                System.out.println(b > a || a > c ) ; // true

                int d = 10;

                System.out.println(d);

                int x = 5;

                int y = 20;

                int max = ( x > y ) ? x : y;

                System.out.println("Max: "+ max);

        }

}
```

## Control Statements

=> Control Statements are used to control program execution flow

1) Conditional Statements

    1.1) Simple if

    1.2) if - else

    1.3) Nested - if

    1.4) Switch Case

2) Looping Statements

    2.1) for loop

    2.2) while loop

    2.3) do-while

    2.4) for-each

3) Jumping Statements

    3.1) break

    3.2) continue

    3.3) return

## Simple if

Syntax :

```
        if ( condition ) {

                //statements

        }
```

```
import java.util.Scanner;

class SimpleIf {

    public static void main(String[] args){

        Scanner s = new Scanner(System.in);

        System.out.println("Enter Number");

        int a = s.nextInt ( );

        if ( a > 0 ){

            System.out.println("a is +ve num");

        }

        System.out.println("Done....");

    }

}
```

## if - else

Syntax :

```
    if ( condition ) {

        //stmts

    }else {

        // stmts

    }
```

**1) Write a java program to check given number is even or odd. Take number from keyboard.**

```
import java.util.Scanner;
class EvenOdd{
    public static void main(String[] args){
        Scanner s = new Scanner(System.in);
        System.out.println("Enter Number");
        int a = s.nextInt ( );
        if( a % 2 == 0 ){
            System.out.println("Even");
        }else{
            System.out.println("Odd");
        }
    }
}
```

**2) Write a java program to print "Hello" if given number is divisible by 5 otherwise print "Bye".**