

# CORE JAVA

---

## 1)What is Java?

Java is a object oriented, platform independent, case sensitive, strongly typed checking , high level , open source programming language developed by James Gosling in the year of 1995.

## 2)Features of Java?

- 1)Simple
- 2)Object oriented
- 3)Platform independent
- 4)Portable
- 5)Architecture Neutral
- 6)Highly secured
- 7)Robust
- 8)Multithreaded
- 9)Distributed
- 10)Dynamic

## 3)Differences between JDK , JRE and JVM ?

### JDK

- JDK stands for Java Development Kit.
- It is a installable software which contains compiler (javac) , interpreter (java), Java virtual machine (JVM), archiever (.jar) , document generator (javadoc) and other tools needed for java application development.

### JRE

- JRE stands for Java Runtime Environment.
- It provides very good environment to run java applications only.

### JVM

- JVM stands for Java Virtual Machine.
- It is an interpreter which is used to execute our program line by line procedure.

#### 4)Types of classloaders in java?

We have three types of predefined classloaders.

- 1)Bootstrap classloader (loads rt.jar file)
- 2)Extension classloader (loads all the jar files from ext folder)
- 3)System/Application classloader(it loads .class file from classpath).

#### 5)Is it possible to execute java program without main methods?

Till 1.6 version it is possible to execute java program without main method using static block. But from 1.7 version onwards it is not possible to execute java program without main method.

```
ex:  class A
      {
          static
          {
              System.out.println("Hello World");
              System.exit(0);
          }
      }
```

#### 6)What is typecasting?

Process of converting from one datatype to another datatype is called typecasting. Typecasting can be performed in two ways.

##### i)Implicit typecasting

- If we want to store small value into a bigger variable then we need to use implicit typecasting.
- A compiler is responsible to perform implicit typecasting.
- Implicit typecasting is also known as Widening/upcasting.

##### ii)Explicit typecasting

- If we want to store bigger value into a smaller variable then we need to use explicit typecasting.
- A programmer is responsible to perform explicit typecasting.
- Explicit typecasting is also known as Narrowing/Downcasting.

## 7)What is static import in java?

Using static import we can access static members directly.

```
Ex:  import static java.lang.System.*;
      class Test
      {
          public static void main(String[] args)
          {
              out.println("Hello World");
          }
      }
```

## 8)Which is a default package in java?

- java.lang package

## 9) What is JIT compiler

- IT is a part of a JVM which is used to increase the execution speed of our program.

## 10)How many memories are there in java?

We have five memories in java.

- 1) Method area
- 2) Heap
- 3) JAvA Stack
- 4) PC Register
- 5) Native method stack

## 11) What is native method in java?

A Method which is developed by using some other language is called native method.

## 12) What is Garbage Collector ?

Garbage collector is responsible to destroy unused or useless objects in java.

There are two ways to call garbage collector in java.

- 1) System.gc()
- 2) Runtime.getRuntime().gc()

### **13) What is identifier?**

- A name in java is called identifier.
- It can be class name, variable name, method name and label name.

### **14) What .class file contains ?**

- A .class file contains byte code instructions.

### **15) Is java support access specifiers?**

Java does not support access specifiers.

Java support access modifiers.

- 1) default
- 2) public
- 3) private
- 4) protected

### **16) What is program?**

- Program is a collection of instructions (or) Program is a set of instructions.

### **17)Types of variables in java?**

we have three types of variables in java.

- 1) Instance variable
- 2) Static variable
- 3) Local variable

### **18) Types of blocks in java?**

We have three types of blocks in java.

- 1) Instance block
- 2) Static block
- 3) Local block

## 19) Explain main method ?

### public:

JVM wants to call this method from any where that's why main method is public.

### static:

JVM wants to call this method without using object reference.

### void:

Main method does not return anything to JVM.

### main:

It is an identifier given to a main method.

### String[] args:

It is a command line argument.

## 20) Is java purely object oriented or not?

- No, java will not consider as purely object oriented because it does not support many OOPS concepts like multiple inheritance, operator overloading and more ever we depends upon primitive datatypes which are non-objects.

# OOPS

## 1) What is class?

- A class is a blue print of an object.
- A class is a collection of variables and methods.
- A class is a reusable component.
- A class will accept following modifiers.

### ex:

default, public, final, abstract

We can declare a class as follow.

### syntax:

optional

|

modifier class class\_name <extends> Parent\_classname

<implements> Interface\_name

```
{  
    - // variables and methods  
}
```

## 2)What is the difference between default class and public class?

### default class

We can access default class within the package.

```
ex:    class A  
    {  
    }
```

### public class

We can access public class within the package and outside of the package.

```
ex: public class A  
    {  
    }
```

## 3)What is final class?

- If we declare any class as final then creating child class is not possible.
- If we declare any class as final then extending some other class is not possible.

ex:

```
final class A  
{  
}  
class B extends A // invalid  
{  
}
```

## Q)What is abstract class ?

If we declare any class as abstract then creating object of that class is not possible.

ex:

```
abstract class A  
{
```

}

### 3) What is object ?

- It is a outcome of a blue print.
- It is a instance of a class.
- Instance means allocating memory for our data members.

#### Q)Object class

- Object class present in java.lang package.
- Object class consider as a parent class for every java program.
- Object class contains following methods.

#### Q)toString()

- It is a method present in Object class.
- Whenever we are trying to display any object reference directly or indirectly toString() method will be executed.

#### Q)Data Hiding

- It is used to hide the data from the outsider.
- It means outside person must not access our data directly.
- Using private modifier we can implement data hiding concept.

```
ex:  class Account
      {
          private double balance;
      }
```

### 4)Types of objects?

We have two types of objects.

#### 1)Immutable object

➤ After object creation if we perform any changes then for every change a new object will be created such behaviour is called immutable object.

ex: String  
 Wrapper classes.

## **2)Mutable object**

- After object creation if we perform any changes then all the changes will reflect to single

object such behaviour is called mutable object.

ex:

StringBuffer  
StringBuilder

## **5)What is singleton class?**

- A class which allows us to create only one object is called singleton class.
- To declare a singleton class we required private constructor and static method.

**ex:**

```
class Singleton
{
    private static Singleton singleton=null;
    private Singleton()
    {
    }
    public static Singleton getInstance()
    {
        if(singleton==null)
        {
            singleton=new Singleton();
        }
        return singleton;
    }
}
```

## **6)What is hashCode?**

- For every object , JVM will create a unique identifier number i.e hash code.
- In order to read hash code we need to use hashCode() method of Object class.

**ex:**

```
Test t=new Test();
System.out.println(t.hashCode());
```



## 7)Interfaces?

- Interface is a collection of zero or more abstract methods.
- Abstract method is a incomplete method which ends with semicolon and does not have any body.

**ex:**

```
public abstract void m1();
```

- It is not possible to create object for interfaces.
- To write the implementation of abstract methods of an interface we will use implementation class.
- It is possible to create object for implementation class because it contains method with body.
- By default , every abstract method is a public and abstract.
- Interface contains only constants i.e **public static final**.

**Syntax:** interface interface\_name

```
{  
    - //constants  
    - //abstract method  
}
```

In java, a class can't extends more than one class.

But interface can extends more than one interface.

## 8)Abstract class?

- Abstract class is a collection of zero or more abstract methods and zero or more concrete methods.
- A "abstract" keyword is applicable only for class and method but not for variable.
- It is **not possible to create object** for abstract class.
- To write the implementation of abstract methods of an abstract class we will use sub classes.
- By default every abstract **method is a public and abstract**.
- Abstract class contains **only instance variables**.

**syntax:**

```
abstract class class_name  
{  
    - //instance variables  
    - //abstract methods  
    - //concrete methods
```

}

## 9)What is Abstraction?

Hiding the internal implementation and highlighting the set of services is called abstraction.

Best example of abstraction is ATM machine, coffee machine, calculator, phone and etc.

The main advantages of abstraction are.

- 1)It gives security because it will hide internal implementation from the outsider.
- 2)Enhancement becomes more easy without affecting enduser they can perform any changes in our internal system.
- 3)It provides flexibility to the enduser to use the system.
- 4)It improves maintainability of an application.

## 10)What is encapsulation?

➤ The process of encapsulating or grouping variables and its associated methods in a single entity is called encapsulation.

- In encapsulation for every variable we need to declare **setter and getter methods**.
- A class is said to be an encapsulated class if it supports **data hiding + abstraction**.

The main advantages of encapsulation are.

- 1)It gives security.
- 2)Enhancement becomes more easy.
- 3)It provides flexibility to the enduser to use the system.
- 4)It improves maintainability of an application.

The main disadvantage of encapsulation is, it will increase the length of our code and slow down the execution process.

ex:    class Employee

```
{
    private int empId;
    //setter
    public void setEmpId(int empId)
    {
        this.empId=empId;
    }
    //getter
    public int getEmpId()
```

```

        {
            return empId;}
    }

```

## 11)What is the difference between POJO class and Java Bean class?

### POJO class:

A class is said to be pojo class if it supports following two properties.

- 1)All variables must be private.
- 2)All variables must have setter and getter method.

### Java Bean class:

A class is said to be java bean class if it supports following four properties.

- 1)A class must be public.
- 2)A class must have constructor.
- 3)All variables should be private.
- 4)All variables should have setter and getter method.

Note:

Every Java bean class is a pojo class.

But every pojo class is not a java bean class.

## CONSTRUCTORS

### 1)What is Constructor?

Constructor is a special method which is used to initialize an object.

ex:

```
Test t=new Test();
```

Having same name as class name is called constructor.

A constructor will execute when we create an object.

A constructor does not allowed any returntype.

A constructor will accept following modifiers.

ex:default, public, private , protected

ex:class A

```

{
    A()

```

```

        {
            System.out.println("0-arg const");
        }
    }
ex: class A
    {
        A(int i)
        {
            System.out.println("parameterized const");
        }
    }

```

### **1) Userdefined constructor:**

A constructor which is created by the user based on the application requirement is called userdefined constructor.

It is categorized into two types.

i) Zero Argument constructor

ii) Parameterized constructor

i) Zero Argument constructor

Suppose if we are not passing at least one argument to userdefined constructor is called zero argument constructor.

### **2) Parameterized constructor:**

Suppose if we are passing at least one argument to userdefined constructor is called parameterized argument constructor.

2) What is default constructor?

It is a compiler generated constructor for every java program where we are not defining any constructor.

Default constructor is an empty implementation.

To see the default constructor we need to use the below command.

ex:

```
javap -c Test
```

### **3) Constructor Overloading:**

Having same constructor name with different parameters in a single class is called constructor overloading.

4) What is method overloading?

Having same method name with different parameters in a single class is called method overloading.

All the methods present in a class are called overloaded methods.

Method resolution will taken care by compiler based on reference type.

ex: class A

```
{
    public void m1()
    {
        System.out.println("0-arg method");
    }
    public void m1(int i)
    {
        System.out.println("int-arg method");
    }
}
```

## 5)Can we overload main method in java?

➤ Yes, we can overload main method in java.But JVM always execute main method with String[] parameter only.

## 6) What is method overriding?

- Having same method name with same parameters in two different classes is called method overriding.
- Methods which are present in **parent class** are called **overridden methods**.
- Methods which are **present in child class** are called **overriding methods**.
- Method resolution will taken care by JVM based on runtime objects.

ex: class A

```
{
    public void m1()
    {
        System.out.println("ITALENT");
    }
}
class B extends A
{
    public void m1()
```

```

        {
            System.out.println("IIHUB TALENT");
        }
    }

```

### **7)Can we override final methods in java?**

ans) No , we can't override final methods in java.

### **29) What is method overloading?**

Having same method name with different parameters in a single class is called method overloading.

All the methods present in a class are called overloaded methods.

Method resolution will taken care by compiler based on reference type.

ex:

```

class A
{
    public void m1()
    {
        System.out.println("0-arg method");
    }
    public void m1(int i)
    {
        System.out.println("int-arg method");
    }
}

```

### **8)What is Method Hiding?**

Method hiding is exactly the same as method overriding with following differences.

### **Q)DIFFERENCE B/W METHOD OVEERIDING AND HIDING?**

## **Method overriding**

All the methods present in method overriding must be **non-static**.

Method **resolution** will taken care by **JVM** on runtime object.

It is also known as **runtime polymorphism**, **dynamic polymorphism**, **late binding**.

## **Method hiding**

All the methods present in method hiding must be **static**.

Method resolution will taken care based by **compiler** based on **reference type**.

It is also known as **compile time polymorphism**, **static polymorphism**, **early binding**.

### **9)Can we override static methods in java?**

No , we can't override static methods in java.

### **10)Can we override main method in java?**

No , we can't override main method because it is static.

### **11)What is this keyword?**

A "this" keyword is a java keyword which is used to **refer current class object reference**.

We can utility this keyword in following ways.

- i)To refer current class variables
- ii)To refer current class methods
- iii)To refer current class constructors

### **12)what is super keyword**

A "super" keyword is a java keyword which is used to refer **super class object reference**.

We can utility super keyword in following ways.

- i)To refer super class variables
- ii)To refer super class methods
- iii)To refer super class constructors

### **13) What is API?**

API stands for application programming interface.

It is a base for the programmer to develop software applications.

API is a collection of packages.

We have three types of API's.

1)Predefined API:

Built-In API is called predefined API.

2)Userdefined API:

API which is created by the user based on the requirement is called userdefined API.

3)Third party API:

API which is given by third party vendor.

ex: JAVAZOOM API , Text API and etc.

#### **14)Differences between interface and abstract class?**

<b>Interface</b>	<b>abstract</b>
1)To declare interface we will use interface keyword.	1) To declare abstract class we will use abstract keyword.
2)Interface is a collection of abstract methods,default methods and static methods.	2)Abstract class is a collection of abstract methods and concrete methods.
3)Interface contains constants.	3)Abstract class contains instance variables.
4)We can achieve multiple inheritance.	4)We can't achieve multiple inheritance.
5)It does not support constructor.	5)It supports constructor.
6)It does not support blocks.	6)It supports blocks.
7)To write the implementation of abstract methods we need to use implementation class.	7)To write the implementation of abstract methods we need to use sub class.
8)If we know only specification then we need to use interface.	8)If we know partial implementation then we need to use abstract class.



--	--

### 15)What is polymorphism?

- Polymorphism has taken from Greek Word.
- Here poly means many and morphism means forms.
- The ability to represent in different forms is called polymorphism.
- In java, polymorphism is categories into two types.

1)Compile time polymorphism/ static polymorphism / early binding.

- A polymorphism which exhibits at compile time is called compile time polymorphism.

ex:

Method overloading  
Method Hiding

2)Runtime polymorphism / dynamic polymorphism / late binding.

- A polymorphism which exhibits at runtime is called run time polymorphism.

ex:

Method overriding

### 16)What is Inheritance?

- Inheritance is a mechanism where one class will derived from the properties from another class.

Using "extends" keyword we can implements inheritance.

We have five types of inheritance.

- 1) Single level inheritance.
- 2) Multi level inheritance
- 3) Multiple inheritance
- 4) Hierarchical inheritance
- 5) Hybrid inheritance

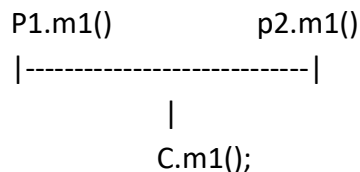
### 17) What are the inheritance not support by java?

- Java does not support multiple inheritance and hybrid inheritance.

## 18) Why java does not support multiple inheritance?

- There is a chance of raising ambiguity problem that's why java does not support multiple inheritance.

ex:



But from java 1.8 version onwards java supports multiple inheritance through default methods of interface.

## 19) What is Has-A relationship

- Has-A relationship is also known as composition and aggregation.
- There is no specific keyword to implement Has-A relationship but mostly we will use new operator.
- The main objective of Has-A relationship is to provide reusability.
- Has-A relationship will increase dependency between two components.

ex:

```
class Engine
{
    - //engine specific functionality
}
class Car
{
    Engine e=new Engine();
    -
}
```

### Composition:

- Without existing container object there is a no chance of having contained object then the relationship between

- container object and contained object is called composition which is strongly association.

### **Aggregation:**

- Without existing container object there is a chance of having contained object then the relationship
- between container object and contained object is called aggregation which is loosely association.

### **20)What is marker interface?**

- An interface which does not have any constants and abstract methods is called marker interface.
- In general, empty interface is called marker interface.
- Marker interfaces are empty interfaces but by using those interfaces we get some ability to do.

ex:

Serializable  
Remote

### **21) What is inner class?**

- Sometimes we will declare a class inside another class such concept is called inner class.

ex: class Outer\_Class

```
{
    class Inner_Class
    {
        - //code to be execute
    }
}
```

Inner classes introduced as a part of event handling to remove GUI bugs.

But because of powerful features and benefits of inner classes, Programmers started to use inner classes in regular programming code.

Inner class does not accept static members i.e static variable, static method and static block.

### **22)What is package?**

Package is a collection of classes, interfaces, enums ,Annotations, Exceptions and Errors.  
Enum,Exception and Error is a special class and Annotation is a special interface.

In general, a package is a collection of classes and interfaces.

Package is also known as folder or a directory.

In java, packages are divided into two types.

All package names we need to declare under lower case letters only.

### **1)Predefined packages:**

Built-In packages are called predefined packages.

ex: java.lang , java.io, java.util

### **2)Userdefined package:**

Packages which are created by ther user are called userdefined packages.

It is highly recommanded to use package name in the reverse order of url.

ex:

com.ihubtalent.www

## **Arrays**

### **1)what is Array?**

- In a normal variable we can store only one value at a time.
- To store more then one value in a single variable then we need to use arrays.

Definition:

- Array is a collection of homogeneous data elements.

The main advantages of array are.

1)We can represent multiple elements using single variable name.

ex: int[] arr={10,20,30};

2)Performance point of view array is recommanded to use.

The main disadvantages of array are.

1)Arrays are fixed in size.Once if we create an array there is no chance of increasing and decreasing the size of an array.

2)To use array concept in advanced we should know what is the size of an array which is always not possible.

In java, arrays are categorised into three types.

- 1)Single Dimensional Array
- 2)Two Dimensional Array / Double Dimensional Array
- 3)Multi-Dimensional Array / Three Dimensional Array

## 2)What is the difference b/w length and length() ?

### Length

It is a final variable which is applicable only for arrays

It will return size of an array.  
String.

Ex:

```
class Test
{
    public static void main(String[] args)
    {
        int[] arr=new int[3];
        System.out.println(arr.length);//3

        System.out.println(arr.length());
    }
}
```

### length()

It is a final method which is applicable only for String objects.

It will return number of character present in

ex:

```
class Test
{
    public static void main(String[] args)
    {
        String str="hello";
        System.out.println(str.length());//5

        System.out.println(str.length());//C.T.E
    }
}
```

## 3)What is Recursion?

- A method is called self for many number of times is called Recursion.
- Recursion is similar to loopings.
- In Recursion post Increment and Decrement are used

## 4) What is Enum?

- Enum is a group of named constants.
- Enum concept is introduced in 1.5 version.
- Using enum we can create our own datatype called enumerated datatype.
- When compare to old language enum, java enum is more powerful.

- Enum is a special class.
- To declare enum we need to use enum keyword.

Syntax:

```
enum enum_type_name
{
    val1,val2,...,valN
}
```

Ex:

```
enum Months
{
    JAN,FEB,MAR
}
```

## 5) what is Wrapper classes?

- The main objective of wrapper class is used to wrap primitive to wrapper object and vice versa.
- To defined serveral utility methods.

### Primitive type

byte  
short  
int  
long  
float  
double  
boolean  
char

### Wrapper class

Byte  
Short  
Integer  
Long  
Float  
Double  
Boolean  
Character

### constructors

Every wrapper contains following two constructors. One will take corresponding primitive as an argument and another will take corresponding String as an argument.

### wrapper class

Byte  
Short  
Integer  
Long  
Float  
Double

### constructor

byte or String  
short or String  
int or String  
long or String  
float or String  
double or String

Boolean  
Character

boolean or String  
char

## **STRINGS**

### **1) What is Strings?**

String is a collection/set of characters.

String is a immutable object.

case1:

Once if we create a String object we can't perform any changes.If we perform any changes then for every change a new object will be created such behaviour is called immutability of an object.

### **2)Difference between == and .equals() method**

It is a equality operator or comparision operator.

It is used for reference comparision or address comparision.

.equals()

It is a method present in String class.

It is used for content comparision.

It is a case sensitive.

### **3)Difference between StringBuffer and StringBuilder?**

<b>StringBuffer</b>	<b>StringBuilder</b>
1.All the methods present in StringBuffer are synchronized.	1.All the methods present in StringBuffer are not synchronized.
2.At a time only one thread is allowed to access StringBuffer object.Hence we can achieve thread safety.	2.Multiple threads are allowed to access StringBuilder object.Hence we can't achieve thread safety.
3.Waiting time of a thread will increase effectively performance is low.	3. There is no waiting thread Effectively performance is high.

4.StringBuffer introduced in 1.0v.	4.StringBuilder introduced in 1.5v.
------------------------------------	-------------------------------------

## **EXCEPTION HANDLING**

### **1)What is the difference between Exception and Error?**

#### **Exception**

Exception is a problem for which we can provide solution programmatically.

Exception will raise due to syntax errors.

ex:

ArithmeticException  
FileNotFoundException  
IllegalArgumentException

#### **Error**

Error is a problem for which we can't provide solution programmatically.

Error will raise due to lack of system resources.

ex:

OutOfMemoryError  
StackOverflowError  
LinkageError  
and etc.

As a part of application development it is a responsibility of a programmer to provide smooth termination for every java program.

We have two types of terminations.

1)Smooth termination / Graceful termination

2)Abnormal termination

#### **1)Smooth termination:**

- During the program execution suppose if we are not getting any interruption in the middle of the program such type of termination is called smooth termination.

**ex:**

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



}

## **2)Abnormal termination:**

- During the program execution suppose if we are getting any interruption in the middle of the program such type of termination is called abnormal termination.

**ex:**

```
class Test
{
    public static void main(String[] args)
    {
        System.out.println(10/0);
    }
}
```

## **2) What is Exception?**

- It is a unwanted, unexpected event which disturbs normal flow of our program.
- Exception always raised at runtime so they are also known as runtime events.
- The main objective of exception handling is to provide graceful termination.

In java , exceptions are divided into two types.

1)Predefined exceptions

2)Userdefined exceptions

### **1)Predefined exceptions**

Built-In exceptions are called predefined exceptions.

It is categories into two types.

i)Checked exception

Exceptions which are checked by a compiler at the time of compilation is called checked exception.

ex:

```
IOException
InterruptedException
```

ii)Unchecked exceptions

Exceptions which are checked by a JVM at the time of runtime is called unchecked exceptions.

ex:

```
ArithmeticException
```

IllegalArgumentException  
ClassCastException  
and etc

If any checked exception raised in our program we must and should handle that exception by using try and catch block.

## **2) Userdefined exceptions**

Exceptions which are created by the user based on the application requirement are called customized exceptions.

ex:

StudentsNotPracticingException  
NoInterestInJavaException  
InsufficientFeeException

## **3) What is try block?**

- It is a block which contains risky code.
- A try block always associate with catch block.
- A try block is used to throw the exception to catch block.
- If any exception raised in try block then try block won't be executed.
- If any exception raised in the middle of the try block then reset of code won't be executed.

## **4) what is catch block?**

- It is a block which contains Error handling code.
- A catch block always associate with try block
- A catch block is used to catch the exception which is thrown by try block
- A catch block will take exception name as a parameter and that name must match with exception class name.
- If there is no exception in try block then catch block won't be executed.

### **Syntax:**

```
try
{
    - //Risky code
}
catch(ArithmeticException ae)
{
```

```
        - //Error handling code
    }
```

## Q)How to display exception details?

Throwable class defines following three method to display exception details.

### 1)printStackTrace()

- It will display name of the exception, description of the exception and line number of the exception.

### 2)toString()

- It will display name of the exception and description of the exception.

### 3)getMessage()

- It will display description of the exception.

### 4)What is finally block

- It is never recommended to maintain cleanup code inside try block because if any exception raised in try block then try won't be executed.
- It is never recommended to maintain cleanup code inside catch block because if no exception raised in try block then catch won't be executed.
- But we need a place where we can maintain cleanup code and it should execute irrespective of exception raised or not. Such block is called finally block.

### syntax

```
try
{
    - // Risky Code
}
catch(Exception e)
{
    - // Errorhandling code
}
finally
{
    - //cleanup code
}
```

## 5)What is the difference between final, finally and finalized method

### Final:

- A final is a modifier which is applicable for variables, methods and classes.
- If we declare any variable as final then reassignment of that variable is not possible.
- If we declare any method as final then overriding of that method is not possible.
- If we declare any class as final then creating child class is not possible.

### Finally

- It is a block which contains cleanup code and it will execute irrespective of exception raised or not.

### finalized method

- It is a method called by garbage collector just before destroying an object for cleanup activity.

## 6) What is throw statement

Sometimes we will create exception object explicitly and handover to JVM manually by using throw statement.

## 7) What is throws statement

If any checked exception raised in our program we must and should handle that exception by using try and catch block or by using throws statement.

## 8) what is Generics?

Array is a typesafe.

We can provide guarantee that what type of elements are present in array.

If requirement to store String values then we need to use String[] array.

ex:

```
String[] str=new String[100];  
str[0]="hi";  
str[2]=10; // invalid
```

At the time of retrieving the data from array , we don't need to perform any typecasting

ex:

```
String[] str=new String[100];
```

```
str[0]="hi";
```

---

## Collections

### 1. Differences between Arrays and Collections?

Arrays	Collections
It is a collection of homogeneous data elements.	It is a collection of homogeneous and heterogeneous data elements.
Arrays are fixed in size.	Collections are growable in nature.
Performance point of view arrays are recommended to use.	Memory point of view collections are recommended to use.
Arrays are type safe.	Collections are not type safe.
Arrays are not implemented based on data structure concept so we can't expect any readymade (utility) methods.	Collections are implemented based on data structure concept so we can expect readymade (utility) methods.
It holds primitive and object types.	It holds only object types but not primitive types.

### 2. What is Collection framework?

It defines several classes and interfaces to represent a group of objects as a single entity.

### 3. What is Collection?

- Collection is an interface which is present in java.util package.
- It is a root interface for entire Collection framework.
- If we want to represent group of individual objects in a single entity then we should go for Collection.
- Collection interface defines the most common methods which can be applicable for any collection object.

### 4. Differences between Collection and Collections?

- Collection is an interface which can be used to represent a group of objects as a single entity.
- Collections is an utility class present in java.util package to define several utility methods for Collection objects.

## 5. What are the methods present in Collection interface?

The following is the list of methods present in Collection interface.

- boolean add(Object o);
- boolean addAll(Collection c);
- boolean remove(Object o);
- boolean removeAll(Collection c);
- boolean retainAll(Collection c);
- void clear();
- boolean contains(Object o);
- boolean containsAll(Collection c);
- boolean isEmpty();
- int size();
- Object[] toArray();
- Iterator iterator();

## 6. Differences between ArrayList vs Vector?

ArrayList	Vector
No method is synchronized	Every method is synchronized
At a time multiple Threads are allow to operate on ArrayList object and hence ArrayList object is not Thread safe.	At a time only one Thread is allow to operate on Vector object and hence Vector object is Thread safe.
Relatively performance is high because Threads are not required to wait.	Relatively performance is low because Threads are required to wait.
It is non legacy and introduced in 1.2v.	It is legacy and introduced in 1.0v.

## 7. Differences between ArrayList and LinkedList?

ArrayList	LinkedList
The underlying data structure is resizable	The underlying data structure is doubly

array or growable array.	linked list.
ArrayList is better for storing and accessing data.	LinkedList is better for manipulating data.
The memory location for the elements of an ArrayList is contiguous.	The location for the elements of a linked list is not contagious.
When an ArrayList is initialized, a default capacity of 10 is assigned to the ArrayList.	There is no case of default capacity in a LinkedList.

## 8. Differences between List and Set?

List	Set
If we want to represent group of individual objects in a single entity where duplicates are allowed and order is preserved then we need to use List.	If we want to represent group of individual objects in a single entity where duplicates are not allowed and order is not preserved then we need to use Set.
List allows us to add any number of null values.	Set allows us to add at least one null value in it.
List implementation classes are ArrayList, LinkedList and Vector.	Set implementation classes are HashSet, LinkedHashSet and TreeSet.
ListIterator cursor is used to iterate the List elements.	Iterator cursor is used to iterate the set elements.

## 9. Differences between HashSet and LinkedHashSet?

HashSet	LinkedHashSet
The underlying data structure is Hashtable.	The underlying data structure is Hashtable and LinkedList.
Insertion order is not preserved.	Insertion order is preserved.
Introduced in 1.2 version.	Introduced in 1.4 version.

## 10. Differences between HashSet and TreeSet?

HashSet	TreeSet
The underlying data structure is Hashtable.	The underlying data structure is Balanced Tree.
Null insertion is possible.	Null insertion is not possible.
Heterogeneous objects are allowed.	Heterogeneous objects are not allowed.

Insertion order is not preserved.	Insertion order is sorting order of an object.
-----------------------------------	--

### 11.Differences between Comparable and Comparator interface?

Comparable	Comparator
It is present in java.lang package	It is present in java.util package
It contains only one method i.e compareTo()	It contains two methods i.e compare() and equals()
If we depend upon natural sorting order then we need to use Comparable interface.	If we depend upon customized sorting order then we need to use Comparator interface.

### 12.What is Map interface?

- Map is an interface which is present in java.util package.
- It is not a child interface of Collection interface.
- If we want to represent group of objects in key, value pair then we need to use Map.
- Each key, value pair is called single entry.
- Both key and value must be objects.
- Key can't be duplicate but value can be duplicate.

### 13.Differences between HashMap and LinkedHashMap?

HashMap	LinkedHashMap
The underlying data structure is Hashtable.	The underlying data structure is Hastable and LinkedList.
Insertion order is not preserved.	Insertion order is preserved.
Introduced in 1.2 version.	Introduced in 1.4 version.

### 14.Differences between HashMap and TreeMap?

HashMap	TreeMap
The underlying data structure is Hashtable.	The underlying data structure is Red Black Tree.



Insertion order is not preserved.	Insertion order is sorting order of an object.
Both key and value can be null.	Key can't be null but value can be null.

### 15.Differences between TreeMap and Hashtable?

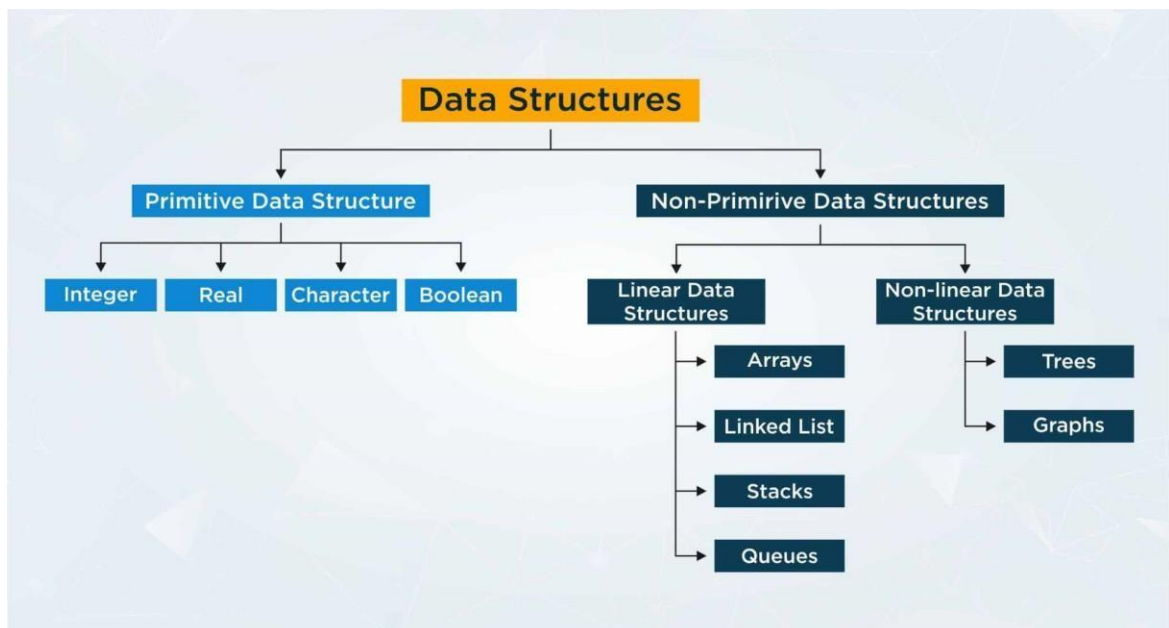
<b>TreeMap</b>	<b>Hashtable</b>
The underlying data structure is Red Black Tree.	The underlying data structure is Hashtable.
Key can't be null but value can be null.	Both key and value can't be null.
Both key and value can be null.	Key can't be null but value can be null.
It is a non-legacy class.	It is a legacy class
It is introduced in 1.2 version.	It is introduced in 1.0 version
Methods are not synchronized.	All methods are synchronized.

## 16.Types of Cursors in java?

We have three types of cursors in java.

Enumeration	Iterator	ListIterator
It is used to read objects one by one from legacy Collection objects.	It is used to read objects one by one from any Collection objects.	It is used to read objects one by one from List Collection objects.
It contains 2 methods i.e hasNextElements() and nextElement().	It contains 3 methods i.e hasNext(), next() and remove()	It contains 9 methods i.e hasNext(), next(), hasPrevious(), previous(), remove(), set(), add(), previousIndex() and nextIndex().
It performs read operation.	It performs read and remove operation.	It perform read, remove, adding and replacement of new objects.
We can create object by using elements() method.	We can create object by using iterator() method.	We can create object by using listIterator() method.
It is not a universal cursor.	It is a universal cursor.	It is a bi-directional cursor.

## 17.Types of Data Structure in java?



# MULTI-THREADING

---

## 1)What is the difference between Thread and Process?

### Thread:

- Thread is a light weight sub process.
- We can run multiple threads concurrently.
- One thread can communicate with another thread.

ex:

class is one thread  
constructor is one thread  
block is one thread  
and etc.

### PROCESS:

- Process is a collection of threads.
- We can run multiple process concurrently.
- One process can't communicate with another process.They are independent to each other.

ex:

Opening a notepad and typing java notes is one process.  
Downloading a file from internet is one process.  
Taking class in zoom meeting is one process.

## 2)What is Multi-tasking?

Executing several task simultaneously such concept is called multi-tasking.

Multi-tasking is divided into two types.

### 1)Thread based multi-tasking

Executing several task simultaneously where each task is a same part of a program.  
It is best suitable for programmatic level.

### 2)Process based multi-tasking

Executing several task simultaneously where each task is a independent process.  
It is best suitable for OS level.

### **3)Multi-Threading?**

Executing several threads simultaneously such concept is called multi-threading.

In multi-threading only 10% of work must be done by a programmer and 90% work will be done by JAVA API.

The main important application area of multi-threading are.

- 1)To develop multimedia graphics
- 2)To develop animation
- 3)To develop video games

Ways to start a thread in java

We can start or create or instantiate a thread in two ways.

- 1)By extending Thread class
- 2)By implementing Runnable interface

Life cycle of a thread

Once if we create a thread object then our thread will be in new/born state.

Once if we call t.start() method then our thread will go to ready/runnable state.

If thread scheduler allocates the CPU then our thread will enter to running state.

Once the run() method execution is completed then our thread will go to dead state.

### **4)Thread priority?**

In java, every thread has a priority automatically generated by JVM or explicitly provided by the programmer.

The valid range for thread priority is 1 to 10. 1 is a least priority and 10 is a highest priority.

Thread class defines following standard constants as thread priorities.

ex:

Thread.MAX\_PRIORITY -- 10

Thread.NORM\_PRIORITY -- 5

Thread.MIN\_PRIORITY -- 1

### **5)Setting and Getting Name of a thread**

In java, every thread has a name automatically generated by JVM or explicitly provided by the programmer.

We have following methods to set and get name of a thread.

ex:

```
public final void setName(String name)
public final String getName()
```

### **Q) In how many ways we can prevent a thread from execution?**

There are three ways to prevent(stop) a thread from execution.

- 1)yield()
- 2)join()
- 3)sleep()

#### **1)yield()**

It will pause the current execution thread and gives chance to other threads having same priority.

If multiple threads having same priority then we can't expect any execution order.

If there is no waiting threads then same thread will continue the execution.

Ex

```
public static native void yield()
```

#### **2)join()**

If a thread wants to wait until the completion of some other thread then we need to use join() method.

A join() method throws one checked exception called InterruptedException so we must and should handle that exception by using try and catch block or by using throws statement

ex:

```
public final void join()throws InterruptedException
public final void join(long ms)throws InterruptedException
public final void join(long ms,int ns)throws InterruptedException
```

#### **3)SLEEP():**

If a thread don't want to perform any operation on perticular amount of time then we need to use

#### **sleep() method.**

A sleep() method will throw one checked exception called InterruptedException so must and should

handle that exception by using try and catch block or by using throws statement.

```
ex: public static native void sleep()throws InterruptedException
public static native void sleep(long ms)throws InterruptedException
public static native void sleep(long ms,int ns)throws InterruptedException
```

## **Q)Lock Mechanism in Java?**

synchronization is build around an entity called lock.

Whenever a thread wants to access any object.First it has to acquire the lock of it and release the lock once thread complete its task.

## **7)Daemon Thread**

Deamon thread is a service provide thread which provides service to user threads.

The life of deamon thread is same as user threads.Once threads executed deamon thread will terminate automatically.

There are many daemon thread are running in our system.

ex:

Garbage collector , Finalizer and etc.

To start a deamon thread we need to use `setDaemon(true)` method.

To check a thread is a daemon or not we need to use `isDaemon()` method.

## **Problems without synchronization**

If we won't have synchronization then we will face following problems.

- 1) Thread interference.
- 2) Data inconsistency problem.

## **8)synchronization?**

- A synchronized keyword is applicable for methods and blocks.
- A synchronization is allowed one thread to execute given object.Hence we achieve thread safety.
- The main advantage of synchronization is we solve data inconsistence problem.
- The main disadvantage of synchronization is ,it will increase waiting time of a thread which reduce the performance of the system.
- If there is no specific requirement then it is never recommanded to use synchronization concept.
- synchronization internally uses lock mechanism.
- Whenever a thread wants to access object , first it has to acquire lock of an object and thread will release the lock when it completes it's task.

- When a thread wants to execute synchronized method. It automatically gets the lock of an object.
- When one thread is executing synchronized method then other threads are not allowed to execute other synchronized methods in a same object concurrently. But other threads are allowed to execute non-synchronized method concurrently.

### **9) synchronized block?**

If we want to perform synchronization on specific resource of a program then we need to use synchronization.

ex:

If we have 100 lines of code and if we want to perform synchronization only for 10 lines then we need to use synchronized block.

If we keep all the logic in synchronized block then it will act as a synchronized method.

### **10) Static synchronization?**

In static synchronization the lock will be on class but not on object.

If we declare any static method as synchronized then it is called static synchronization method.

### **11) DeadLock in java?**

DeadLock will occur in a situation when one thread is waiting to access object lock which is acquired by another thread and that thread is waiting to access object lock which is acquired by first thread.

Here both the threads are waiting release the thread but no body will release such situation is called DeadLock.

### **12) Drawbacks of multithreading?**

1) DeadLock

2) Thread Starvation

# **JAVA 1.8 FEATURES**

## **1)Functional Interface?**

An interface that contains exactly one abstract method is known as functional interface.

ex:

Runnable -> run()

Comparable -> compareTo()

ActionListener -> actionPerformed()

It can have any number of default and static methods.

Function interface is also known as Single Abstract Method Interface or SAM interface.

It is a new feature in java which helps in to achieve functional programming .

ex:

```
a=f1(){}
f1(f2(){}){}
```

@FunctionalInterface is a annotation which is used to declare functional interface and it is optional.

## **2)Lamda Expression?**

- Lamda Expression introduced in java 1.8v.
- Lamda Expression is used to enable functional programming.
- It is used to concise the code (To reduce the code).
- Lamda Expression can be used when we have functional interface.
- Lamda expression considered as a method not a class.

> Lamda expression does not support modifier,return type and method name.

## **3)Stream API?**

- If we want to process the objects from Collections then we need to use Stream API.
- Stream is an interface which is present in java.util.stream package.
- Stream is use to perform bulk operations on Collections.
- We can create Stream object as follow.

**Syntax:**

```
Stream s=c.stream();
```



# ADVANCE JAVA

---

## JDBC

### **1)What is JDBC ?**

JDBC is a persistence technology which is used to develop persistence logic having the capability to perform persistence operations on persistence data of a persistence store.

### **2)How many steps are there to develop jdbc application?**

There are six steps are there to develop jdbc application.

- 1)Register JDBC driver with DriverManager service.
- 2)Establish the connection with database software.
- 3)Create statement object
- 4)Sends and executes SQL query in database software.
- 5)Gather the result from database software to result.
- 6)Close all jdbc connection objects.

### **3)How many drivers are there in jdbc?**

We have four types of drivers in jdbc?

- 1) Type1 JDBC driver (JDBC-ODBC bridge driver)
- 2) Type2 JDBC driver (Native API)
- 3) Type3 JDBC driver (Net Protocol)
- 4) Type4 JDBC driver (Native Protocol)

#### **4)How many statements are there in JDBC?**

We have three statements in jdbc.

1)Simple Statement

2)PreparedStatement

3)CallableStatement

#### **5)What is DatabaseMetaData?**

DatabaseMetaData is an interface which is present in java.sql package.

DatabaseMetaData provides metadata of a database.

DatabaseMetaData gives information about database product name, database product version, database driver name, database driver version, database username and etc.

We can create DatabaseMetaData object by using getMetaData() method of Connection obj.

ex:

```
DatabaseMetaData dbmd=con.getMetaData();
```

#### **5)What is ResultSetMetaData?**

ResultSetMetaData is an interface which is present in java.sql package.

ResultSetMetaData provides metadata of a table.

ResultSetMetaData gives information about number of columns, datatype of a columns, size of a column and etc.

We can create ResultSetMetaData object by using getMetaData() method of ResultSet obj.

ex:

```
ResultSetMetaData rsmd=rs.getMetaData();
```

## 6)Types of Queries in jdbc?

We have two types of queries in jdbc.

### 1)Select query

It will give bunch of records from database table.

ex:

```
select * from student order by sno;
```

To execute select query we need to executeQuery() method.

### 2)Non-Select query

It will give numeric value represent number of records effected in a database table.

ex:

```
delete from student;
```

To execute non-select query we need to use executeUpdate() method.

## 7)What is JDBC Connection pool?

- It is a factory containing a set of readily available JDBC connection objects before actual being used.
- JDBC Connection pool represent connectivity with same database software.
- A programmer is not responsible to create, manage or destroy JDBC connection objects in jdbc connection pool.
- A jdbc connection pool is responsible to create,manage and destroy jdbc connection objects in jdbc connection pool.

## 8)Types of ResultSet objects?

We have two types of ResultSet objects in jdbc.

1)Non-Scrollable ResultSet object

2)Scrollable ResultSet object

### 1)Non-Scrollable ResultSet object

By default every ResultSet object is a non-scrollable ResultSet object.

It allows us to read the records sequentially or uni-directionally such type of ResultSet object is called non-scrollable ResultSet object.

### 2)Scrollable ResultSet object

It allows us to read the records non-sequentially , bi-directionally or randomly such type of ResultSet object is called scrollable ResultSet object.

## 9)Write a jdbc application to create a table in database?

or

## 9)Write a jdbc application to perform aggregate function?

```
import java.sql.*;

class CreateTableApp
{
    public static void main(String[] args)throws Exception
```

```

{
    Class.forName("oracle.jdbc.driver.OracleDriver");

    Connecton con=DriverManager.getConnection
("jdbc:oracle:thin:@localhost:1521:XE","system","admin");

    String qry="create table student(sno number(3),
                                sname varchar2(10),sadd varchar2(12))";

    PreparedStatement ps=con.prepareStatement(qry);

    int result=ps.executeUpdate();
    if(result==0)
        System.out.println("Table not created");
    else
        System.out.println("Table created");
    ps.close();
    con.close();
}
}

```

### 9)Write a jdbc application to insert the record into student table?

```

import java.sql.*;

import java.util.*;

```

```
class CreateTableApp
{
    public static void main(String[] args)throws Exception
    {
        Scanner sc=new Scanner(System.in);

        System.out.println("Enter the student no: ");

        int no=sc.nextInt();

        System.out.println("Enter the student name :");

        String name=sc.next();

        System.out.println("Enter the student address :");

        String add=sc.next();

        Class.forName("oracle.jdbc.driver.OracleDriver");

        Connecton
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","admin");

        String qry="insert into student values(?,?,?)";

        PreparedStatement ps=con.prepareStatement(qry);

        //set the values

        ps.setInt(1,no);

        ps.setString(2,name);
```

```
        ps.setString(3,add);

        int result=ps.executeUpdate();

        if(result==0)

            System.out.println("Table not created");

        else

            System.out.println("Table created");

        ps.close();

        con.close();

    }

}
```

### **Persistence:**

The process of storing and managing the data for a long period of time is called persistence.

### **Persistence store:**

It is a place where we can store and manage the data for a long period of time.

### **Persistence data :**

Data of a persistence store is called persistence data.

### **Persistence operation:**

Insert, Update, Delete, Create and Select are called persistence operations.

In the real-time this operation is also known as CURD operation or CRUD operation or SCUD operation.

### **Persistence logic:**

Persistence logic having the capability to perform persistence operations.

### **persistence technology :**

Persistence technology is used to develop persistence logics.

## **Serialization:**

---

The process of changing object state to file state is called serialization.  
In serialization object will not store in a file , object data will store in a file.

## **Deserialization:**

The process of taking the data from file and representing an object is called deserialization.

## **Class.forName():**

It is used to register JDBC driver with DriverManager service.  
This method is used to load the driver class but it won't create an object.

## **Connection object:**

A Connection is an interface which is present in java.sql package.  
It is an object of underlying supplied java class which implements java.sql.Connection interface.

To perform any operation on database we need to establish the connection.  
Once work with database software is completed then we need to close the connection.

## **DriverManager.getConnection():**

DriverManager is a class which is present in java.sql package.

A getConnection() is a static method which returns one JDBC connection object represent connectivity between java application and database software.

## **Statement object :**

A Statement is an interface which is present in java.sql package.  
It is used to sends and executes our SQL query in database software.  
We can create Statement object by using createStatement() method of Connection object.

## **Types of Connection objects?**

We have two types of Connection objects.

### *1) Direct Connection object:*

A connection object which is created by the user is called direct direct connection object.

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

```
Connection
```



```
con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:XE","system","admin");
```

## *2) Pooled JDBC Connection object:*

A connection object which is gathered from JDBC Connection pool is called pooled jdbc connectionobject.

### **Limitations with Simple Statement object:**

- > It is not suitable to execute same query for multiple times with same or different values.
- > It raises SQL injection problem.
- > Framing query with variables is quite complex.
- > We can't use string values directly in a query.
- > It does not allow us to insert date values in a database table column.
- > It does not allow us to insert LOB values in a database table column.
- > To overcome these limitations we need to use PreparedStatement object.

### **JDBC Flexible Application:**

In JDBC, Connection object is a heavy weight object.

It is never recommended to create Connection object in every JDBC application because it will increase the burden of an application over the network.

It is always recommended to have a separate class which returns Connection object.

### **Types of statements objects in jdbc:**

#### *Simple Statement Object:*

It is an object of underlying supplied java class which implements `java.sql.Statement()`.

#### *PreparedStatement Object:*

It is an object of underlying supplied java class which implements

`java.sql.PreparedStatement()`. PreparedStatement obj gives pre-compiled sql query.

Using this obj we can perform.

- We can set values to pre-compiled sql query for multiple times.
- We can execute pre compiled sql query for multiple times.
- We can gather the result from pre compiled sql query for multiple times.

#### *CallableStatement Object:*

It is an object of underlying supplied java class which implements `java.sql.CallableStatement()`.

# SERVLET

---

## 1)Q)What is Servlet ?

Servlet is a dynamic web resource program which enhanced the functionality of web server , proxy server ,HTTP server and etc.

or

Servlet is a java based dynamic web resource program which is used to generate dynamic web pages.

or

Servlet is a single instance multi-thread java based web resource program which is used to develop web applications.

## 2)What is web application?

Web application is a collection of web resource programs having the capability to generate web pages.

We have two types of web pages.

- 1)Static web page
- 2)Dynamic web page

## 3)What is web resource program?

We have two types of web resource programs.

- 1)Static web resource progra

It is responsible to generate static web pages.

ex:

- HTML program
- CSS program
- Bootstrap program
- Angular program
- and etc.

- 2)Dynamic web resource program

It is responsible to generate dynamic web pages.

ex:

Servlet program  
JSP program  
and etc.

#### **4) What is Web container**

- It is a software application or program which is used to manage whole life cycle of web resource program i.e from birth to death.
- Servlet container manages whole life cycle of servlet program.
- Similarly , JSP container manages whole life cycle of jsp program.
- Some part of industry considers servlet container and jsp container are web containers.

#### **5) What is ServletConfig object?**

- ServletConfig is an interface which is present in javax.servlet package.
- ServletConfig object will be created by the web container for every servlet.
- ServletConfig object is used to read configuration information from web.xml file
- We can create ServletConfig object by using getServletConfig() method.

ex:

```
ServletConfig config=getServletConfig();
```

#### **6)What is ServletContext object?**

- ServletContext is an interface which is present in javax.servlet package.
- ServletContext object is created by the web container for every web application i.e it is one per web application.
- ServletContext object is used to read configuration information from web.xml file and it is for all servlets.
- We can create ServletContext object by using getServletContext() method.

ex:

```
ServletContext context=getServletContext();
```

```
Or ServletConfig config=getServletConfig();  
ServletContext context=config.getServletContext();
```

## 7) what is Servlet Filters?

Filter is an object which is executed at the time of preprocessing and postprocessing of the request.

The main advantages of using filter is to perform filter task such as

- 1) Counting number of request
  - 2) To perform validation
  - 3) Encryption and Decryption
- and etc.

Like Servlet, Filter is having it's own Filter API.

The javax.servlet package contains three interfaces of Filter API.

- 1)Filter
- 2)FilterChain
- 3)FilterConfig

## 8)Differences between GET And POST methodology?

### GET

It is a default methodology.

It sends the request fastly.

It will carry limited amount of data.

It is not suitable for secure data.

It is not suitable to perform encryption or fileuploading.

To process GET methodology we need to use doGet(-,-) method.

### Post

It is not a default methodology.

It sends the request bit slow

It will carry unlimited amount of data.

It is suitable for secure data.

It is suitable to perform encryption and file uploading.

To process POST methodology we need to use doPost(-,-) method.

## 9)Explain Servlet Life cycle methods?

We have three life cycle methods in Servlets

### 1)public void init(ServletConfig config)throws ServletException

It is used for instantiation event.

This method will execute just before Servlet object creation.

### 2)public void service(ServletRequest req,ServletResponse res)throws ServletException,IOException

It is used for request processing event.

This method will execute when request goes to servlet program.

### **3)public void destroy**

It is used for destruction event.

This method will execute just before Servlet object destruction.

### **10)Limitations with servlets?**

>To work with servlets strong java knowledge is required.

> It is suitable for only java programmers.

> Configuration of servlet program in web.xml file is mandatory.

> Handling exceptions are mandatory.

> It does not give any implicit object.

(Object which can be used directly without any configuration is called implicit object).

> We can't maintain html code and java code sperately.

> It does not support tag based language.

### **What is deployment?**

The process of keeping the application in the server is called deployment and reverse is called undeployment.

### **What is web server ?**

A server is a piece of software which is used to automate whole process of web application and webresource program execution.

### **Responsibilities of a web server:**

- 1) It takes continues request from client.
- 2) It traps the request and pass that request to appropriate web resource program.
- 3) It provides environment to deploy and undeployed the web application.
- 4) It will add middleware services only to deployed web application.
- 5) It provides environment to execute client side web resource programs at browser window.
- 6) It sends the output on a browser window as dynamic response.
- 7) It is used to automate whole process of web application and web resource

programexecution.

### **Types of URL patterns:**

URL Pattern will hide technology name or class name from the outsider for security reason.

Our web container, web resource program and client will recognize each servlet by using url patternonly.

Every server is designed to support three types of url patterns.

We have three types of URL patterns.

1)Exact match url pattern

2)Directory match url

pattern3)Extension match

url pattern

### **1) Exact match url pattern**

This url pattern starts with '/' forward slash and it ends with any single name.

### *Types of Communications:*

We can communicate to servlet program in three ways.

**1)Browser to Servlet communication**

**2)HtmltoServletcommunication**

**3)ServlettoServletcommunication**

In Browser to Servlet communication we need to type our request url in browser address bar. But typing request url in browser address bar is complex.

To overcome this limitation we need to use HTML to Servlet communication.

In html to servlet communication we can generate the request by using html based hyperlinks or form pages.

The request which is generated by using hyperlink does not carry the data.

The request which is generated by using form page will carry the data.

In html based hyperlink to servlet communication we need to type our request url as

href url.ex: `<a href="http://localhost:2525/DateApp/test"> clickMe </a>`

In html based form page to servelt communication we need to type our request url

asaction url.

## **Form Validation:**

The process of checking format and pattern of form data is called form validation. Such logic is called form validation logic.

There are two ways to perform form validation.

1) Client Side Form Validation

2) Server Side Form Validation

### **1) Client Side Form Validation:**

A validation which is performed at client side is called client side form

validation. To perform client side form validation we need to use javascript.

### **2) Server Side Form Validation:**

A validation which is performed at server side is called server side form

validation. To perform server side form validation we need to use servlets.

## **Limitations with servlets**

- To work with servlet strong java knowledge is required.
- It is not suitable for non-java programmers.
- Handling exceptions are mandatory.
- Configuration of each servlet program in web.xml file is mandatory.
- It does not give any implicit object. (Object which can be used without any configuration is called implicit object)
- We can't maintain html code and java code separately.



# JSP

---

## 1)Advantages of JSP ?

- > To work with JSP strong java knowledge is not required.
- > It is suitable for java and non-java programs.
- > It supports tag based language.
- > It allows us to create custom tags in jsp.
- > Configuration of jsp program in web.xml file is optional.
- > Handling exceptions are optional.
- > It gives 9 implicit objects.
- > We can maintain html code and java code sperately.
- > It gives all the features of servlets.

## 2)What is jsp?

JavaServer Pages (JSP) is a technology for developing Webpages that supports dynamic content. This helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

## 3) Types of Jsp Implicit Objects?

There are 9 implicit objects present in jsp.

Implicit objects are created by the web container that is available for every jsp program.

Object which can be used directly without any configuration is called implicit object.

The list of implicit objects are.

Object	Type
out	JspWriter
request	HttpServletRequest
response	HttpServletResponse
config	ServletConfig
application	ServletContext
session	HttpSession
pageContext	pageContext
page	Object
exception	Throwable

## 4) Types of JSP Tags/Elements?

### 1) Scripting tags

#### i) Scriptlet tag

ex:

```
<%      code      %>
```

#### ii) Expression tag

ex:

```
<%=      code      %>
```

#### iii) Declaration tag

ex:

```
<%!      code      %>
```

### 2) Directive Tags

#### i) Page directive tag

ex:

```
<%@page  attribute=value %>
```

#### ii) include directive tag

ex:

```
<%@include  attribute=value %>
```

### 3) Standard Tags

```
<jsp:include>
```

```
<jsp:forward>
```

```
<jsp:setProperty>
```

```
<jsp:getProperty>
```

```
<jsp:useBean>
```

and etc.

### 4) JSP comment

```
<%--      jsp comment      --%>
```

#### i) Scriptlet tag

It is used to declare java code.

syntax:

```
<%      code      %>
```

## 5) JSP life cycle methods

JSP contains three life cycle methods.

### 1) \_jspInit()

It is used for instantiation event

This method will execute just before JES class object creation.

JES class stands for Java Equivalent Servlet class.

## 2)\_jspService()

It is used for request arrival event

This method will execute when request goes to JSP program.

## 3)\_jspDestroy()

It is used for destruction event.

This method will execute just before JES class object destruction

## 6)Phases in JSP

In JSP , we have two phases.

### 1)Translation phase

In this phase, our JSP program converts to JES class

(ABC\_jsp.class and ABC\_jsp.java) object.

### 2)Request processing phase

In this phase, our JES class will be execute and result will send to browser window as dynamic response.

## 7)MVC in JSP?

MVC stands for Model View Controller.

It is a design partern which seperates business logic , persistence logic and data.

Controller acts like an interface between Model and View.

Controller is used to intercept with all the incoming requests.

Model contains data.

View represent User interface i.e UI.

## 9)What is JES class?

## Exception Handling in JSP

There may chance of raising exception anywhere in our application so handling the exception is saferside for the programmer. Runtime errors are called exceptions.

Exceptions always raised at runtime so they are also known as runtime

events.In jsp , exception handling can be performed in two ways.

1)By using errorPage and isErrorPage attribute of page directive

tag.2)By using element in web.xml file.

Action tags provides functionality along with servlet API features. It contains xml tags. It does

not have any standard tags. All action tags are executed dynamically at runtime. Action tags are divided into two types

1) Standard action tags            2) Custom action tags

### *1) Standard action tag*

Built-In action tags are called standard action tags.

#### *Response:*

In jsp, response is an implicit object of type `HttpServletResponse`. It can be used to add or manipulate response such as redirect response or another resources, send error and etc.

### *JSP config*

It is an implicit object of type `ServletConfig`. The config object is created by web container for each jsp page. This object is used to read initialized parameters for a particular jsp page.

# **HTML-5**

## **What is HTML?**

HTML stands for Hypertext Markup Language. HTML was developed by "Burners Lee" in late, 1991.

HTML is widely used language on web to develop web pages and web applications. HTML element represented by tags so it is also known as tag based language.

HTML is case insensitive language.

All HTML documents must save with ".html" and ".htm" extension.

## **Advantages of HTML?**

HTML is easy to learn and easy to use

It is free to use.

HTML is supported by all browsers

No special software needed to run HTML.

It is platform independent.

HTML can integrate easy with other languages

It supports powerful formatting facilities.

It allows us to add Graphics, Videos, and audios in web page.

Finding Errors are easy.

## **Disadvantages of HTML?**

HTML is used to create Static web pages but not Dynamic web pages.

Need to write lot of code for making simple web page.

Security features are not good in HTML.

If code is increases then it produces some complexity.

## **What is <meta> tag in HTML?**

The <meta> tag defines metadata about an HTML document. The <meta> tag must be used inside <head> tag.

The <meta> tag/element are used to specify

Page description

Keywords

Author

Portview and

Other metadata

## **What is void element in HTML?**

A void element cannot have any content but may have attributes. Void elements are self-closing, so they must not have a closing tag. The following lists of void elements are

Ex:

<br>

<hr>

<input>

<link> and etc.

## **Difference between HTML tag, HTML element and HTML attribute?**

### **HTML tag:**

HTML tag starts with '<' and ends with '>'.Ex:

`<html>,<head>,<body>` and etc.

### **HTML element:**

An HTML element is defined by a start tag, some content and end tag.Ex:

`<h1>This is Heading Tag</h1>`

### **HTML attribute:**

Attributes provide additional information about elements.Attributes are always specified in the start tag.

Attributes usually come in name/value pairs.Ex:

`<body bgcolor="#FFFF00">`

## **Types of Tags in HTML?**

We have two types of HTML tags.

### **Container Tag/ Paired Tag:**

It contains an opening tag and a closing tag.Ex:

`<h1>,<p>,<b>,<i>` and etc.

### **Empty Tag/ Unpaired Tag:**

It contains only an opening tag and does not need a closing tag.Ex:

`<br>,<hr>,<meta>,<link>` and etc.

## **Types of Elements in HTML?**

We have two types of HTML elements.

### **Block Elements:**

Block elements always start with a new line and occupies completed width of aviewport /device.

ex:

`<p>,<h1> to <h6>,<div>,<address>,<ol>,<ul>` and etc.

### Inline Elements:

Inline elements start with same line and occupy width as much as required.ex:

<i>, <b>, <span>, <u>, <strong>, and etc.



## What is HTML entity?

An HTML entity is a piece of text ("string") that begins with an ampersand (&) and ends with a semicolon (;). Entities are frequently used to display reserved characters, and invisible characters.

Ex:

Character	Entity
&	&amp;
>	&gt;
<	&lt;
"	&quot;

## Types of list in HTML?

We have three types of list in HTML

### Order list:

It is used to group a set of related items in a specific order.Ex:

```
<ol>  
  
    <li>HTML</li>  
  
    <li>CSS</li>  
  
    <li>JAVASCRIPT</li>
```

### Unorder list:

```
</ol>
```

It is used to group a set of related items in no particular order.Ex:

```
<ul>  
  
    <li>HTML</li>  
  
    <li>CSS</li>  
  
    <li>JAVASCRIPT</li>  
  
</ul>
```

Description list:

It is used to display name/value pairs such as terms and definitions.Ex:

<dl>

<dt>HTML</dt>

<dd>HTML is widely used language on web.</dd>

</dl>

## What is DHTML?

DHTML stands for Dynamic Hypertext Markup language.

DHTML is a combination of HTML, CSS and JavaScript and etc. which are used to create dynamically changing websites.

Difference between `<b>` tag and `<strong>` tag?

The HTML `<b>` element defines bold text, without any extra importance. Ex:

`<b>`This text is bold`</b>`

The HTML `<strong>` element defines text with strong importance. The content inside is typically displayed in bold.

Ex:

`<strong>`This text is important`</strong>`

## Difference between `<i>` tag and `<em>` tag?

The HTML `<i>` element defines a part of text in an alternate voice or mood. The content inside is typically displayed in italic.

Ex:

`<i>`This is italic tag`</i>`

The HTML `<em>` element defines emphasized text. The content inside is typically displayed in italic.

Ex:

`<em>`This text is emphasized`</em>`

List out some Formatting tags/elements in HTML?

Formatting elements were designed to display special types of text. Ex:

`<b>` - Bold text

`<strong>` - Important text

`<i>` - Italic text

`<mark>` - Marked text

<code>&lt;small&gt;</code>	-	Smaller text
<code>&lt;del&gt;</code>	-	Deleted text
<code>&lt;ins&gt;</code>	-	Inserted text
<code>&lt;sub&gt;</code>	-	Subscript text
<code>&lt;sup&gt;</code>	-	Superscript text

## List out some phrasing tags in HTML?

The phrase tags are special purpose tags which define structural meaning to the block of a content/text.

Ex:

	Abbreviation	--	<code>&lt;abbr&gt;</code>
	Definition	--	<code>&lt;dfn&gt;</code>
	Short quote	--	<code>&lt;q&gt;</code>
code	--	<code>&lt;code&gt;</code>	
keyboard	--	<code>&lt;kbd&gt;</code>	
address	--	<code>&lt;address&gt;</code>	

List out some HTML layout elements?

<code>&lt;header&gt;</code>	- Defines a header for a document or a section
<code>&lt;nav&gt;</code>	- Defines a set of navigation links
<code>&lt;section&gt;</code>	- Defines a section in a document
<code>&lt;article&gt;</code>	- Defines an independent, self-contained content
<code>&lt;aside&gt;</code>	- Defines content aside from the content (like a sidebar)
<code>&lt;footer&gt;</code>	- Defines a footer for a document or a section
<code>&lt;details&gt;</code>	- Defines additional details that the user can open and close on demand
<code>&lt;summary&gt;</code>	- Defines a heading for the <code>&lt;details&gt;</code> element

## How to display another document/webpage in current HTML document?

An inline frame is used to embed another document within the current HTML document.Ex:

```
<iframe src="demo.htm" height="200" width="300" title="Iframe Sample"></iframe>
```

## Difference between <div> and <span> tag?

Div	span
It is block level element.	It is inline element.
It is used to wrap sections of a document.	It is used to wrap small portion of text, images and etc.
It is used to create CSS based layouts.	It is used to stylize texts.

## What is Web Application?

It is a collection of web resource programs having a capability to generate web pages.Ex:

Facebook whatsapp webGmail and etc.

## What is web resource program?

A web resource program is used to create web pages.We have two types of web resource programs.

### Client Side web resource program:

A web resource program which executes at client side(Browser) is called clientside web resource program.

Ex:

HTML program CSS program Javascript programBootstrap programand etc.

### Server Side web resource program

A web resource program which executes at server side is called serverside web resource program.

Ex:

java program

.net program python program php program

nodejs program and etc.

## **What is web page?**

A web page is a hypertext markup language which is embedded with World Wide Web(www). Each web page we can access via protocol i.e. TCP/IP, HTTP, HTTPS, SMTP, POP and etc.

We have two types of web pages.

### **Static web page:**

A static web page is a page that written in HTML, CSS, JavaScript, Bootstrap and etc.

It displays the content exactly as stored or represent. Ex:

Home page Aboutus page ContactUs page

Services page and etc.

### **Dynamic web page:**

A dynamic web page is a page whose construction is controlled by an application server.

It displays different content each time. Ex:

Facebook page Instagram page

live cricket score page

stock market share value page and etc.

## **What is Website?**

Website is a collection of web pages.

Website resides and executes in internet environment and it will give 24/7 accessibility for our web pages.

Ex:

[www.ihubtalent.com](http://www.ihubtalent.com) [www.qualitythought.in](http://www.qualitythought.in) [www.amazon.com](http://www.amazon.com)

[www.flipkart.com](http://www.flipkart.com)

## List out some Tags introduced in HTML5?

The following tags introduced in HTML5 areEx:

<header>	<footer>	<section>	<article>	<aside>
<nav>	<audio>	<video>	<command>	<datalist>
<figure>	<hgroup>	<mark>	<meter>	<summary>
<progress>	<output>	<ruby>	<time>	and etc.

List out some Tags/Elements removed from HTML5? The following tags/elements removed from HTML5 are Ex:

<big>	<center>	<font>	<frame>	<frameset>
<noframes>	<s>	<strike>	<u>	<dir>
<applet>	<basefont>	and etc.		

## In how many ways we can display graphics in HTML?

There are two ways to draw the graphics in html.

### SVG:

SVG stands for "Scalable Vector Graphics".

Using SVG we can derive 2 dimensional vector based graphics on a web page. The <svg> tag/element is a container for SVG graphics.

### CANVAS:

A <canvas> tag/element is used to draw the graphics via JavaScript. A <canvas> tag/element is a container for html graphics.

Explain some media tags of HTML5?

<audio> : It is an inline element that is used to embed sound files into a web page.

<video>: It is used to embed video files into a web page.

`<source>` : It is used to attach multimedia files like audio, video, and pictures.

`<track>` : It specifies text tracks for media components audio and video.

`<embed>` : It is used for embedding external applications which are generally multimedia content like audio or video into an HTML document.

## **What is datalist tag in HTML5?**

The `<datalist>` tag specifies a list of pre-defined options for an `<input>` element.

The `<datalist>` tag is used to provide an "autocomplete" feature for `<input>` elements. Users will see a drop-down list of pre-defined options as they input data.

Ex:

```
<input type="text" list="browsers">
```

```
<datalist id="browsers">
```

```
<option value="Chrome">
```

```
<option value="firefox">
```

```
<option value="Edge">
```

```
</datalist>
```

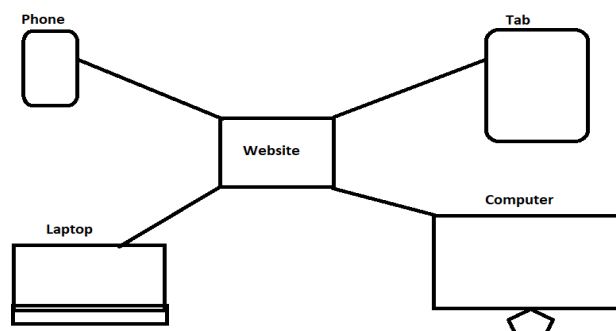


## Differences between HTML and HTML5?

HTML	HTML5
To represent HTML document we need to use <code>&lt;!DOCTYPE&gt;</code> .	To represent HTML5 document we need to use <code>&lt;!DOCTYPE html&gt;</code> .
HTML is bit slow.	HTML5 is faster because it is light weight.
HTML is not efficient.	HTML5 is more efficient.
HTML is inflexible for the developer.	HTML5 is flexible for the developer.
HTML is Less mobile friendly.	HTML5 is mobile friendly.
Does not support drag and drop effects.	It supports Drag and Drop effects.
Not supported Audio and video without using flash player.	Supports audio and video with the help of <code>&lt;audio&gt;</code> and <code>&lt;video&gt;</code> tag without using flashplayer.
IT does not support JavaScript to run on browser.	It supports JavaScript to run in background with the help of JS Web Worker API.
Vector graphics supported with the help of technologies such as VML, Silver light, adobeflash and etc.	Vector graphics is supported with the help of internal technologies such as SVG and CANVAS.
Can't handle inaccurate syntax.	Can't handle inaccurate syntax.
Shapes like circle, rectangle, and triangle are not possible.	Shapes like circle, triangle, and rectangle are easy to draw.

## What is responsive web design?

Responsive web design is about creating web pages which automatically adjust the content for difference



screens and viewports.

# CSS

---

## Interview Questions

### 1) What is CSS?

CSS stands for Cascading Style Sheets.

It is widely used language on web like HTML.

It is extremely used to apply the styles on HTML elements.

In general, CSS is used to design the HTML elements/tags.

The latest version of CSS3 was introduced in 2001.

All CSS files we need to save with ".css" extension.

### 2) Advantages of CSS?

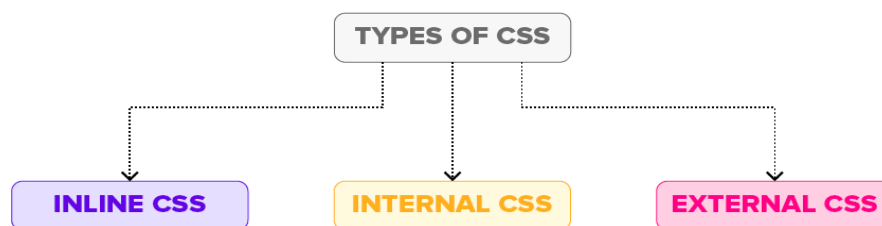
- Flexibility
- Saves lot of time
- Performance is faster
- Support Global Change
- Compatibility with multiple devices

### 3) Disadvantages of CSS?

- Fragmentation
- Need to update all the versions of CSS
- There exists a scarcity of security
- Browser compatibility

### 4) Different types of CSS?

We have three types of CSS.



#### Inline CSS:

Inline CSS may use to apply a unique style for a single Element/Tag.

We can achieve inline CSS with the help of "style" attribute.

The "style" attribute may contain any CSS property.

Ex:

```
<h1 style="color:red">Heading Tag</h1>
<p style="background-color:yellow"></p>
<b style="font-size:30px"></b>
```

### Internal CSS or Embedded CSS:

Internal CSS may use when a single HTML document must be styled uniquely.

Using <style> tag we can achieve internal CSS.

A <style> tag must be placed inside <head> tag.

Ex:

```
<!DOCTYPE html>
<html>
  <head>
    <title>IHUB TALENT</title>
    <style type="text/css">
      body
      {
        background-color:red;
      }
    </style>
  </head>
  <body>
    Welcome to IHUB TALENT MANAGEMENT
  </body>
</html>
```

### External CSS:

External CSS may use to change the look of the entire website by changing just one file.

Each html pages includes a reference to the external style sheet inside the <link> tag.

In external css we have two create two files. i.e ".html" and ".css" file.

The external ".css" file should not contain any html tags.

Ex:

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>IHUB Talent</title>
    <!-- adding external css file -->
    <link rel="stylesheet" type="text/css" href="mystyles.css">
  </head>
  <body>
    Welcome to IHUB TALENT MANAGEMENT
  </body>
</html>
mystyles.css
body
{
  background-color:violet;
}
```

## 5) Limitations of CSS?

CSS has various limitations as a programming language as follows

- ✓ CSS cannot perform any logical operations like if/else or for/while or +/-.
- ✓ We can't read any files using CSS.
- ✓ It can't interact with databases.
- ✓ CSS can't request a web page.

## 6) What is CSS framework?

A CSS framework comprises several CSS style sheets ready for use by web developers and designers. The framework already has built-in classes for common website elements – footer, slider, navigation bar, hamburger menu, column-based layouts, etc.

We have following list of CSS frameworks.

Ex:

Bootstrap, Tailwind CSS, Bulma, Skeleton, Foundation and etc.

## 7) How to declare units in CSS?

A CSS unit is used to determine the property size, which we set for an element or its content. Units are used to set margin, padding, lengths, and so on.

Unit	Name	Explanation
cm	Centimeters	It is used to define the measurement in centimeters.
mm	Millimeters	It is used to define the measurement in millimeters.
in	Inches	It is used to define the measurement in inches. 1in = 96px = 2.54cm
pt	Points	It is used to define the measurement in points. 1pt = 1/72 of 1 inch.
pc	Picas	It is used to define the measurement in picas. 1pc = 12pt so, there 6 picas is equivalent to 1 inch.
px	Pixels	It is used to define the measurement in pixels. 1px = 1/96th of inch

Ex:

```
<p style = "font-size: 20px;" > It has a font-size: 20px; </p>
<p style = "font-size: 1.2cm;" > It has a font-size: 1.2cm; </p>
<p style = "font-size: .7in;" > It has a font-size: .7in; </p>
<p style = "font-size: 18pt;" > It has a font-size: 18pt; </p>
<p style = "font-size: 2pc;" > It has a font-size: 2pc; </p>
<p style = "font-size: 10mm;" > It has a font-size: 10mm; </p>
```

## 8) How to include CSS in a web page?

There are three ways to include CSS in a web page

- Inline CSS: We can achieve inline CSS using style attribute.
- Internal CSS: We can achieve internal CSS using <style> tag.
- External CSS: We can achieve external CSS using <link> tag.

## 9) What are CSS selectors and Types of selectors in CSS?

CSS selectors are used to find/select the elements to which we want to apply styles.

We have five types of CSS selectors.

- **CSS element selector:**

- The element selector selects HTML elements based on element name.

Ex:

```
P
{
    text-align:center;
    color:red;
}
```

- **CSS id selector:**

- The id selector uses the id attribute of an HTML element to select a specific element.
- The id of an element is unique within a page, so the id selector is used to select one unique element.
- To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Ex:

```
#para1
{
    text-align: center;
    color: red;
}
<p id="para1">This is paragraph tag </p>
```

- **CSS class selector:**

- The class selector selects HTML elements with a specific class attribute.
- To select elements with a specific class, write a period (.) character, followed by the class name.

Ex:

```
.center
{
    text-align: center;
    color: red;
}
<p class="center">This is paragraph</p>
```

- **CSS grouping selector:**

The grouping selector selects all the HTML elements with the same style definitions.

Ex:

```
h1, h2, p
{
    text-align: center;
    color: red;
}
<p>This is paragraph</p>
<h1>This is Heading1 Tag</h1>
<h2>This is Heading2 Tag</h2>
```

- **CSS universal selector:**

The universal selector (\*) selects all HTML elements on the page.

Ex:

```
*
{
    text-align: center;
    color: blue;
}
<p>This is Paragraph</p>
<h1>This is heading1 tag</h1>
```

## 10) Difference between id selector and class selector?

id selector	class selector
Using '#' symbol we can create id selector.	Using dot(.) symbol we can create class selector.
The id selector uses the "id" attribute of an HTML element to select a specific element.	The class selector uses the "class" attribute of an HTML element to select a specific element.
We can't use multiple ids on elements.	We can use multiple classes on elements.
A id selector is used for single tags.	A class selector is used for multiple tags.

## 11) Difference between inline, block and inline-block?

- **inline :**

We can't define height and width.

It automatically adjusts its height and width based on the content.

- **block:**

We can define height and width.

Block elements always starts on a new line.

By default it will occupy complete width of a window/viewport.

- **inline-block:**

Like block elements, we can define height and width.  
It does not start on a new line.

## 12) Which property is used to change the font face?

A “font-family” property is used to change the font face.

Ex:

```
<p style="font-family:georgia,garamond,serif;"> This is Paragraph </p>
```

## 13) Difference between content-box and border-box?

**content-box:** Box-sizing content-box consider width equals to content width. Adding border and padding can increase size of box.

Ex:

```
.box
{
    width:320px;
    height:200px;
    background:#ccc;
    padding:10px;
    box-sizing: content-box
}
<div class="box"></div>
```

**border-box:** Box-sizing border-box consider width equals to total border box. Adding border and padding will not change size of box.

Ex:

```
.box
{
    width:320px;
    height:200px;
    background:#ccc;
    padding:10px;
    box-sizing: border-box
}
<div class="box"></div>
```

## 14) Explain float property in CSS?

The float property specifies whether an element should float to the left or right or not at all.

Ex:

```
float: none|left|right|initial|inherit;
```

### 15) Describe z-index in CSS?

- The z-index property specifies the stack order of an element.
- An element with greater stack order is always in front of an element with a lower stack order.
- A z-index only works on positioned elements and flex items.

Ex:

```
img
{
    position: absolute;
    left: 0px;
    top: 0px;
    z-index: -1;
}
<h1>This is a heading</h1>

<p>Image has a z-index of -1, it will be placed behind the text.</p>
```

### 16) Explain CSS position property?

The position property specifies the type of positioning method used for an element (static, relative, absolute, fixed, or sticky).

Ex:

```
position: static | absolute | fixed | relative | sticky | initial | inherit;
```

### 17) Difference between CSS Grid and Flexbox?

GRID	FLEXBOX
Grid is made for two-dimensional layout	Grid is made for two-dimensional layout
Grid can work on rows and columns at a time.	Flexbox can work on either rows or columns at a time.
Grid helps you create the outer layout of the webpage.	Flexbox mostly helps align content & move blocks.

### 18) Explain transition property?

CSS transitions allow you to change property values smoothly, over a given duration. We have following list of transition properties.

Ex:

```
transition
transition-delay
transition-duration
transition-property
transition-timing-function
```

We can see transition effects, if we use mouse over (hover) property.



## 19) Difference between margin property and padding property?

**CSS margins:** It is used to create space around the element. We can set the different sizes of margins for individual sides (top, right, bottom, left).

**CSS padding:** It is used to create space around the element, inside any defined border. We can set different paddings for individual sides (top, right, bottom, left). It is important to add border properties to implement padding properties.

## 20) Difference between visibility:hidden and display:none?

**visibility:hidden-** It is not visible but gets up its original space.

Ex:

```
h1
{
    visibility: hidden;
}
```

**display:none-** It is hidden and takes no space.

Ex:

```
h1
{
    display:none;
}
```

## 21) How to set background image in CSS?

We can use following properties to define background image in css.

Ex:

```
body
{
    background-color: #FFF;
    background-image: url('images/bg2.jpg');
    background-repeat: no-repeat;
    background-size: cover;
    background-position: center 0px;
    background-attachment: fixed;
}
```

//or

We can use shorthand property to define background image in css.

Ex:

```
body
{
    Background: #fff url('images/bg2.jpg') no-repeat center 0px fixed;
}
```

## 22) Explain about CSS gradients?

CSS gradients let you display smooth transitions between two or more specified colors. CSS defines three types of gradients

- Linear Gradients (goes down/up/left/right/diagonally)

Ex:

```
background-image: linear-gradient(red, yellow);
```

- Radial Gradients (defined by their center)

Ex:

```
background-image: radial-gradient(red, yellow, green);
```

- Conic Gradients (rotated around a center point)

Ex:

```
background-image: conic-gradient(red, yellow, green);
```

## 23) Difference between px, em and %?

Pixel is a static measurement, while percent and EM are relative measurements.

Ex:

<u>px</u>	<u>em</u>	<u>percent</u>
25px	1.5625em	156.25%

## 24) In how many ways we can define colors in CSS?

There are following ways to define colors in CSS.

- ✓ With color name like red.
- ✓ With HEX value like #FFFF00.
- ✓ With RGB value like rgb(255,0,0).
- ✓ With HSL value like hsl(0,100%,50%).

## 25) Explain about transform property in CSS?

The transform property applies a 2D or 3D transformation to an element. This property allows you to rotate, scale, move, skew, etc., elements.

Ex:

```
transform: none|transform-functions|initial|inherit;
```

## 26) Explain about clear property in CSS?

- The clear property controls the flow next to floated elements.
- The clear property specifies what should happen with the element that is next to a floating element.

## 27) Explain about cursor property in CSS?

The cursor property specifies the mouse cursor to be displayed when pointing over an element.

Ex:

```
p:hover {cursor: pointer};
```

# JavaScript

---

## 1) What is JavaScript?

JavaScript is a scripting language.  
It is a case sensitive language.  
It is an object based language.  
It is a loosely typed checking language.  
It was developed by Brendan Eich in 1995.  
The original name of JavaScript is LiveScript.

## 2) Advantages of JavaScript?

Speed/Faster  
Simplicity  
Interoperability  
Versatility  
Rich interfaces  
Reduce Server Load  
No compiler and No interpreter  
Weakly typed language  
Platform independent  
Client Side validation

## 3) Disadvantages of JavaScript?

Client-Side Security  
Browser Support  
Stop Rendering  
Slow Bitwise Operation  
Single Inheritance

## 4) Difference between Java and JavaScript?

Java	JavaScript
It is a non-scripting language.	It is a scripting language.
Java can run individually.	JavaScript can't run individually.
Java does not required browser window for execution	JavaScript requires browser window for execution.
It is an object oriented programming language.	It is an object based programming language.
It is a strongly typed checking language.	It is a loosely typed checking language
It is complex language.	It is easy language.

## 5) What is JavaScript Engine?

JavaScript Engine is software or a program which converts user understandable scripting language to machine understandable scripting language.

By default, every browser contains JavaScript engine.

The following lists of JavaScript engine present in a browser are.

Ex:

<u>Browser</u>	<u>JavaScript Engine</u>
1) Chrome	V8 Engine
2) Firefox	Spidermonkey
3) Safari	javascriptcore
4) Edge	Chakra

## 6) Types of JavaScript?

We have two type of JavaScript.

- **Internal JavaScript / Embedded JavaScript**

It contains html code and JavaScript code inside a single file with ".html" extension.

- **External JavaScript / External JavaScript**

It contains html code in ".html" file and JavaScript code in ".js" file.

## 7) Types of functions in JavaScript?

We have two types of functions.

- **Named Functions**

These types of functions contains name at the time of definition.

Ex:

```
function f1()
{
    document.writeln("hello world");
}
f1();
```

- **Anonymous Functions**

These types of functions don't contain any name.

They are declared dynamically at runtime.

Ex:

```
var f1=function(){
    document.writeln('hello world');
}
f1();
```

## 8) What is JavaScript Closure?

A closure is the combination of a function bundled together along lexical scope.

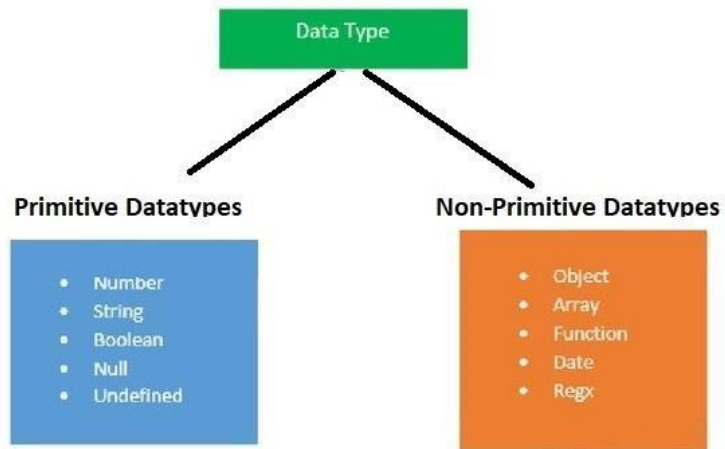
In other words, a closure gives you access to an outer function's scope from an inner function.

In JavaScript, closures are created every time when function is created.

Ex:

```
var a=10;
function f1()
{
    var b=20;
    function f2()
    {
        var c=30;
        document.writeln(a+b+c);
    }
    f2();
}
f1();
```

### 9) Explain JavaScript Data Types?



### 10) Difference between null and undefined?

In JavaScript, undefined is a type, whereas null an object.

- **undefined**

It means a variable declared, but no value has been assigned a value.

Ex:

```
<script>
    var x;
    document.writeln(x); //undefined
</script>
```

- **null**

A null in JavaScript is an assignment value.

Ex:

```
<script>
    var x=null;
    document.writeln(x); //null
</script>
```

## 11) Difference between == and ===?

- **==:**

It is used to check only equality

Ex:

```
<script>
    document.writeln ((10==10) + "<br>");//true
    document.writeln ((10=="10") + "<br>");//true
    document.writeln ((false==false) + "<br>");//true
    document.writeln ((0==false) + "<br>");//true
    document.writeln ((true==true) + "<br>");//true
    document.writeln ((1==true) + "<br>");//true
```

- **===:**

```
</script>
```

It is used to check equality and datatype.

Ex:

```
<script>
    document.writeln((10===10) + "<br>");//true
    document.writeln((10=== "10") + "<br>");//false
    document.writeln((false===false) + "<br>");//true
    document.writeln((0===false) + "<br>");//false
    document.writeln((true===true) + "<br>");//true
    document.writeln((1===true) + "<br>");//false
</script>
```

## 12) Difference between innerText and innerHTML?

- **innerText:**

The innerText property is used to write the simple text using JavaScript dynamically.

Ex:

```
<div id="result"></div>
<script>
    document.getElementById ('result').innerText="Simple Text";
</script>
```

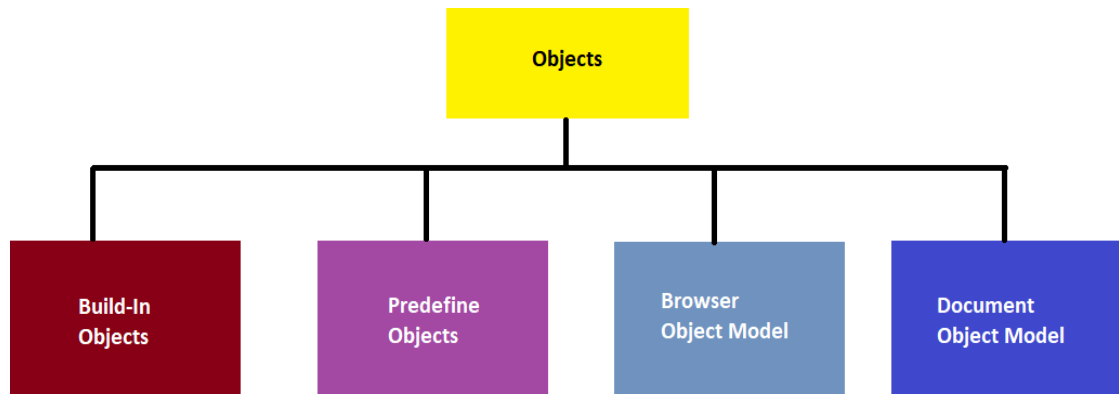
- **innerHTML:**

The innerHTML property is used to write the HTML code using JavaScript dynamically.

Ex:

```
<div id="result"></div>
<script>
    document.getElementById ('result').innerHTML="<h1>HTML Text</h1>";
</script>
```

### 13) Types of objects in JavaScript?



#### ❖ **Build-In Objects:**

Built-in objects are provided by JavaScript.

Built-in objects are not related to any Window or DOM.

These objects are used for simple data processing in the JavaScript.

Ex:

String  
Number  
Boolean  
Math  
Array

#### ❖ **Predefine Objects:**

Objects which are created by the user based on the requirement are called custom objects.

JavaScript provides built-in object called Object.

We can use “Object” to create custom objects in JavaScript.

Ex:

```
<script>
    var emp={id:101,name:"IHUB TALENT",salary:40000}
    document.writeln (emp.id+" "+emp.name+" "+emp.salary);
</script>
```

#### ❖ **Browser Object Model:**

The Browser Object Model is used to interact with browser.

The default object for a browser is window object.

We can call all the functions/methods by using window or directly.

Ex:

```
window.alert("Welcome to JavaScript");
or
alert("Welcome to JavaScript");
```

### ❖ **Document Object Model:**

When a web page is loaded, the browser creates a Document Object Model of the page.

Using document object, Our HTML document is represented in a tree node hierarchy.

A document object is a root node for HTML document tree node structure/hierarchy.

DOM always looks for three nodes.

- 1) Element node
- 2) Attribute node
- 3) Text node

Using document object we can add dynamic content to the web page.

A document object is a property of window.

It means we can call document object directory or by using window.

Ex:

```
window.document  
or  
document
```

### **14) What is JavaScript hoisting?**

Hoisting is the default behavior of JavaScript where all the variable and function declarations are moved on top.

This means that irrespective of where the variables and functions are declared, they are moved on top of the scope.

The scope can be both local and global.

Ex1:

hoistedVariable = 3;		var hoistedVariable;
console.log(hoistedVariable);	==>	console.log(hoistedVariable);
var hoistedVariable;		var hoistedVariable;

Ex2:

hoistedFunction(); //calling		function hoistedFunction()
function hoistedFunction()		{
{	==>	console.log(" Hello world! ");
console.log(" Hello world! ");		}
}		hoistedFunction(); //calling

### **15) What is window object in JavaScript?**

It is used to create a window on a browser.

A window object is created atomically by the browser.

A "window" is object of browser but not JavaScript.

A "window" object is used to write programming related to browser.

With the help of window object we can perform following activities very easily.



- ✓ It display dialog boxes and pop boxes.
- ✓ We can find width and height of a browser.
- ✓ We can move or resize the browser.
- ✓ Scroll to the browser.
- ✓ Get URL, hostname, protocol and etc. of a browser.
- ✓ We can get JavaScript history.

#### 16) List of some methods present in window object?

We have following methods in window object.

<u>Method</u>	<u>Description</u>
alert()	It will display alert box.
confirm()	It will display confirm dialog box.
prompt()	It will display prompt dialog box.
open()	Open a new window
close()	Close the current window
moveTo()	Move the current window
moveBy()	Move the current window
resizeTo()	Resize the current window

#### 17) List of some methods present in document object?

We have following methods in document object.

<u>Method</u>	<u>Description</u>
getElementById (id)	Find an element by element id
getElementsByTagName (name)	Find elements by tag name
getElementsByClassName (class)	Find elements by class name
createElement(element)	Create an HTML element
removeChild(element)	Remove an HTML element
appendChild(element)	Add an HTML element
replaceChild(new, old)	Replace an HTML element
write(text)	Write into the HTML output stream

#### 18) Explain High order functions in JavaScript?

Functions that operate on other functions, either by taking them as arguments or by returning them, are called higher-order functions.

Ex:

```
<script>
    function higherOrder (fn) {
        fn();
    }
    higherOrder(function() {document.writeln("Hello world") });
</script>
```

Ex:

```
<script>
  function higherOrder2()
  {
    return function() {
      return "Do something";
    }
  }
  var x = higherOrder2();
  x();
</script>
```

### 19) What is currying in JavaScript?

Currying simply means evaluating functions with multiple arguments and decomposing them into a sequence of functions with a single argument.

By using the currying technique, we do not change the functionality of a function, we just change the way it is invoked.

Ex:

```
<script>
  function add (a) {
    return function(b){
      return a + b;
    }
  }
  add(3)(4) ;
</script>
```

Ex:

```
<script>
  function multiply(a,b){
    return a*b;
  }

  function currying(fn){
    return function(a){
      return function(b){
        return fn(a,b);
      }
    }
  }

  multiply(4, 3); // Returns 12
  var curriedMultiply = currying(multiply);
  curriedMultiply(4)(3); // Also returns 12
</script>
```

## 20) What is the difference between `exec ()` and `test ()` methods in JavaScript?

- ✓ `test ()` and `exec ()` are RegExp expression methods used in JavaScript.
- ✓ We'll use `exec ()` to search a string for a specific pattern, and if it finds then it will return the pattern directly otherwise it will return an 'empty' result.
- ✓ We will use a `test ()` to find a string for a specific pattern. It will return the Boolean value 'true' on finding the given text otherwise, it will return 'false'.

## 21) What are the types of errors in JavaScript?

There are two types of errors in JavaScript.

- **Syntax error:**  
Syntax errors are mistakes or spelling problems in the code that cause the program to not execute at all or to stop running halfway through.
- **Logical error:**  
Reasoning mistakes occur when the syntax is proper but the logic or program is incorrect. The application executes without problems in this case. However, the output findings are inaccurate.

## 22) What is Recursion in JavaScript?

A function which calls itself for many numbers of times is called recursion.

Ex:

```
<script>
    function add(number) {
        if (number <= 0) {
            return 0;
        } else {
            return number + add(number - 1);
        }
    }
    add(3)
</script>
```

Explanation:

=> 3 + add(2) => 3 + 2 + add(1) => 3 + 2 + 1 + add(0) => 3 + 2 + 1 + 0 = 6

## 23) What is rest parameter in JavaScript?

It provides an improved way of handling the parameters of a function.

Any number of arguments will be converted into an array using the rest parameter.

Rest parameters can be used by applying three dots (...) before the parameters.

Ex:

```
<script>
    function f1(...args) { document.writeln("hello");}
    f1(10);
    f1(10,20);
    f1(10,20,30);
</script>
```

## 24) What is spread operator in JavaScript?

The spread operator is used to spreading an array.

Ex:

```
<script>
    function addFourNumbers(num1,num2,num3,num4){
        document.writeln (num1+" "+num2+" "+num3+" "+num4);
    }
    let fourNumbers = [5, 6, 7, 8];
    addFourNumbers(...fourNumbers);
</script>
```

## 25) What is Object Destructuring?

Object destructuring is a new way to extract elements from an object or an array.

Ex:

```
<script >
    const classDetails = {
        no: 101,
        name: "Alan",
        add:"hyd"
    }

    const {no:classNo, name:className,add:classAdd} = classDetails;
    document.writeln(classNo);//101
    document.writeln(className);//Alan
    document.writeln(classAdd);//hyd
```

Ex: </script>

```
<script>
    const arr = [1, 2, 3, 4];
    const [first,second,third,fourth] = arr;
    document.writeln(first); // Outputs 1
    document.writeln(second); // Outputs 2
    document.writeln(third); // Outputs 3
    document.writeln(fourth); // Outputs 4
</script>
```

## 26) Difference between localStorage and sessionStorage?

- **localStorage:**

A localStorage property allows us to save key/value pair in a browser window.

A localStorage will store the data with no expiry date. Our data will not be removed if we close the browser window or tab. It will be present in next days.

Ex:

```
localStorage.setItem("name","Alan");
var val1=localStorage.getItem("name");
```

- **sessionStorage:**

A sessionStorage property allows us to save key/value pair in a browser window. A sessionStorage store the values based on one session. Our data will be deleted once we close the browser or tab.

Ex:

```
sessionStorage.setItem("name","Alan");
var val1=sessionStorage.getItem("name");
```

## 27) Differences between var, let and const in JavaScript?

var	let	const
It is a functional scope	It is a block scope	It is a block scope
It can be declared without initialization.	It can be declared without initialization.	It can't be declared without initialization.
It can be updated.	It can be updated.	It can't be updated.
It can be re-declared.	It can't be re-declared.	It can't be re-declared.
It can be access without initialization as it default value is undefined.	It can be access without initialization as it default value is undefined.	It can't be access without initialization.

## 28) Explain debugger keyword?

The debugger keyword stops the execution of JavaScript, and calls (if available) the debugging function.

This has the same function as setting a breakpoint in the debugger.

If no debugging is available, the debugger statement has no effect.

Ex:

```
<p id="demo"></p>
<p>With the debugger turned on, the code below should stop executing
    before it executes the third line.</p>
<script>
    let x = 15 * 5;
    debugger;
    document.getElementById("demo").innerHTML = x;
</script>
```

## 29) What is form validation?

The process of checking format and pattern of form data is called form validation.

There are two ways to perform form validation.

- **Client Side Form Validation**

Form validation which is performed at client side is called client side form validation.

To perform client side form validation we need to use JavaScript.

- **Server Side Form Validation**

Form validation which is performed at server side is called server side form validation.

To perform server side form validation we will use php, .net, java, python & etc.

### **30) What is Array in JavaScript?**

A JavaScript array is an object which is collection of same or similar elements.

There are three ways to create JavaScript arrays.

- **By using Array literal**

Ex:

```
var arr=[10,20,30,40];  
var arr=["one","two","three","four"];
```

- **By creating instance of an Array**

Ex:

```
var arr=new Array();  
arr[0]=10;  
arr[1]=20;  
arr[2]=30;  
arr[3]=40;
```

- **By creating Array constructor**

Ex:

```
var arr=new Array(10,20,30,40);
```

### **31) What is String in JavaScript?**

A JavaScript String is an object which is collection of characters.

There are two ways to create JavaScript string.

- **By using String literal**

Ex:

```
var str="This is IHUB Talent";
```

- **By create instance of a String**

Ex:

```
var str=new String("IHUB Talent");
```

### **32) What is RegEx?**

In JavaScript, regular expressions are also objects.

A regular expression is a sequence of characters that forms a search pattern in text.

Usually such patterns are used by string-searching algorithms for "find" or "find and replace" operations on strings, or for input validation.

Ex:

```
[A-Za-z. ]{6,20}$
```

### 33) What is JavaScript Event?

- ✓ The change in the state of an object is known as an Event.
- ✓ In html, there are various events which represents that some activity is performed by the user or by the browser.
- ✓ When JavaScript code is included in HTML, JavaScript react over these events and allow the execution. This process of reacting over the events is called Event Handling.
- ✓ JavaScript handles the HTML events via Event Handlers.
- ✓ We have different types of events.

#### Mouse Events:

Event Performed	Event Handler	Description
Click	OnClick	When mouse click on an element
mouseover	onmouseover	When the cursor of the mouse comes over the element
mouseout	onmouseout	When the cursor of the mouse leaves an element
mousedown	onmousedown	When the mouse button is pressed over the element
mouseup	onmouseup	When the mouse button is released over the element
mousemove	onmousemove	When the mouse movement takes place.

#### Form Events:

Event Performed	Event Handler	Description
Focus	Onfocus	When the user focuses on an element
submit	Onsubmit	When the user submits the form
Blur	Onblur	When the focus is away from a form element

## Keyboard Events:

Event Performed	Event Handler	Description
Keydown & Keyup	onkeydown & onkeyup	When the user press and then release the key

## Document/Window Event:

Event Performed	Event Handler	Description
Load	Onload	When the browser finishes the loading of the page
Unload	Onunload	When the visitor leaves the current webpage, the browser unloads it
Resize	Onresize	When the visitor resizes the window of the browser

### 34) What is Synchronous JavaScript?

As the name suggests synchronous means to be in a sequence, i.e. every statement of the code gets executed one by one. So, basically a statement has to wait for the earlier statement to get executed.

Ex:

```
<script>
    document.write("Hi"); // First
    document.write("<br>");
    document.write("Mayukh") ;// Second
    document.write("<br>");
    document.write("How are you"); // Third
</script>
```

### 35) What is Asynchronous JavaScript?

Asynchronous code allows the program to be executed immediately where the synchronous code will block further execution of the remaining code until it finishes the current one. This may not look like a big problem but when you see it in a bigger picture you realize that it may lead to delaying the User Interface.

Ex:

```
<script>
    document.write("Hi");
    document.write("<br>");
```



```

        setTimeout(() => {
            document.write("Let us see what happens");
        }, 2000);
        document.write("<br>");
        document.write("End");
        document.write("<br>");
    </script>

```

### 36) Explain JavaScript promises?

Promises are used to handle asynchronous operations in JavaScript.

They can handle multiple asynchronous operations easily and provide better error handling than callbacks and events.

A Promise has four states:

**fulfilled:** Action related to the promise succeeded

**rejected:** Action related to the promise failed

**pending:** Promise is still pending i.e. not fulfilled or rejected yet

**settled:** Promise has fulfilled or rejected

Ex:

```

<script>
    var promise = new Promise(function(resolve, reject) {
        const x = "ihubtalent";
        const y = "ihubtalent1";
        if(x === y) {
            resolve();
        } else {
            reject();
        }
    });
    promise.
        then(function () {
            console.log('Success, You are a GEEK');
        }).
        catch(function () {
            console.log('Some error has occurred');
        });
</script>

```

### 37) What is JavaScript Math object?

The JavaScript Math object allows you to perform mathematical tasks on numbers.

Ex:

Math.round(x)	Returns x rounded to its nearest integer
Math.ceil(x)	Returns x rounded up to its nearest integer
Math.floor(x)	Returns x rounded down to its nearest integer
Math.trunc(x)	Returns the integer part of x (new in ES6)

### 38) What is class in JavaScript?

A JavaScript class is not an object.

It is a template for JavaScript objects.

Use the class keyword to create a class.

A class keyword is used to declare a class with any particular name.

According to JavaScript naming conventions, the name of the class always starts with an uppercase letter.

Ex:

```
<script>
    class Example
    {
        -
        -//code to be declare
        -
    }
</script>
```

### 39) What is Constructor in JavaScript?

A JavaScript constructor method is a special type of method which is used to initialize and create an object.

It is called when memory is allocated for an object.

The constructor keyword is used to declare a constructor method.

The class can contain one constructor method only.

JavaScript allows us to use parent class constructor through super keyword.

Ex:

```
class Example
{
    constructor()
    {
        -
        -// code to be declare
        -
    }
}
```

### 40) What is object in JavaScript?

A JavaScript object is an entity having state and behavior (properties and method).

Syntax:

```
var objectname =new Object();
```

### 41) What is Encapsulation in JavaScript?

The process of wrapping property and function within a single unit is known as encapsulation.

To achieve an encapsulation in JavaScript we need to do following things.

- ✓ Use var keyword to make data members private.
- ✓ Use setter methods to set the data and getter methods to get that data.

Ex:

```
class person{
    constructor(name,id){
        this.name = name;
        this.id = id;
    }
    add_Address(add){
        this.add = add;
    }
    getDetails(){
        console.log(`Name is ${this.name},Address is: ${this.add}`);
    }
}

let person1 = new person('Mukul',21);
person1.add_Address('Delhi');
person1.getDetails();
```

#### 42) What is Inheritance in JavaScript?

The JavaScript inheritance is a mechanism that allows us to create new classes on the basis of already existing classes.

It provides flexibility to the child class to reuse the methods and variables of a parent class.

The JavaScript extends keyword is used to create a child class on the basis of a parent class.

Ex:

```
<script>
    class Moment extends Date {
        constructor() {
            super();
        }
    }
    var m=new Moment();
    document.writeln("Current date:")
    document.writeln(m.getDate()+"-"+(m.getMonth()+1)+"-"+m.getFullYear());
</script>
```

#### 43) What is Abstract in JavaScript?

Hiding internal implementation and highlighting the set of services is called Abstraction. The best example of Abstraction is GUI(Graphical User Interface) ATM machine where bank people will hide internal implementation and highlights the set of services like banking, withdrawal, mini statement, balance enquiry and etc.

#### 44) What is polymorphism in JavaScript?

The process of encapsulating data members and its behaviors in a single entity is called polymorphism.

Ex:

```
class A
{
    display()
    {
        document.writeln("A is invoked<br>");
    }
}
class B extends A
{
    display()
    {
        document.writeln("B is invoked");
    }
}
A a=new A();
a.display(); // A is
invokedB b=new B();
b.display(); // B is invoked
```

#### 45) What is addEventListener in JavaScript?

The addEventListener() method attaches an event handler to the specified element.

We can add many event handlers to one element.

We can easily remove an event listener by using the removeEventListener() method.Ex:

```
p.addEventListener("click", function(){ alert("Hello World!"); });
```

## Interview Preparation

<<<<<<<<<>>>>>>>>>>>>

- 1)What is ReactJS?
- 2)Advantages of ReactJS?
- 3)What is JSX?
- 4)What is React Fragment?
- 5)What is React Component?
- 6)Function component vs Class component?
- 7)Write a program on function component and class component?
- 8)Real DOM vs Virtual DOM
- 9) props vs state
- 10) var vs let vs const
- 11) Phases of react component?
- 12) Life cycle method in react?
- 13) What is Hook?
- 14) What is react router?
- 15) What is SPA?
- 16) What is React event handling?
- 17) What is React redux?
- 18) Render vs Re-Render?
- 19) What is Reconciliation?
- 20) What is map() function?
- 21) What is filter() function?
- 22) What is reducer() function?
- 23) Controlled component vs Uncontrolled component (forms)

## Questions and Answers

=====

## 1) What is React/ReactJS?

It is a declarative, efficient and flexible javascript frontend library responsible for building frontend applications or User interfaces(UI).

It is an open source, component based frontend library responsible only for view layer of the application.

It was developed by Jordan Walke who was a software engineer at Facebook.

## 2) Advantages of React/ReactJS ?

- 1) It is easy to learn and easy to use.
- 2) It supports one way data binding.
- 3) It supports Virtual DOM.

- 4) It supported by all major browsers.
- 5) It creates reusable components.
- 6) Good Documentation and Community support.

### **3)What is JSX ?**

JSX stands for JavaScript XML.

JSX allows us to write HTML in React.

JSX tags having a tagname, attributes and children.

JSX is necessity to write React applications.

JSX makes your react code simpler and elegant.

JSX ultimately transpiles to pure JavaScript which is understood by the browsers.

### **4)What is React Fragment ?**

Fragment is used to group of list of childrens without adding extra nodes of the DOM.

In general, We can return only one element at a time but we can't return more then one element directly.

To return more then one element we need to use React Fragment.

ex:

```
<>
  -
  -
</>
```

### **5) What is React Components?**

Components are Building blocks of any React app.

Component allows us to split UI into independent reusable pieces.

ex:

navbar, header, footer , body and etc.

Components are like Javascript functions.They accept arbitrary inputs called "props" and return React Element describing what should appears on the screen.

A Component name always starts with capital letter.

we can create react component in two ways.

- 1)Function Component /functional component
- 2)Class Component

## 6)Difference between function component vs class component?

### function component

It is also known as stateless component.

In a function component we will use return keyword.

It supports hooks.

Constructor is not used.

### class component

It is a statefull component.

In a class component we will use render() method.

It does not support hooks.

Constructor is used.

## 7)Write a program on function component and class component?

function component

```
-----  
function App()  
{  
    return(  
        <h1>Function component</h1>  
    )  
}  
export default App;
```

class component

```
-----  
import {Component} from 'react';  
export default class App extends Component  
{  
    render()  
    {  
        return(  
            <h1>Class Component</h1>  
        )  
    }  
}
```

## 8)Difference between real dom vs virtual dom ?

### Real dom

It updates slow.

Can directly updates HTML.

### virtual dom

It updates faster.

Can't directly updates HTML.

Creates a new dom if element updates.  
DOM manipulation is very expensive.  
Too much of memory wastage.

Update the jsx if element updates.  
DOM manipulation is very easy.  
No memory wastage.

## 9)What is event handling in react?

Action to which a javascript can respond is called event.

ex:

clicking on button  
hovering of an element  
and etc.

Handling events on react Elements are same like handling events on DOM elements.

ex:

### Javascript

```
<button onclick="f1()">clickMe</button>
```

### React

```
<button onClick={handleClick}>clickMe</button> --> Function component
```

```
<button onClick={this.handleClick}>clickMe</button> --> Class component
```

## 10)Difference between props and state ?

### **props**

Props are read-only.  
Props are immutable.  
Props allow us to pass data from one component to other components as an argument.

### **state**

States are updatable.  
State is mutable.  
State holds information about the components.

Props can be accessed by the child component.  
State cannot be accessed by child components because it is private.

Stateless component can have Props.      Statefull components can have state.

## 11) Phases of ReactJS component ?

There are four Phases of components in ReactJS.



- 1)Mounting
- 2)Updating
- 3)Error Handling
- 4)Unmounting
- 1)Mounting

-----

Mounting is a process of creating an element and inserting it in a DOM tree.

- 2)Updating

-----

Updating is a process of changing state or props of a component and update changes to nodes already existing in the DOM.

- 3)Error Handling

-----

Error Handling used when there is a error during rendering ,in lifecycle method or in the constructor of any child component.

- 4)Unmounting

-----

Unmounting is a process of removing components from the DOM.

In general it will clear the reserved memory.

## **12)Explain life cycle methods of mounting ?**

Mounting phase contains four methods.

- 1) constructor
- 2) getDerivedStateFromProps
- 3) render()
- 4) coumponentDidMount()

## **13)Explain life cycle methods of unmounting?**

Unmounting phase contains one method.

- 1) componentWillUnmount()

## **14)Explain life cycle methods of updating?**

updating phase contains five methods.

- 1) getDerivedStateFromProps
- 2) shouldComponentUpdate()
- 3) render()
- 4) getSnapshotBeforeUpdate()
- 4) ComponentDidUpdate()

### **15)What is Hooks?**

Hooks allow us to access state , lifecycle methods and other React features in a function component..

React provides a few built-In hooks like useState,useEffect and etc.

### **16)What is React Router?**

React Router is a standard library system built on top of the React and used to create routing in the React application using React Router Package.

ReactJS Router is mainly used for developing Single Page Web Applications.

### **17)What is SPA?**

A Single Page Application (SPA) is a single web page, website, or web application that works within a web browser and loads just a single document. It does not need page reloading during its usage.

## **Programs**

### **1)Write a javascript program to display 1 to 100 prime numbers ?**

```
let count=0
let i,j
for(j=2;j<=100;j++)
{
    for( i=1;i<=j;i++)
    {
        if(j%i==0)
            count++
    }

    if(count==2)
    {
        console.log(j)
    }
    count=0
}
```

## 2) Difference between render and re-render ?

Render refers to rendering the content for the initial load.

Re-render refers to re-rendering the content upon updating of props.

## 3)What is map() method ?

The map() method is used for traversing or displaying the list of similar objects of a component.  
ex:

```
var arr = [10,20,30];
const result = arr.map((element)=>{
  return (element + 10);
});
console.log(result);
```

## 4)What is filter() method ?

The filter() method is an iterative method.

It calls a provided callbackFn function once for each element in an array, and constructs a new array of all the values for which callbackFn returns a truthy value.

ex:

```
const ages = [32, 33, 16, 40];
const result = ages.filter(checkAdult);
document.writeln(result);

function checkAdult(age) {
  return age >= 18;
}
```

## 5)What is reduce() method?

The reduce() method executes a reducer function for array element. The reduce() method returns a single value: the function's accumulated result.

ex:

```
const numbers = [175, 50, 25];
var result=numbers.reduce(myFunc);
document.writeln(result);
```

```
function myFunc(total, num) {  
    return total + num;  
}
```

## **6)What is Reconciliation?**

Reconciliation is the process by which React updates the UI to reflect changes in the component state.

The reconciliation algorithm is the set of rules that React uses to determine how to update the UI in the most efficient way possible. React uses a virtual DOM (Document Object Model) to update the UI.

## **7)What is Redux?**

Redux is a pattern and library for managing and updating application state using events called actions.

React uses Redux for building the user interface.

It serves as a centralized store for the state that needs to be used across your entire application, with the rules ensuring that the state can only be updated in a predictable fashion.

We have three components in react redux.

- 1)STORE
- 2)ACTION
- 3)REDUCER

## **8)What is react form?**

React offers a stateful, reactive approach to build a form.

In React, form data is usually handled by the components.

There are mainly two types of form input in React.

- 1)controlled component
- 2)UnControlled component

In React, the form is usually implemented by using controlled components.

## **9)What is DOM manipulation?**

It is the process of interacting with the DOM API to change or modify an HTML document that will be displayed in a web browser.

DOM manipulation is when we use JavaScript to add, remove, and modify elements of a website.

### 10) How to combine two arrays using spread operator?

```
const arr1=[1,2,3,4,5];
const arr2=[6,7,8,9,10];
const result=[...arr1,...arr2];
document.writeln(result);
```

### 11) Write a javascript program to find out fibonacci series of a given number?

```
var n=5;

var a=0,b=1,c;

document.writeln(a+" "+b+" ");

for(var i=2;i<=n;i++)
{
    c=a+b;
    document.writeln(c+" ");
    a=b;
    b=c;
}
```

### 12)Write a javascript program to find out reverse of a given string?

```
var str="hello";
var carr=str.split('').reverse().join('');
document.writeln(carr);
```

### 13) How to declare a state in react?

There are two ways are there two declare the state.

1)Inside the class

2)Inside the constructor

#### 1)Inside the class

```
import {Component} from "react";
```

```
export default class App extends Component
```

```
{
    state={
```

```

        name: "Alan"
    }
    render()
    {
        return(
            <h1>My Name : {name}</h1>
        )
    }
}

```

## **2)Inside the constructor**

```
import {Component} from "react";
```

```
export default class App extends Component
```

```

{
    constructor(props)
    {
        super(props);

        this.state={
            name: "Alan"
        }
    }
    render()
    {
        return(
            <h1>My Name : {name}</h1>
        )
    }
}

```

## **12)How to declare a props in react?**

### **index.js**

```
<App name="Alan"/>
```

### **App.js**

```
function App(props)
```

```

{
    return(

```

```
        <h1>My Name : {props.name}</h1>
      )
}
export default App;
```

# Spring Framework

---

Framework :

Application FrameworkVersion : 5.3.9 (5.X)

Vendor : Interface2

Creator : Rod Johnson Containers : BeanFactory container

ApplicationContext containerwebsite : [www.tutorialspoint.com](http://www.tutorialspoint.com)

[www.roseindia.net](http://www.roseindia.net) [www.javatpoint.com](http://www.javatpoint.com) [www.Dzone.com](http://www.Dzone.com)

Books : spring in ActionDownload link :

<https://repo.spring.io/ui/native/release/org/springframework/spring/>

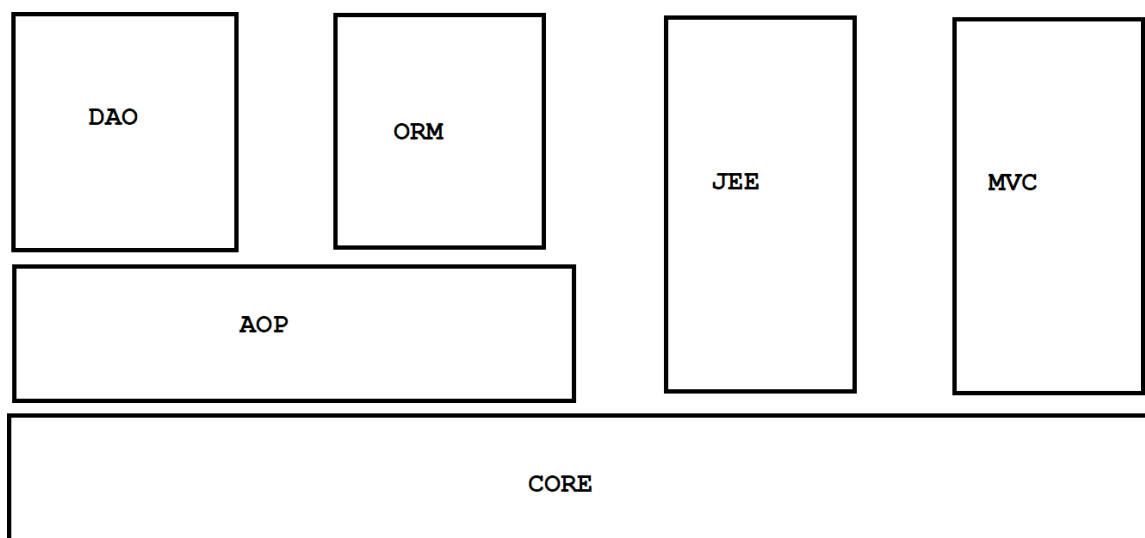
## Q) What is spring Framework?

A spring is a non-invasive, light weight, loosely coupled, dependency injection,

Aspect oriented , open source java based application framework which is used to develop all types of applications.

## How many modules are there in spring framework?

There are six modules present in spring framework.





## **CORE**

It is a base module for entire spring framework.

It is used to perform following activities.ex: 1) It obtain containers  
2) It performs dependency injections  
3) It creates spring beans

## **AOP**

AOP stands for Aspect Oriented Programming.

It is used to inject and eliminate middleware services to/from the application.

## **DAO**

**DAO stands for Data Access Object.**

It is a abstract layer on JDBC.

While working JDBC we need to handle the exceptions. That problem is eliminated in DAO module. Here checked exceptions converts to unchecked exceptions. So we don't need to handle any exceptions.

## **ORM**

ORM stands Object Relational Mapping.

ORM is a abstract layer on ORM tools like hibernate, ibatis, jpa and etc.

While transfer the data in the form objects there are some drawbacks, those problems solved by ORM module.

## **JEE**

This module is used to develop middleware services.

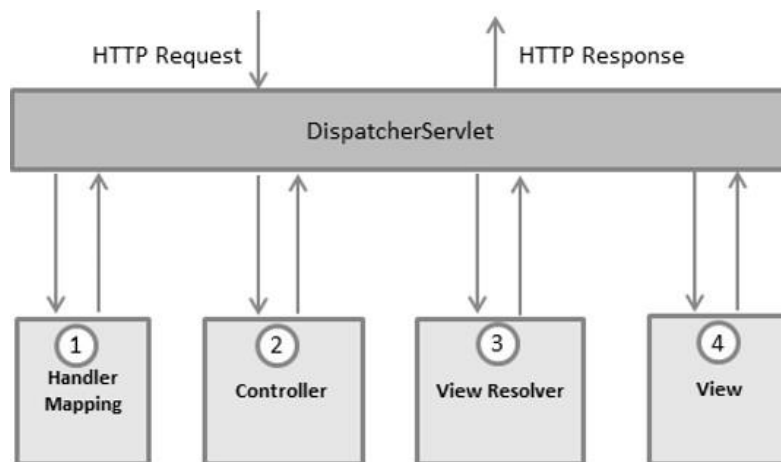
## **MVC**

MVC stands for Model View Controller.

It is used to create completely MVC based web application.

## Q) Explain spring MVC architecture?

Diagram: sb1.2



After receiving an HTTP request, DispatcherServlet consults the HandlerMapping to call the appropriate Controller.

The Controller takes the request and calls the appropriate service methods based on used GET or POST method. The service method will set model data based on defined business logic and returns view name to the DispatcherServlet.

The DispatcherServlet will take help from ViewResolver to pickup the defined view for the request.

Once view is finalized, The DispatcherServlet passes the model data to the view which is finally rendered on the browser.

## Limitations with Spring Framework

In spring framework, A developer is responsible to 1) Add dependencies/Jar files

Perform configurations.

Arrange a virtual server (Tomcat, GlassFish and etc) to deploy web applications.

Arrange a Physical database (Oracle, MySQL, mongodb and etc) to perform database operations. We can overcome above limitations by using spring Boot.

Diagram:

- Developer
- | Spring Boot
- | Spring

## What is Spring Boot?

Spring Boot is an open source Java-based framework developed by Pivotal Team.

Spring Boot provides the RAD (Rapid Application Development) feature to the Spring framework.

Spring Boot is used to create stand-alone, production-grade Spring based Applications with minimum configuration.

In short, Spring Boot is the combination of:

Spring Framework + Embedded Servers + Embedded Database.(Tomcat/Jetty/Undertow)  
(H2/HSQ/ Derby)

In Spring Boot, there is no requirement for XML configuration. XML configurations are replaced by Annotations.

## Advantages of Spring Boot (stater)

It creates stand-alone Spring applications that can be started using Java -jar.

It provides production-ready features such as metrics, health checks, and externalized configuration. It increases productivity and reduces development time.

It offers the number of plug-ins.

There is no requirement for XML configuration.

It provides opinionated 'starter' POMs to simplify our Maven configuration.

It tests web applications easily with the help of different Embedded HTTP servers such as Tomcat, Jetty, Undertow etc. We don't need to deploy WAR files.

It offers a CLI tool for developing and testing the Spring Boot application.

It also minimizes writing multiple boilerplate codes (the code that has to be included in many places with little or no alteration), XML configuration, and annotations.

## Q) Components of spring boot?

We have four components in spring boot.

Autoconfiguration

starter

CLI

Actuators

## Q) In how many ways we can create a spring boot project?

There are two ways to create spring boot project.

1)Using IDE's (STS , IntelliJ)

2)Using spring Initializr

### STS IDE

Spring Tool Suite is an IDE to develop Spring applications. It is an Eclipse-based development environment.

It provides a ready-to-use environment to implement, run, deploy, and debug the application.

Step 1: Download Spring Tool Suite

ex: <https://spring.io/tools#suite-thre>

Step 2: Extract the zip file and install the STS.

Ex: sts-bundle -> sts-3.9.9.RELEASE -> Double-click on the STS.exe.

### Spring Boot Starters

Spring Boot provides a number of starters that allow us to add jars in the classpath. Spring Boot built-in starters make development easier and rapid.

Spring Boot Starters are the dependency descriptors.

In the Spring Boot Framework, all the starters follow a similar naming pattern: spring-boot-starter-\*, where \* denotes a particular type of application.

ex: spring-boot-starter-test spring-boot-starter-web

spring-boot-starter-validation (bean validation) spring-boot-starter-security

spring-boot-starter-data-jpa spring-boot-starter-data-mongodb spring-boot-starter-mail

### Third-Party Starters

We can also include third party starters in our project.

The third-party starter starts with the name of the project.

ex: abc-spring-boot-starter.

### Spring Boot Starter Web

There are two important features of spring-boot-starter-web.

>It is compatible for web development

>AutoConfiguration

If we want to develop a web application, we need to add the following dependency in pom.xml file.ex:

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-web</artifactId>

<version>2.2.2.RELEASE</version>

</dependency>
```

Spring web starter uses Spring MVC, REST and Tomcat as a default embedded server.

The single spring-boot-starter-web dependency transitively pulls in all dependencies related to web development.

By default, the spring-boot-starter-web contains the following tomcat server dependency:ex:

```
<dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-tomcat</artifactId>

<version>2.0.0.RELEASE</version>

<scope>compile</scope>

</dependency>
```

The spring-boot-starter-web ,auto-configures the following things that are required for the web development:

Dispatcher Servlet

Error Page

Web JARs for managing the static dependencies

Embedded servlet container

Interview question

### **Q)Explain @SpringBootApplication annotation?**

This annotation is used to enable the following annotations. @EnableAutoConfiguration: It enables the Spring Boot auto-configuration mechanism.@ComponentScan: It scans the package where the application is located.

@Configuration: It allows us to register extra beans in the context or import additional configuration classes.

## **Where to do spring boot application configurations?**

ex: We can configure application configuration in two files.

application.properties file 2) application.yml file

## **Q)List of stereotype Annotation?**

@Component @Configuration @Service @Repository @Controller and etc.

## **Q)In spring boot mvc based web application who will pass HTTP request to controller?**

ans) RequestDispatcher.

## **Q)List of embedded servers present in spring boot?**

ans) Tomcat, Jetty and undertow

## **Q)Tomcat embedded server by default runs under which port no?**

8080.

## **Q)To create a spring mvc based web application we need to add which starter?**

spring-boot-starter-web

## **Q)How to change tomcat embedded server port no?**

ans) *application.properties*

server.port = 9090

Spring Data JPA

Spring Data JPA handles most of the complexity of JDBC-based database access and ORM (Object Relational Mapping).

It reduces the boilerplate code required by JPA (Java Persistence API).

It makes the implementation of your persistence layer easier and faster.

Spring Data JPA aims to improve the implementation of data access layers by reducing the effort to the amount that is needed.

Spring Boot provides spring-boot-starter-data-jpa dependency to connect Spring application with relational database efficiently.

```
ex:    <dependency>

<groupId>org.springframework.boot</groupId>

<artifactId>spring-boot-starter-data-jpa</artifactId>

<version>2.2.2.RELEASE</version>

</dependency>
```

The spring-boot-starter-data-jpa internally uses the spring-boot-jpa dependency.

### **Spring Data JPA provides three repositories are as follows:**

**CrudRepository:**

It offers standard create, read, update, and delete. It contains methods like findOne(), findAll(), save(), delete(), etc.

**PagingAndSortingRepository:**

It extends the CrudRepository and adds the findAll methods. It allows us to sort and retrieve the data in a paginated way.

**JpaRepository:**

It is a JPA specific repository. It is defined in Spring Data Jpa. It extends the both repository CrudRepository and PagingAndSortingRepository. It adds the JPA-specific methods, like flush() to trigger a flush on the persistence context.

### **Exception Handling in Spring Boot**

If we give/pass wrong request to our application then we will get Exception.ex:

```
http://localhost:9090/fetch/102
```

Here '102' record is not available so immediately our controller will throw below exception.ex:

```
{

"timestamp": "2021-02-14T06:24:01.205+00:00",
```

```
"status": 500,  
  
"error": "Internal Server Error", "path": "/fetch/102"  
  
}
```

Handling exceptions and errors in APIs and sending the proper response to the client is good for enterprise applications.

In Spring Boot Exception handling can be performed by using Controller Advice.

@ControllerAdvice

The @ControllerAdvice is an annotation is used to to handle the exceptions globally.

@ExceptionHandler

The @ExceptionHandler is an annotation used to handle the specific exceptions and sending the custom responses to the client.

Monolithic Architecture

Monolith means composed all in one piece.

The Monolithic application describes a single-tiered software application in which different components combined into a single program from a single platform.

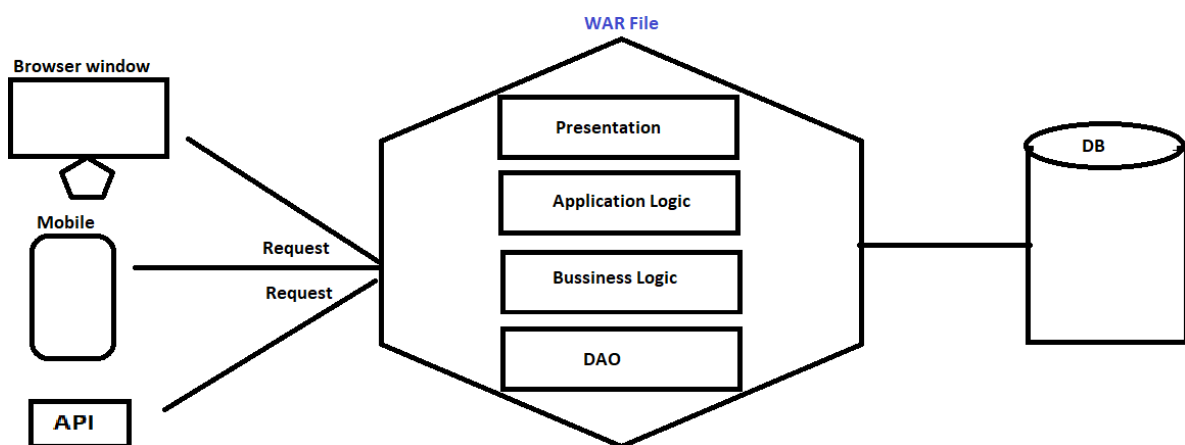


Diagram: sb5.1



In Monolithic Architecture we are developing every service individually and at end of the development we are packaging all services as single war file and deploying in a server.

### **Advantages**

- 1) Simple to develop
- 2) Simple to test
- 3) Simple to deploy
- 4) Simple to scale

### **Drawbacks of Monolithic Architecture**

- 1) Large and Complex Application
- 2) Slow Development
- 3) Blocks Continuous development
- 4) Unscalable
- 5) Unreliable
- 6) Inflexible

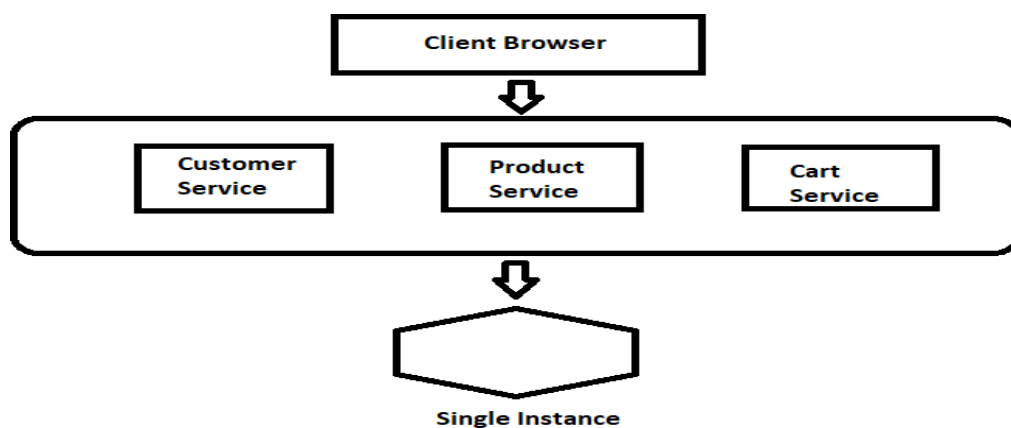
### **MicroService Architecture**

Microservices are the small services that work together

The microservice defines an approach to the architecture that divides an application into a pool of loosely coupled services that implements business requirements.

In Microservice architecture, Each service is self contained and implements a single business capability.

Diagram: sb5.2



The microservice architectural style is an approach to develop a single application as a suite of small services. It is next to Service-Oriented Architecture (SOA).

Each microservice runs its process and communicates with lightweight mechanisms. These services are built around business capabilities and independently developed by fully automated deployment machinery.

Advantages of Microservice Architecture

### **Independent Development**

Each microservice can be developed independently.

A single development team can build, test, and deploy the service.

### ***Independent Deployment***

We can update the service without redeploying the entire application. Bug release is more manageable and less risky.

### ***Mixed Technology Stack***

It is used to pick the best technology which is best suitable for our application.

### ***Granular Scaling***

In granular scaling, services can be scaled independently instead of the entire application.

## **List of companies which are working with Microservices**

Amazon Netflix SoundCloud Twitter Uber PayPal

eBay GILT

The Guardian NORDSTROM

and etc.

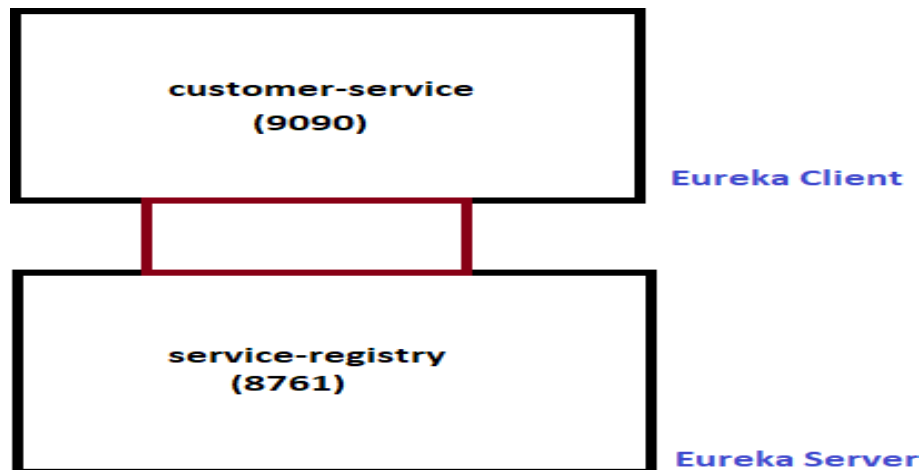
customer-microservice

### **Eureka Server**

Eureka server is also known as service registry. Eureka server runs on 8761 port number.

It will register the micro-services in our application.

Diagram: sb6.1



## **Spring Cloud API Gateway**

Spring Cloud Gateway aims to provide a simple, effective way to route to API's and provide cross cutting concerns to them such as

security, monitoring/metrics , authentication, authorization , adaptor and etc.

### **Features**

Built on spring Framework 5, project reactor and spring boot 2.0.

Able to match routes on any request attributes. Predicates and filters are specific to routes.

Circuit breaker integration.

Spring cloud Discovery Client integration. Easy to write predicates and filters.

Request Rate limiting.

## **Spring Cloud Hystrix**

Hystrix is a fault tolerance library provided by Netflix.

Using Hystrix we can prevent Deligation of failure from one service to another service. Hystrix internally follows Circuit Breaker Design pattern.

In short circuit breaker is used to check availability of external services like web service call, database connection and etc.

## **Spring Security**

Spring Security is a framework which provides various security features like authentication, authorization to create secure Java Enterprise Applications.

It is a sub-project of Spring framework which was started in 2003 by Ben Alex.

Later on, in 2004, It was released under the Apache License as Spring Security 2.0.0.

This framework targets two major areas of application i.e authentication and authorization.

### **Authentication**

It is a process of knowing and identifying the user that wants to access.

### **Authorization**

It is a process to allow authority to perform actions in the application.

We can apply authorization to authorize web request, methods and access to individual domain. Spring Security framework supports wide range of authentication models.

These models either provided by third parties or framework itself.

