

Bootstrap and jQuery

**15-07-2022

The Responsive web pages are made with the help of bootstrap.

sm-small devices

md-medium size

lg-large size

xl-extra large devices

Container-fluid class-Less space from body

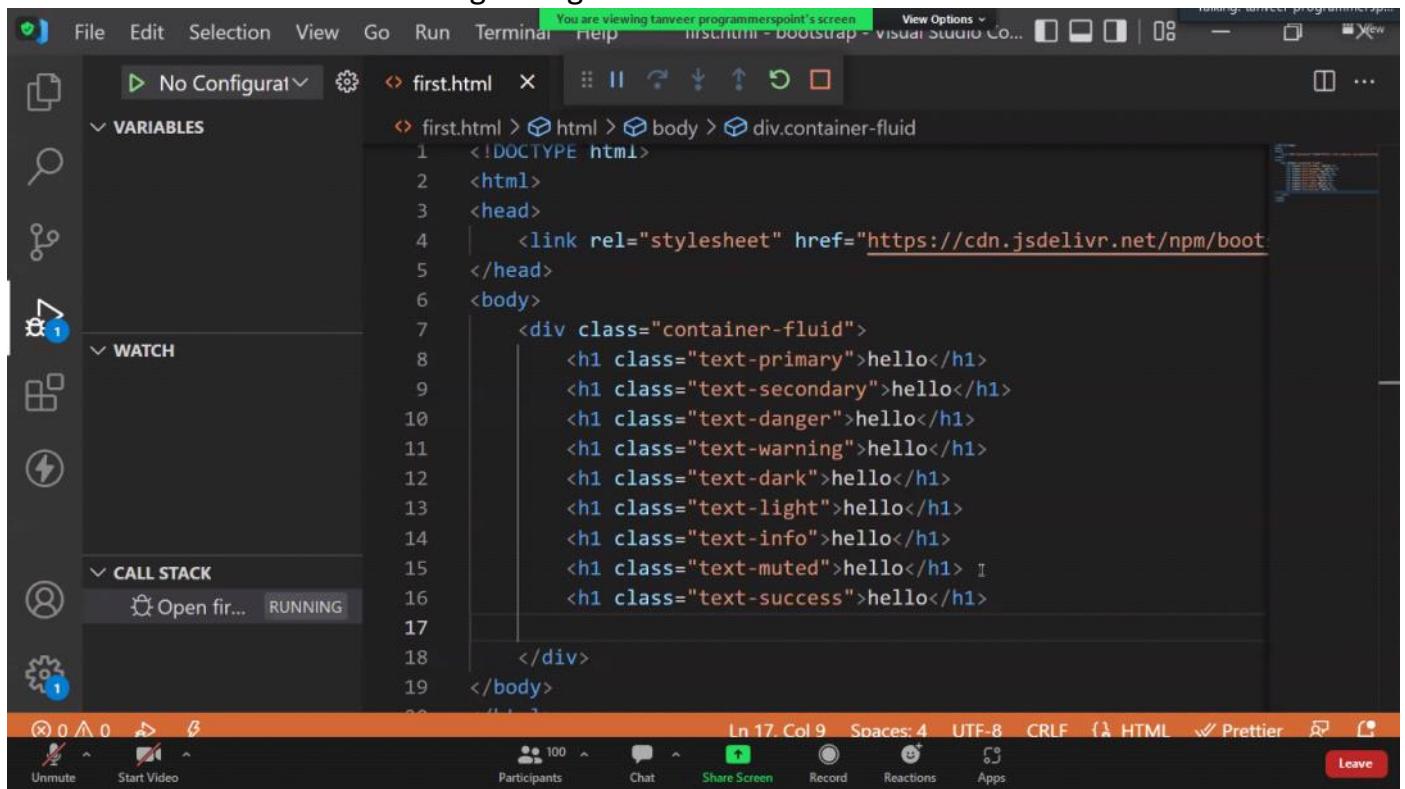
Container class-More space from body

Ctr+space(Shortcut key for suggestions)

**AOS-Animation on scroll

CDN=Content Delivery Network

Add colour to the text in html tags using text-

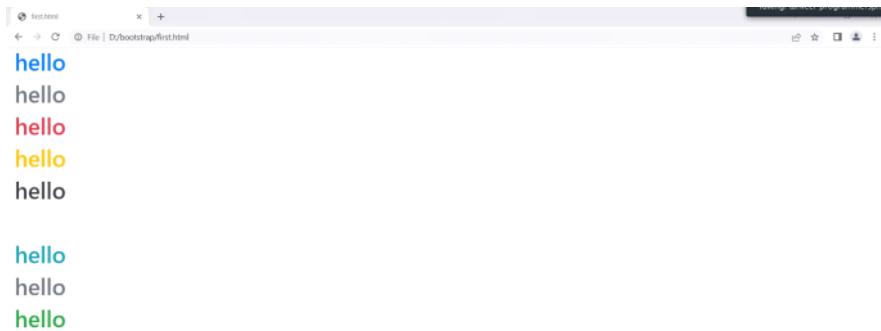


The screenshot shows a Visual Studio Code interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, Terminal, Help.
- Code Editor:** The file 'first.html' is open, showing the following code:

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"/>
</head>
<body>
    <div class="container-fluid">
        <h1 class="text-primary">hello</h1>
        <h1 class="text-secondary">hello</h1>
        <h1 class="text-danger">hello</h1>
        <h1 class="text-warning">hello</h1>
        <h1 class="text-dark">hello</h1>
        <h1 class="text-light">hello</h1>
        <h1 class="text-info">hello</h1>
        <h1 class="text-muted">hello</h1>
        <h1 class="text-success">hello</h1>
    </div>
</body>
```
- Sidebar:** Variables, Watch, Call Stack.
- Bottom Bar:** Line numbers 1-19, Col 9, Spaces: 4, UTF-8, CRLF, HTML, Prettier, Participants: 100, Chat, Share Screen, Record, Reactions, Apps, Unmute, Start Video, Leave.

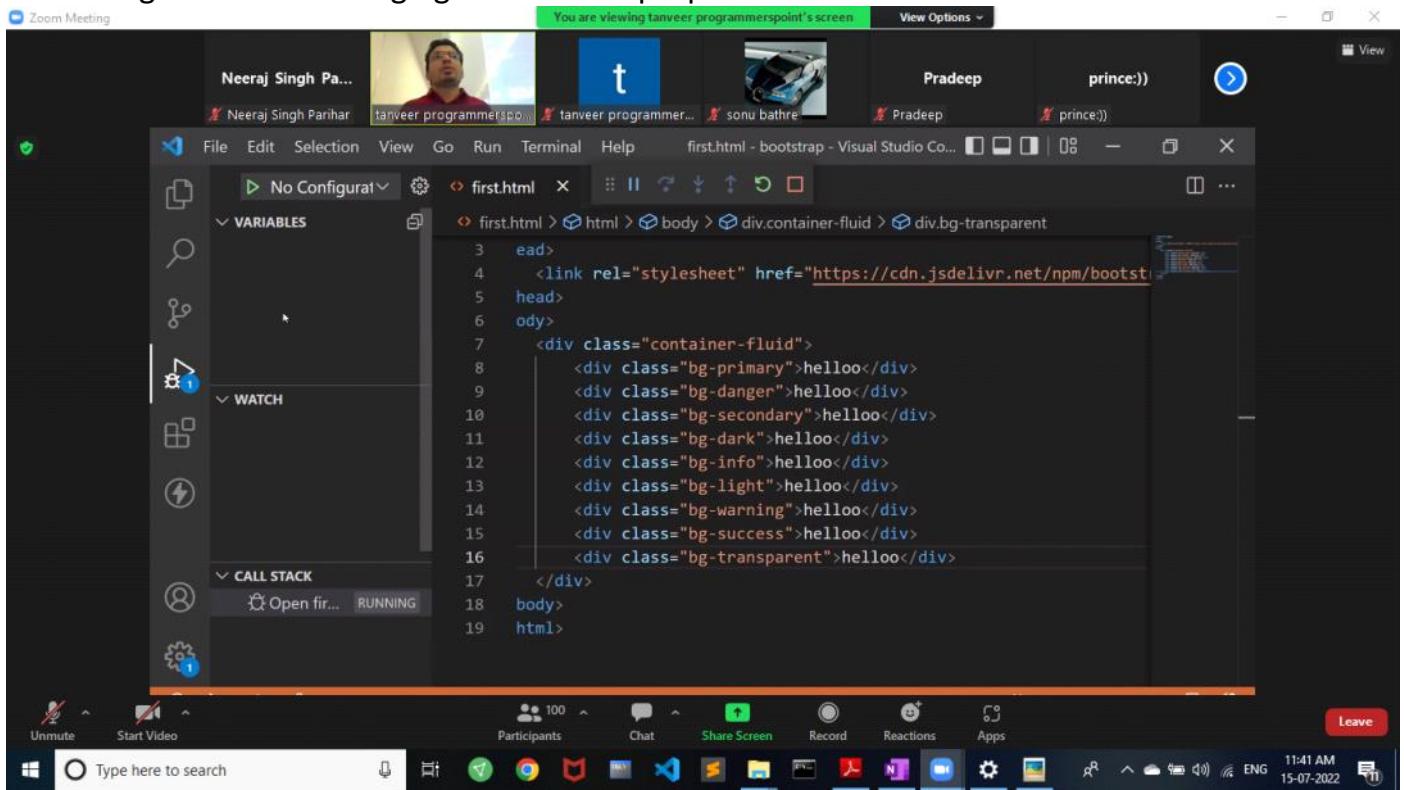
Output:-



The screenshot shows a browser window with the title 'first.html'. It displays several lines of text: 'hello' in blue, 'hello' in black, 'hello' in red, 'hello' in green, and 'hello' in black again. This demonstrates the use of Bootstrap's background color classes.



Add Background color using bg class and its properties



This screenshot shows a Zoom meeting interface. At the top, it says 'You are viewing tanveer.programmerspoint's screen'. Below the video feed, there's a toolbar with file operations like File, Edit, Selection, View, Go, Run, Terminal, Help, and a tab for 'first.html - bootstrap - Visual Studio Co...'. The main area shows a code editor with the following HTML snippet:

```
3 <head>
4   <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css">
5 <head>
6 <body>
7   <div class="container-fluid">
8     <div class="bg-primary">hello</div>
9     <div class="bg-danger">hello</div>
10    <div class="bg-secondary">hello</div>
11    <div class="bg-dark">hello</div>
12    <div class="bg-info">hello</div>
13    <div class="bg-light">hello</div>
14    <div class="bg-warning">hello</div>
15    <div class="bg-success">hello</div>
16    <div class="bg-transparent">hello</div>
17  </div>
18 </body>
19 </html>
```

The code editor has a sidebar with 'VARIABLES' and 'WATCH' sections. At the bottom, there are controls for Unmute, Start Video, Participants, Chat, Share Screen, Record, Reactions, and Apps. The Windows taskbar at the very bottom shows various pinned icons and the date and time as '15-07-2022 11:41 AM'.

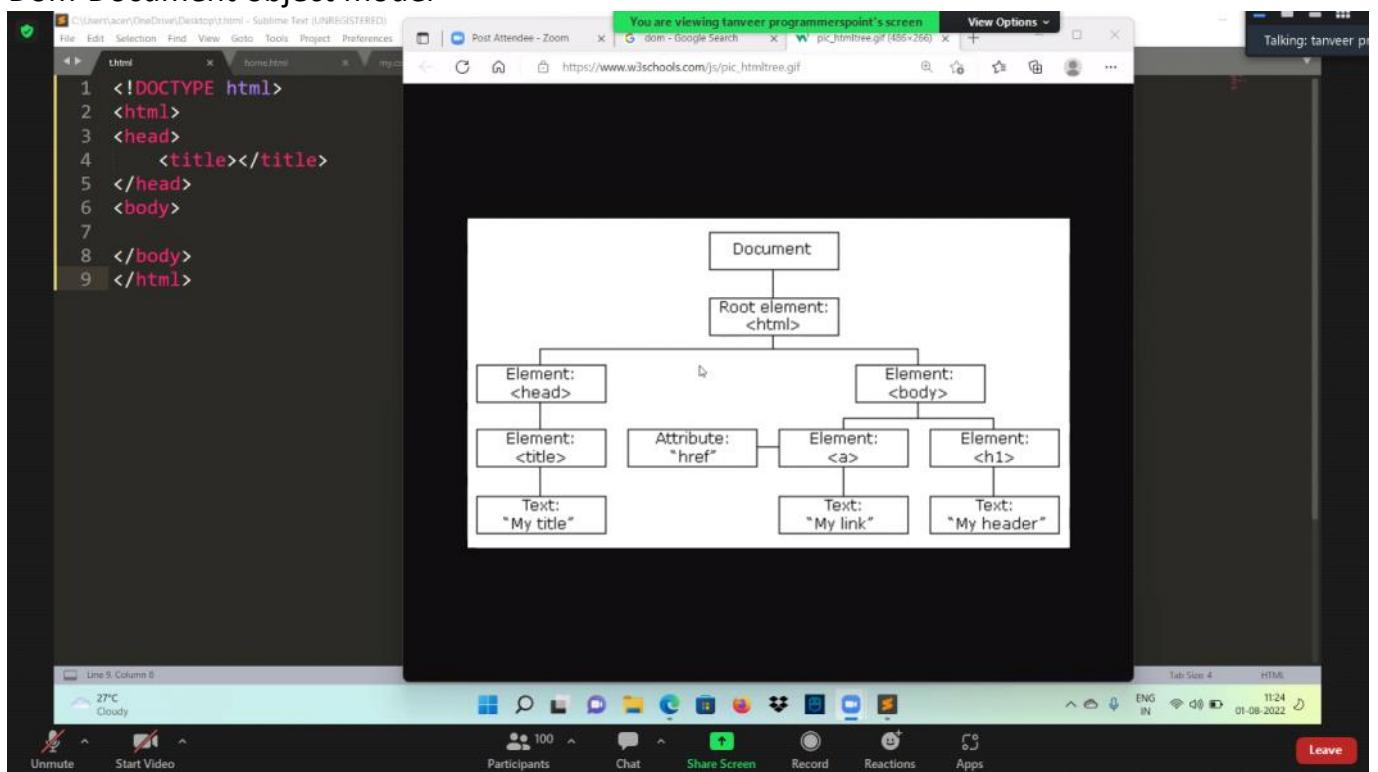
Grid:- A browser or a client document area ie screen can be divided into rows and columns. A row can contain maximum 12 column and vice versa
We will use col and row classes to create grid in bootstrap and it will be responsive

```
<head>
</head>
<body>
    <div class="container-fluid">
        <div class="row">
            <div class="col-12 bg-dark">12</div>
        </div>
        <div class="row">
            <div class="col-6 bg-primary">6</div>
            <div class="col-6 bg-danger">6</div>
        </div>
        <div class="row">
            <div class="col-4 bg-dark">4</div>
            <div class="col-4 bg-info">4</div>
            <div class="col-4 bg-warning">4</div>
        </div>
    </div>
</body>
</html>
```

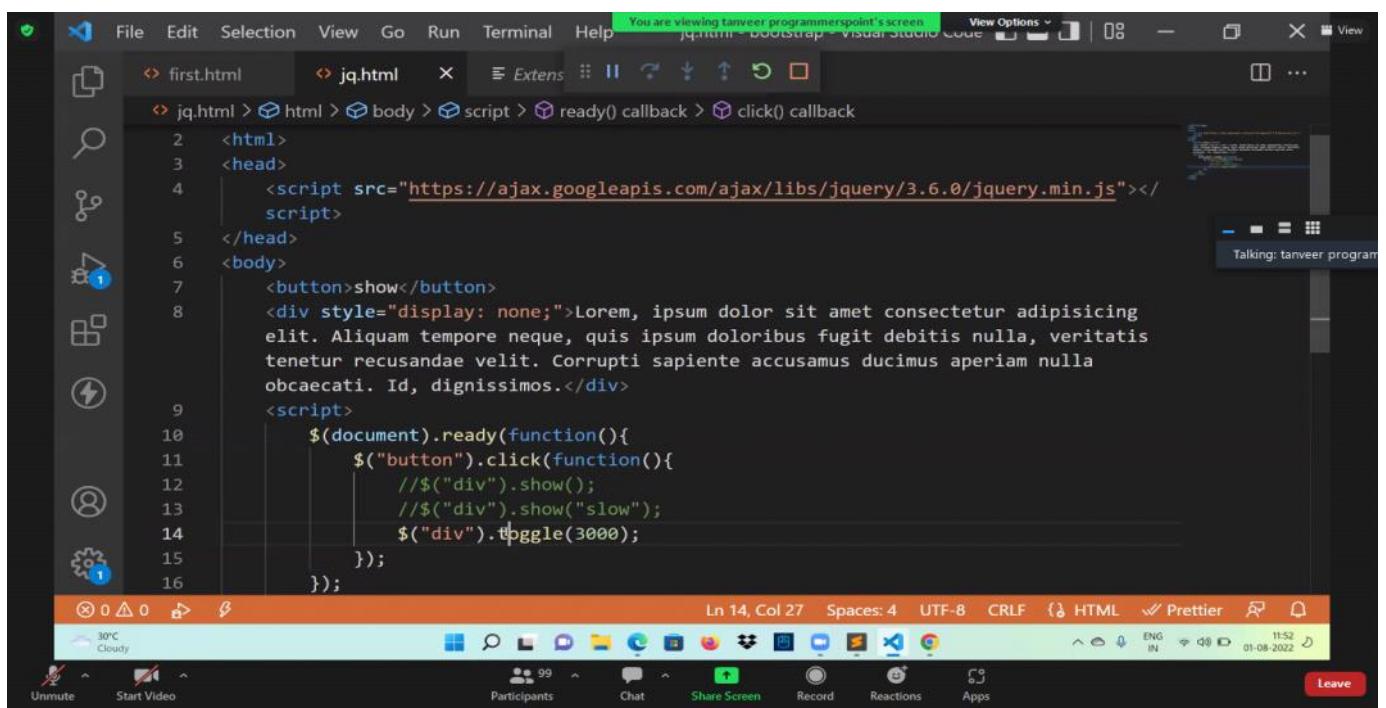
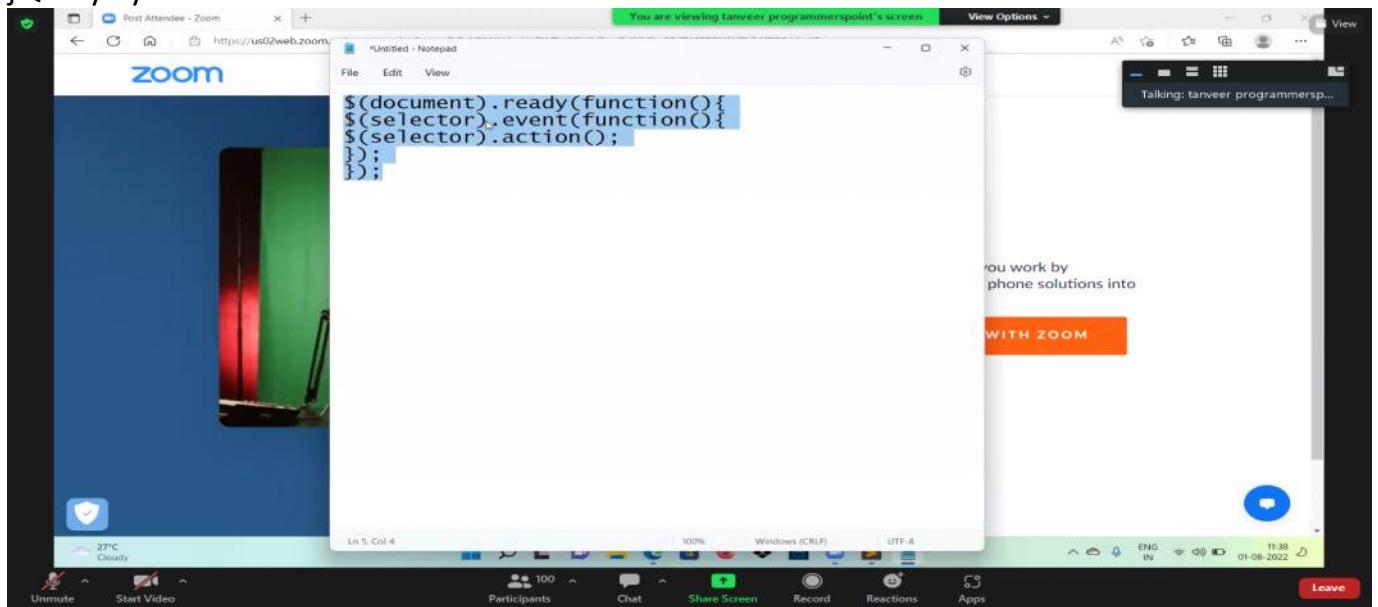
Scroll spy is mainly used to create single page application using bootstrap

JQuery

jQuery is basically JavaScript library write less do more library , which is mainly used to construct interactive front end application, in jQuery we will use \$ symbol to access jQuery elements
Dom-Document object model



jQuery Syntax



```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>
<body>
<button>show</button>
<div style="display: none;">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Aliquam tempore neque, quis ipsum doloribus fugit debitis nulla, veritatis tenetur recusandae velit. Corrupti sapiente accusamus ducimus aperiam nulla obcaecati. Id, dignissimos.</div>
<script>
$(document).ready(function(){
    $("button").click(function(){
        //$("#div").show();
        //$("#div").show("slow");
        $("div").toggle(3000);
    });
})</script>
```

Participants: 99 Chat Share Screen Record Reactions Apps

Unmute Start Video Talking: tanveer programmerspoint 11:55 01-08-2022 Leave

```
<button id="b1">show</button>
<button id="b2">hide</button>
<button class="b3">show/hide</button>
<div style="display: none;">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Aliquam tempore neque, quis ipsum doloribus fugit debitis nulla, veritatis tenetur recusandae velit. Corrupti sapiente accusamus ducimus aperiam nulla obcaecati. Id, dignissimos.</div>
<script>
$(document).ready(function(){
    $("#b1").click(function(){
        $("div").show();
    });
    $("#b2").click(function(){
        $("div").hide();
    });
    $(".b3").click(function(){
        $("div").toggle();
    });
})</script>
```

Participants: 97 Chat Share Screen Record Reactions Apps

Unmute Start Video Talking: tanveer programmerspoint... 12:02 01-08-2022 Leave

jQuery chaining

This:-

This is used to select current tag or property

Empty will delete child content of a specific element or tag

Remove will delete current content as well as child content

AJAX ajax stands for asynchronous JavaScript and xml. It is mainly used to construct a application that generate response without reload or refresh current page, it means it will take content from server without refresh or reload current html page

Load method:-load method is used to load some content like a text file or jason file from server without loading or refreshing current page

Get-to fetch data from server

jQuery \$.get:-\$.get is mainly used to execute get type request, we can fetch data from server with the help of get method

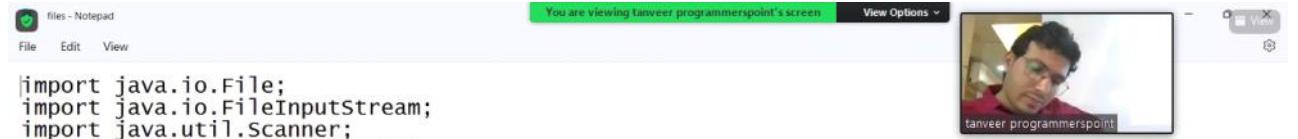
Get method contains two arguments , 1st one is url and 2nd one is call back function.

Post-it is mainly used to transfer data on server, post contains 3 argument

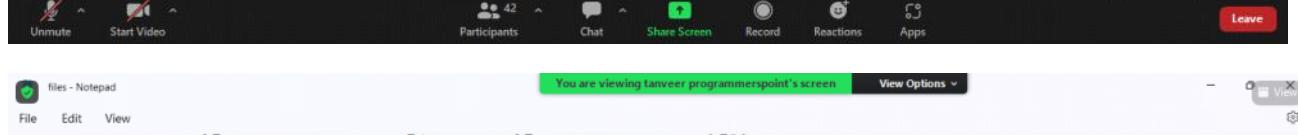
1st one url, 2nd one is javascript object in which we will pass our data and 3rd one is callback function that contains data and status in arguments

Java Core

Sequence input stream class :-it is mainly used to read data from multiple files it's constructor contain varags in arguments.



```
files - Notepad
File Edit View
You are viewing tanveer programmerspoint's screen View Options
tanveer programmerspoint
import java.io.File;
import java.io.FileInputStream;
import java.util.Scanner;
import java.io.SequenceInputStream;
class Demo
{
    public static void main(String gg[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter first file name");
        String fname1=sc.next();
        System.out.println("enter second file name");
        String fname2=sc.next();
        File f3=new File("D:\\aaa\\"+fname1);
        File f4=new File("D:\\aaa\\"+fname2);
        if(f3.exists()&&f4.exists())
        {
            try
            {
                FileInputStream f1=new FileInputStream(f3);
                FileInputStream f2=new FileInputStream(f4);
                if(f3.canRead()&&f4.canRead())
                {
                    SequenceInputStream se=new SequenceInputStream(f1,f2);
                    int i;
                    String s="";
                    while((i=se.read())!=-1)
                    {
                        s=s+(char)i;
                    }
                    System.out.println(s);
                }
                f1.close();
                f2.close();
            }
            catch(Exception e)
            {
                e.printStackTrace();
            }
        }
        else
        {
            System.out.println("file not exist");
        }
    }
}
Unmute Start Video Participants Chat Share Screen Record Reactions Apps Talking: Leave
```



```
files - Notepad
File Edit View
You are viewing tanveer programmerspoint's screen View Options
tanveer programmerspoint
Talking:
Unmute Start Video Participants Chat Share Screen Record Reactions Apps Leave
```



```
Unmute Start Video Participants Chat Share Screen Record Reactions Apps Leave
```

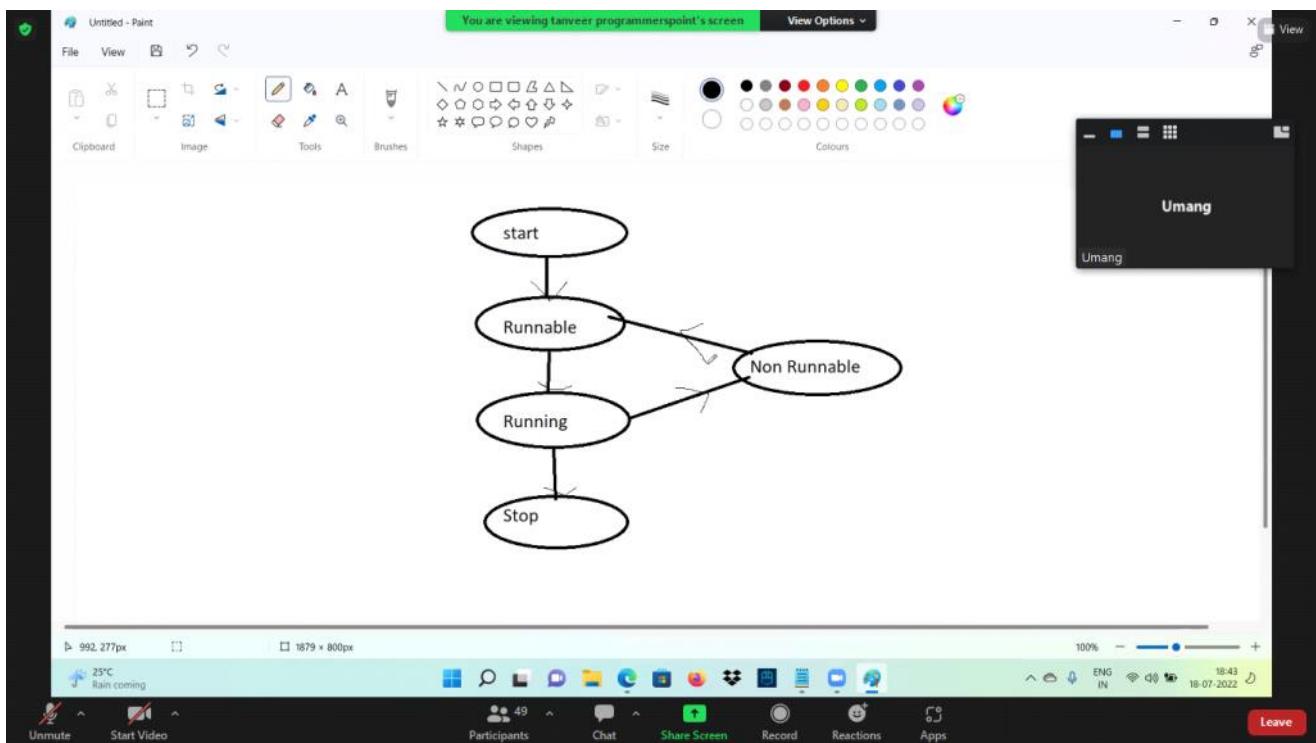
Multi- threading

Collection of threads is process, and collection of process is software.

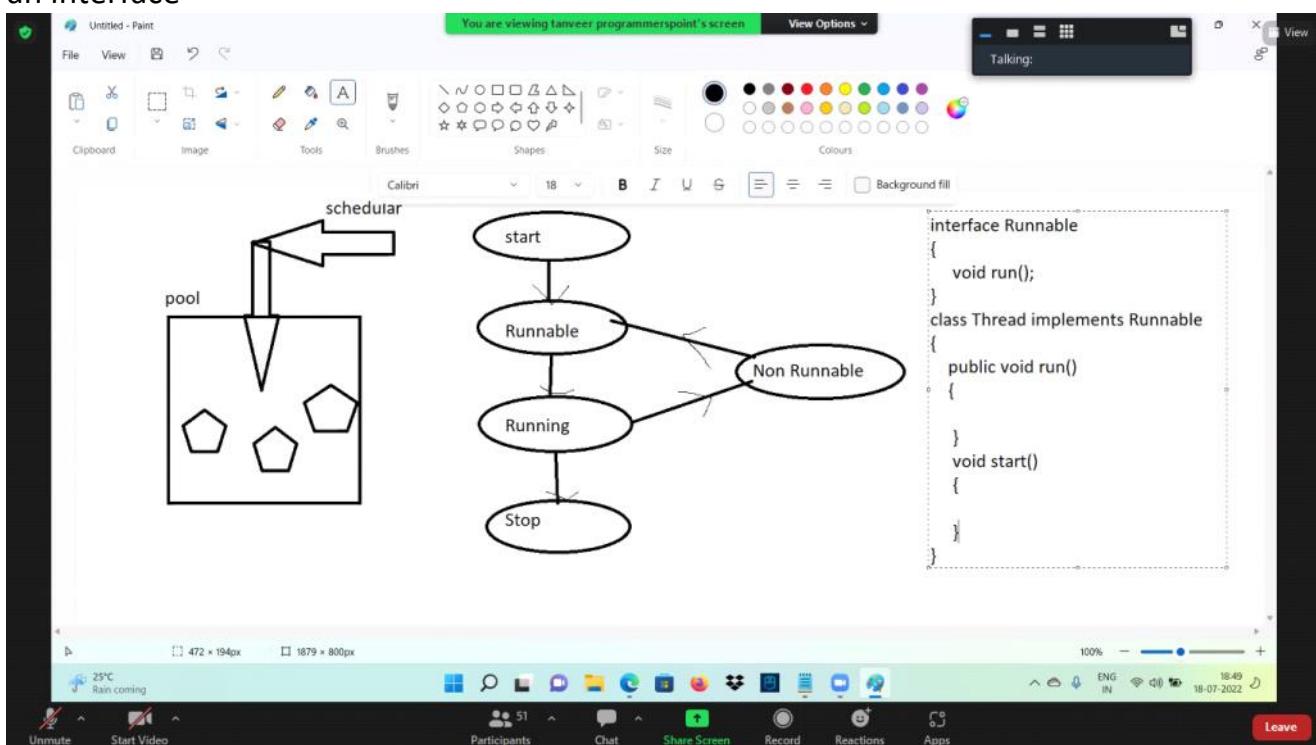
Thread :- Thread is a lightweight process or smallest unit of process, but in Java thread is a class which is stored in `java.lang`.

package.

Lifecycle of threads contains following states



(Thread scheduler and thread pool) Thread is a class and Runnable is an interface



**JVM will call run method automatically when it will go from runnable to running.

Ways to create a thread in java

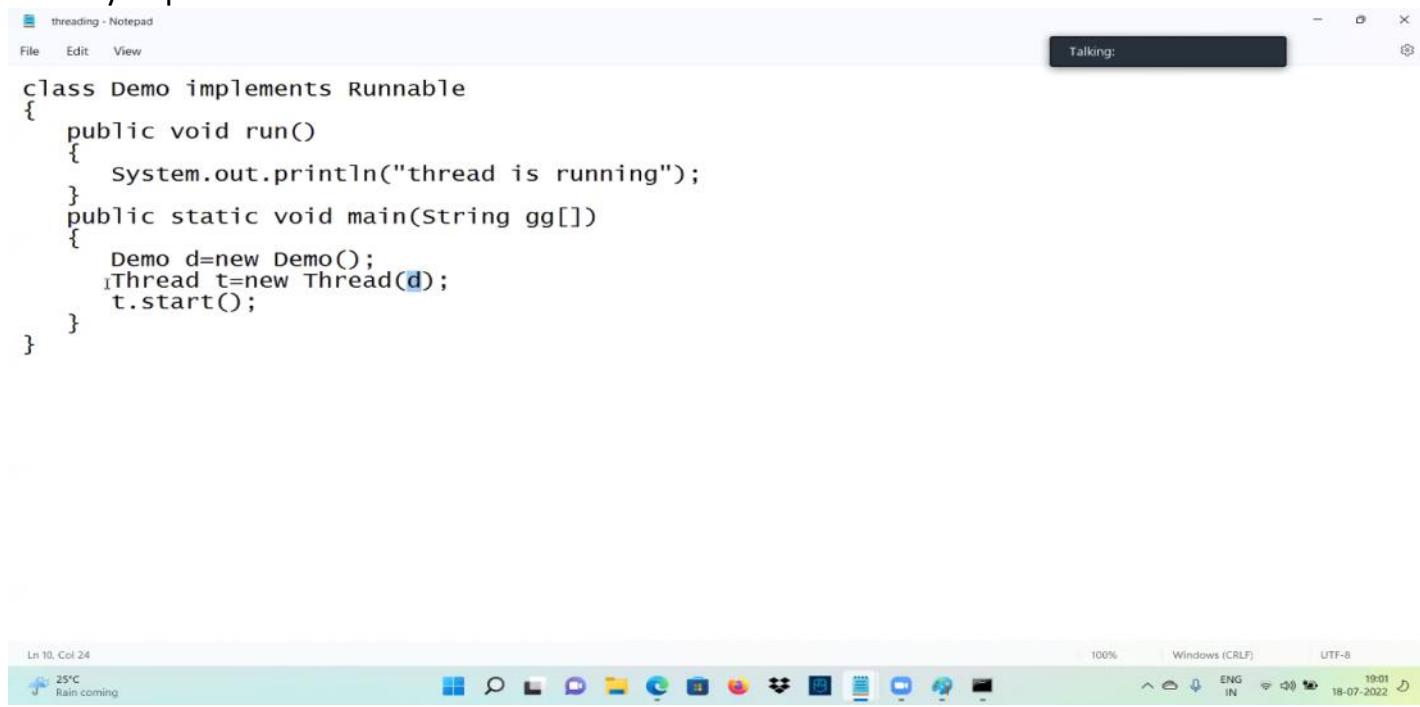
We can create a java thread with the help of following two ways:-

1st-By extends thread class



```
class Demo extends Thread
{
    public void run()
    {
        System.out.println("thread is running");
    }
    public static void main(String gg[])
    {
        Demo d=new Demo();
        d.start();
    }
}
```

2nd:-By implements runnable interface



```
class Demo implements Runnable
{
    public void run()
    {
        System.out.println("thread is running");
    }
    public static void main(String gg[])
    {
        Demo d=new Demo();
        Thread t=new Thread(d);
        t.start();
    }
}
```

Here reference variable of Demo class is passed in constructor of thread class so that run method is called and run by Demo class and not of Thread class as its run method is Empty

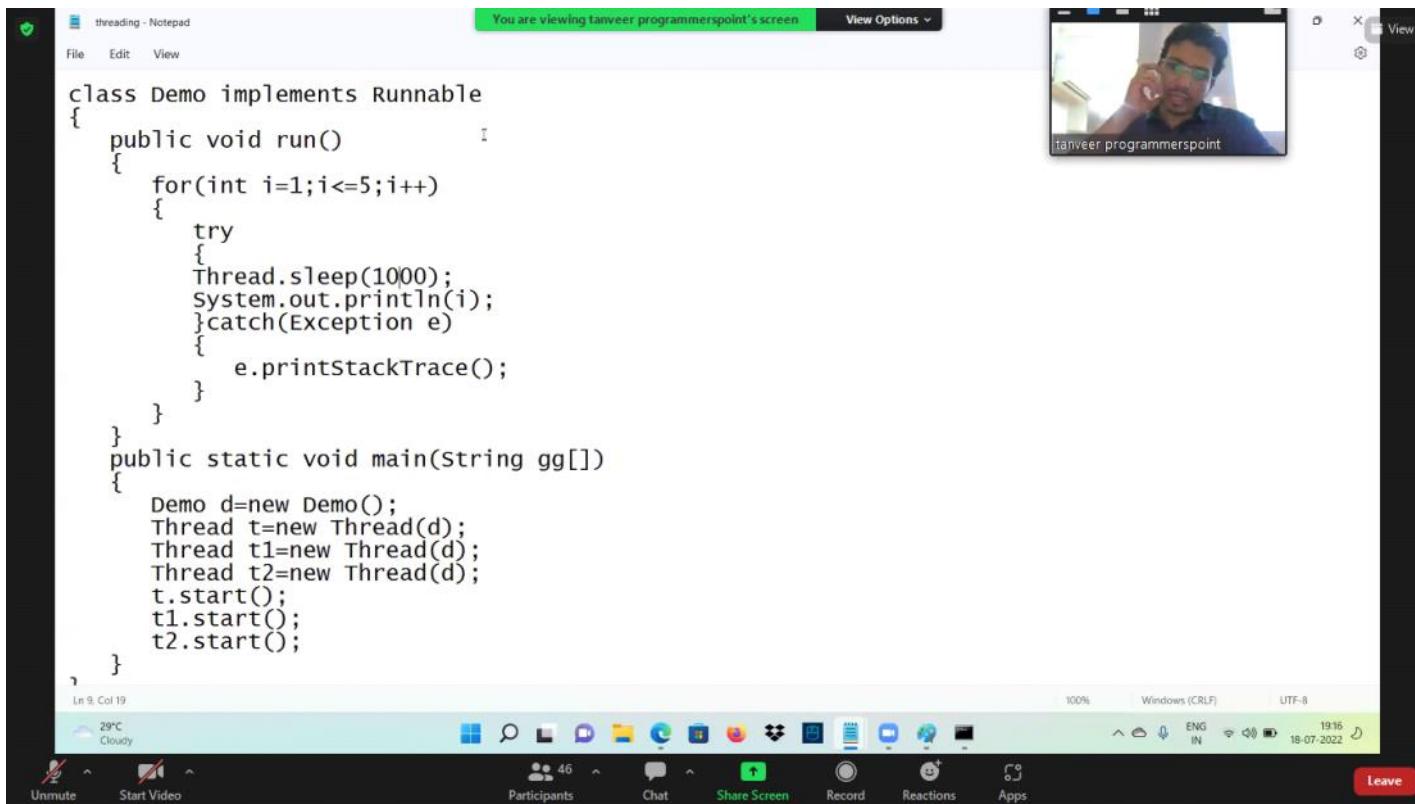
Setter and Getter name methods for Thread

```
threading - Notepad
File Edit View Talking:
class Demo implements Runnable
{
    public void run()
    {
        Thread a=Thread.currentThread();
        System.out.println(a.getName());
    }
    public static void main(String gg[])
    {
        Demo d=new Demo();
        Thread t=new Thread(d);
        Thread t1=new Thread(d);
        Thread t2=new Thread(d);
        t.setName("one");
        t1.setName("two");
        t2.setName("three");
        t.start();
        t1.start();
        t2.start();
    }
}
```



Sleep method of thread class

We can achieve multithreading in java with the help of sleep method of Thread class, it's a static method that contains time in milliseconds and throws interrupted checked exception.
So when we will call sleep method of thread class then we will surround it within try and catch block



Join method of thread class:-

In multithreading environment if we want to execute 1st thread separately and remaining all threads simultaneously we will use join method of thread class, it also throws interrupted exception so we will surround it with try and catch

```
threading - Notepad
File Edit View
You are viewing tanveer programmerspoint's screen View Options
Talking:
{
    Thread.sleep(1000);
    System.out.println(i);
}catch(Exception e)
{
    e.printStackTrace();
}
}
public static void main(String gg[])
{
    Demo d=new Demo();
    Thread t=new Thread(d);
    Thread t1=new Thread(d);
    Thread t2=new Thread(d);
    t.start();
    try
    {
        t.join();
    }catch(Exception e)
    {
        e.printStackTrace();
    }
    t1.start();
    t2.start();
}
}
In 33:00 Unmute Start Video Participants 43 Chat Share Screen Record Reactions Apps Leave
```

Synchronization

Synchronization is a process of multithreading in which we will execute each and every thread separately .We can achieve synchronization either by method or by block.

Synchronization by Method/Synchronization method

We can define method by **synchronized** keyword

Either we can create static method or we can create non static method

threading - Notepad

You are viewing tanveer programmerspoint's screen

View Options

Talking: tanveer programmersp...

```
class Demo
{
    synchronized static void show(int n)
    {
        for(int i=1;i<=5;i++)
        {
            try
            {
                Thread.sleep(1000);
                System.out.println(i+n);
            }catch(Exception e)
            {
                e.printStackTrace();
            }
        }
    }
}
class Psp1 implements Runnable
{
    public void run()
    {
        Demo.show(10);
    }
}
class Psp2 implements Runnable
{
    public void run()
    {
        Demo.show(100);
    }
}
```

Unmute Start Video Participants Chat Share Screen Record Reactions Apps

100% Windows (CPU) LITE-B Leave

threading - Notepad

You are viewing tanveer programmerspoint's screen

View Options

Swapnil Totani

Press ESC or double-click to exit full screen mode

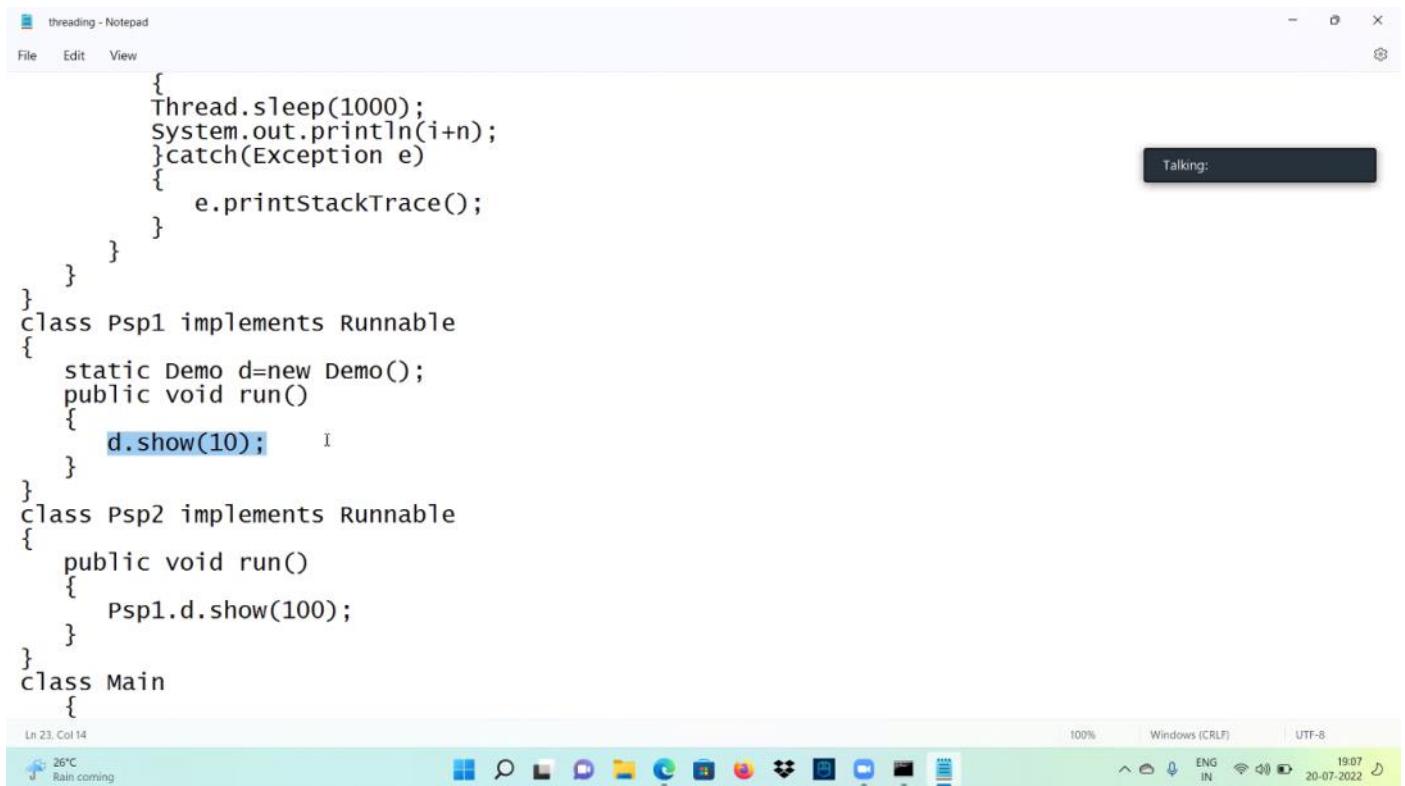
```
}
```

```
class Psp1 implements Runnable
{
    public void run()
    {
        Demo.show(10);
    }
}
class Psp2 implements Runnable
{
    public void run()
    {
        Demo.show(100);
    }
}
class Main
{
    public static void main(String gg[])
    {
        Psp1 p1=new Psp1();
        Psp2 p2=new Psp2();
        Thread t1=new Thread(p1);
        Thread t2=new Thread(p2);
        t1.start();
        t2.start();
    }
}
```

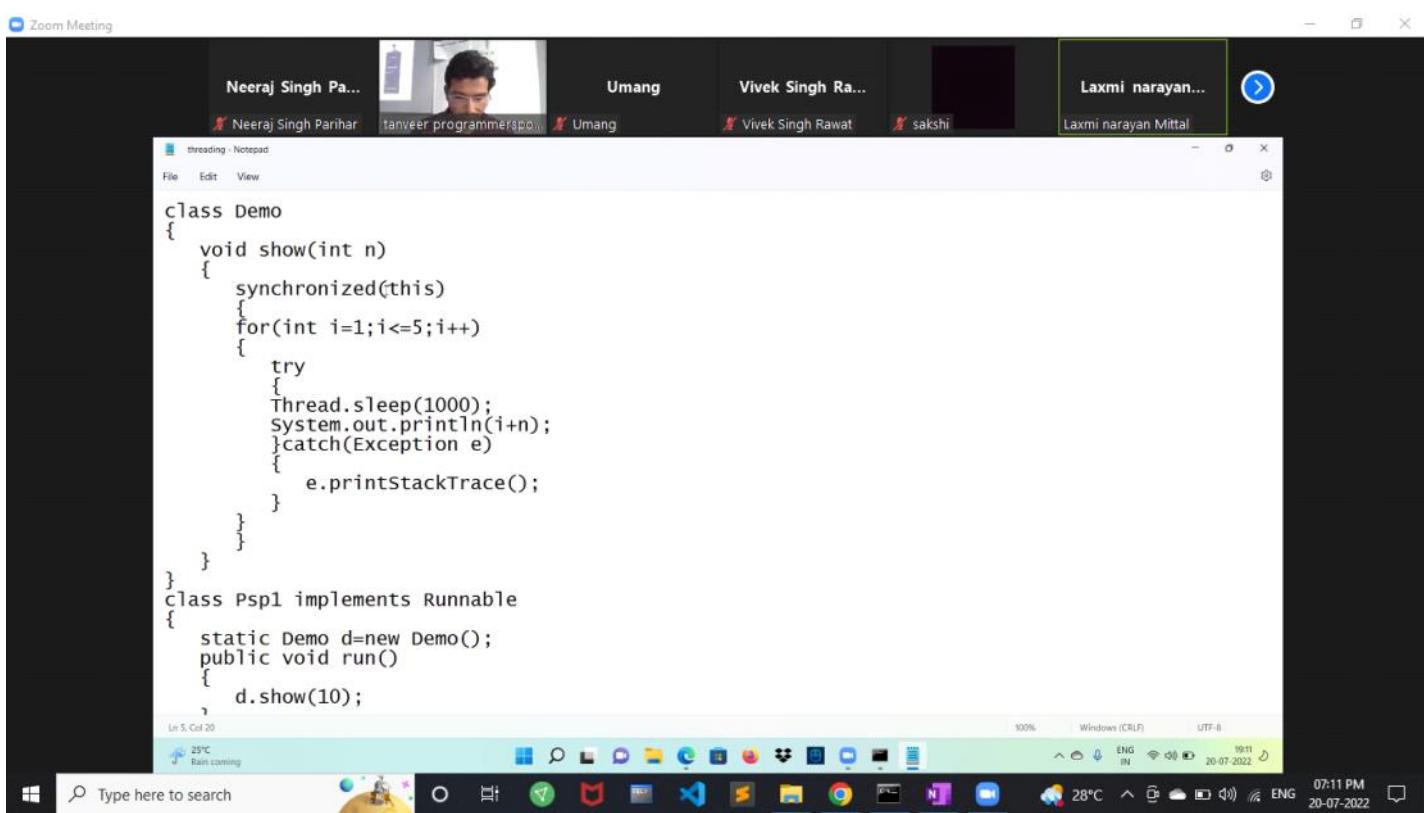
Unmute Start Video Participants Chat Share Screen Record Reactions Apps

100% Windows (CPU) LITE-B Leave

By non static method



```
threading - Notepad
File Edit View
{
    Thread.sleep(1000);
    System.out.println(i+n);
}catch(Exception e)
{
    e.printStackTrace();
}
}
}
class Psp1 implements Runnable
{
    static Demo d=new Demo();
    public void run()
    {
        d.show(10);
    }
}
class Psp2 implements Runnable
{
    public void run()
    {
        Psp1.d.show(100);
    }
}
class Main
{
}
Ln 23, Col 14
Windows (CRLF) UTF-8
100% 26°C Rain coming 19:07
Zoom Meeting 20-07-2022
```



```
threading - Notepad
File Edit View
class Demo
{
    void show(int n)
    {
        synchronized(this)
        {
            for(int i=1;i<=5;i++)
            {
                try
                {
                    Thread.sleep(1000);
                    System.out.println(i+n);
                }catch(Exception e)
                {
                    e.printStackTrace();
                }
            }
        }
    }
}
class Psp1 implements Runnable
{
    static Demo d=new Demo();
    public void run()
    {
        d.show(10);
    }
}
class Psp2 implements Runnable
{
    public void run()
    {
        Psp1.d.show(100);
    }
}
Ln 5, Col 20
Windows (CRLF) UTF-8
100% 25°C Rain coming 19:11
Zoom Meeting 20-07-2022
```

Garbage collection

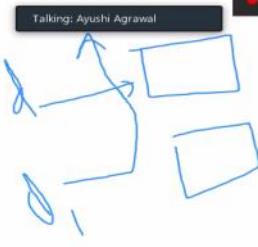
gt - Notepad

You are viewing tanveer.programmerspoint's screen

File Edit View Options

00:12:36

Talking: Ayushi Agrawal

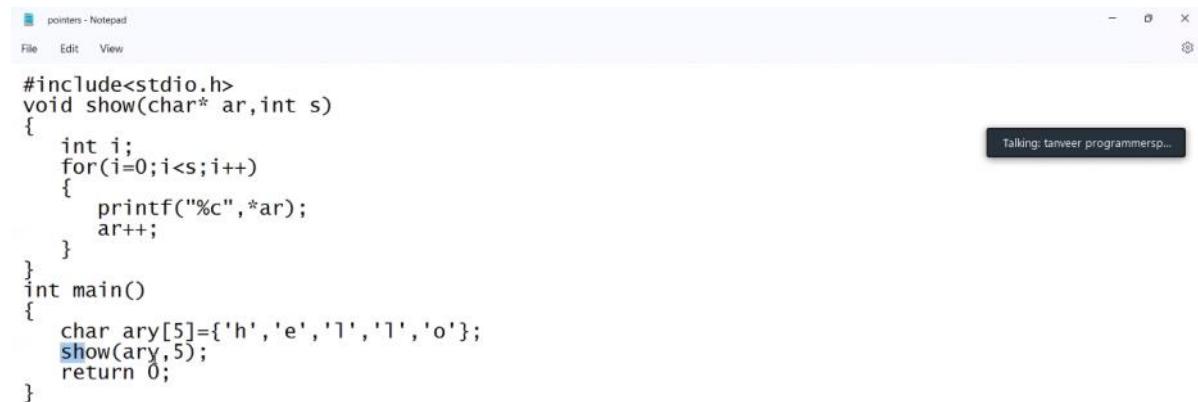


```
class Demo
{
    public void finalize()
    {
        System.out.println("object gone");
    }
    public static void main(String gg[])
    {
        Demo d=new Demo();
        Demo d1=new Demo();
        d=null;
        d1=d;
        System.gc();
    }
}
```



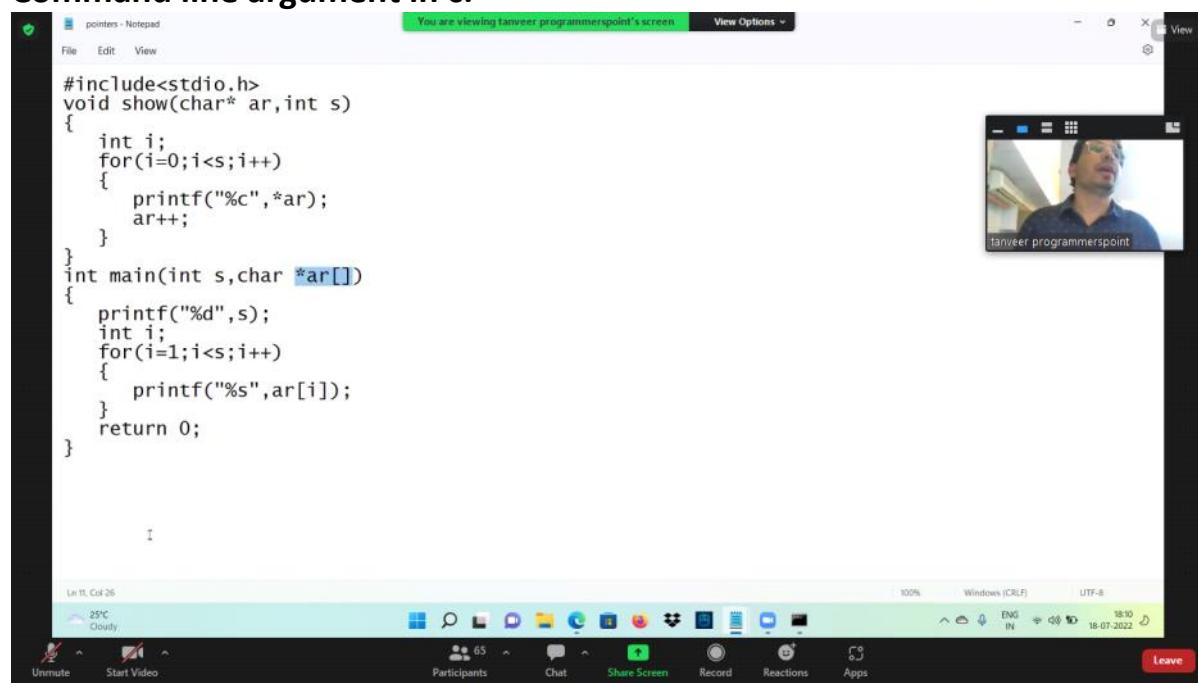
Pointer in arguments of a function

:-



```
#include<stdio.h>
void show(char* ar,int s)
{
    int i;
    for(i=0;i<s;i++)
    {
        printf("%c",*ar);
        ar++;
    }
}
int main()
{
    char ary[5]={'h','e','l','l','o'};
    show(ary,5);
    return 0;
}
```

Command line argument in c:-



```
#include<stdio.h>
void show(char* ar,int s)
{
    int i;
    for(i=0;i<s;i++)
    {
        printf("%c",*ar);
        ar++;
    }
}
int main(int s,char *ar[])
{
    printf("%d",s);
    int i;
    for(i=1;i<s;i++)
    {
        printf("%s",ar[i]);
    }
    return 0;
}
```

A function can return a pointer

You are viewing tanveer programmerspoint's screen View Options

```
#include<stdio.h>
int* show()
{
    int *p,a=10;
    p=&a;
    return p;
}
int main()
{
    int *p1;
    p1=show();
    printf("value is %d",*p1);
    return 0;
}
```

Ln 9, Col 26

26°C Rain coming Participants Chat Share Screen Record Reactions Apps Unmute Start Video 52 100% Windows (CRLF) UTF-8 ENG IN 17:50 20-07-2022 Leave

Array of pointer

You are viewing tanveer programmerspoint's screen View Options

```
#include<stdio.h>
int main()
{
    int ary[5]={1,2,3,4,5},i;
    int *paray[5];
    for(i=0;i<5;i++)
    {
        paray[i]=&ary[i];
        printf("%d\n",*paray[i]);
    }
    return 0;
}
```

1 2 3 4 5
↓↑↓↑↓↑
1 2 3 4 5
↓↑↓↑↓↑
3 4 5 1 2
↓↑↓↑↓↑
4 5 1 2 3
↓↑↓↑↓↑
5 1 2 3 4
↓↑↓↑↓↑

Ln 9, Col 15

26°C Rain coming Participants Chat Share Screen Record Reactions Apps Unmute Start Video 53 100% Windows (CRLF) UTF-8 ENG IN 18:04 20-07-2022 Leave

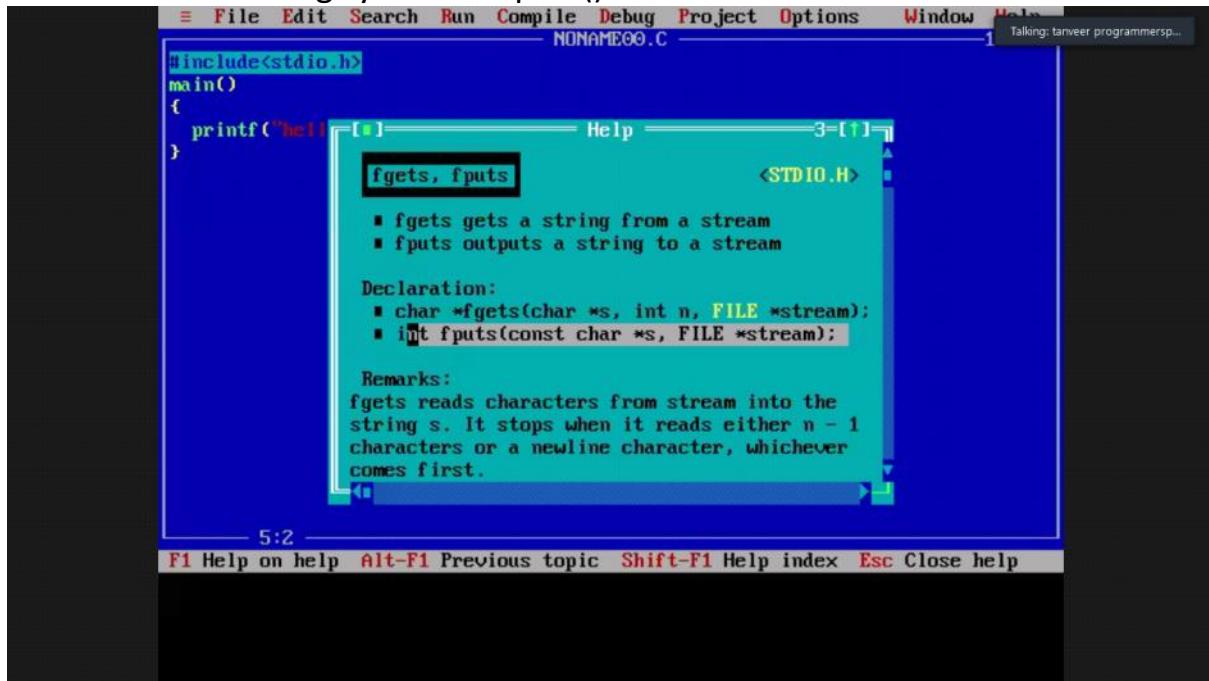
***Array of array is directly not possible in C ,but with the help of pointers array in C we can store the base address of multiple array in indexes of pointer array and can traverse from there.

String:- Internally string is an character array, in this string we can store values, and with the help of string functions we can perform n number of operations on string.
<String.h> header file provides us some functions ,through which we can calculate length, reverse, concatenation etc operation on string , some commonly used function are:-

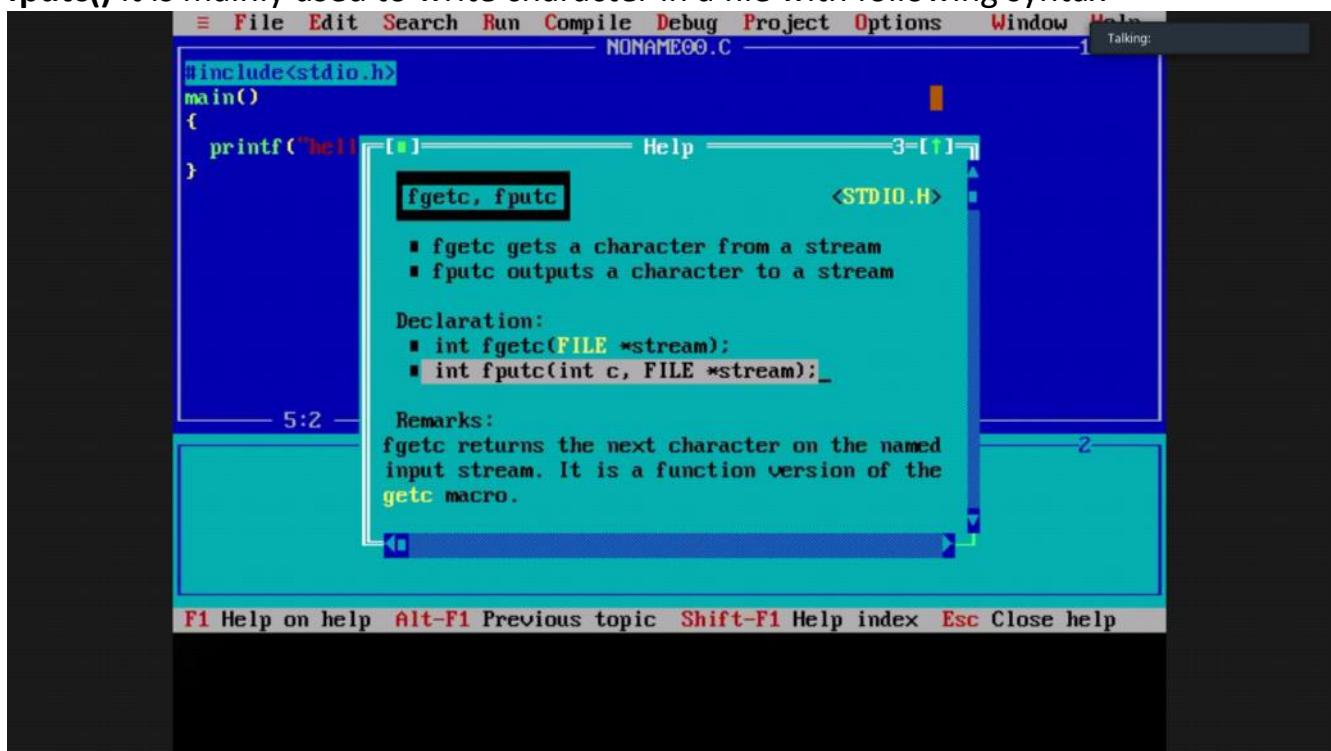
strlen() function returns length of string as a integer
strupr() function convert a string into upper case
strlwr() convert a string into lower case
Strcpy() is used to copy one string into another string
Gets

Functions for writing data in a file in file handling:->
fprintf() it is mainly used to write data in a single file, we will use following syntax for

fputs() function is mainly used to write a string into files
We will use following syntax for fputs()



fputc() it is mainly used to write character in a file with following syntax



Reading file function

fscanf() it is mainly used to read data from file with following syntax



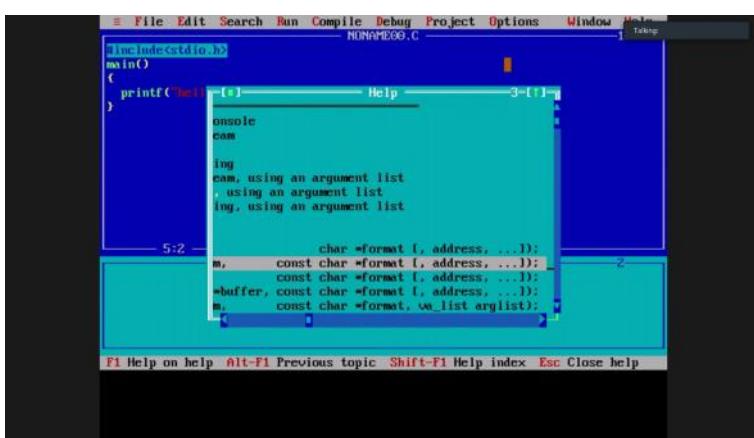
```
#include<stdio.h>
main()
{
    printf("hell");
}
```

The help menu shows the following declarations for fscanf():

- cscanf COMIO.H The console
- fscanf STDIO.H A stream
- scanf STDIO.H stdio
- sscanf STDIO.H A string
- vscanf STDIO.H A stream, using an argument
- vsprintf STDIO.H stdio, using an argument
- vsscanf STDIO.H A string, using an argument

Declaration:

- int cscanf (char
- int fscanf (FILE *stream, const char
- int scanf (const char
- int sscanf (const char *buffer, const char
- int vsprintf (FILE *stream, const char

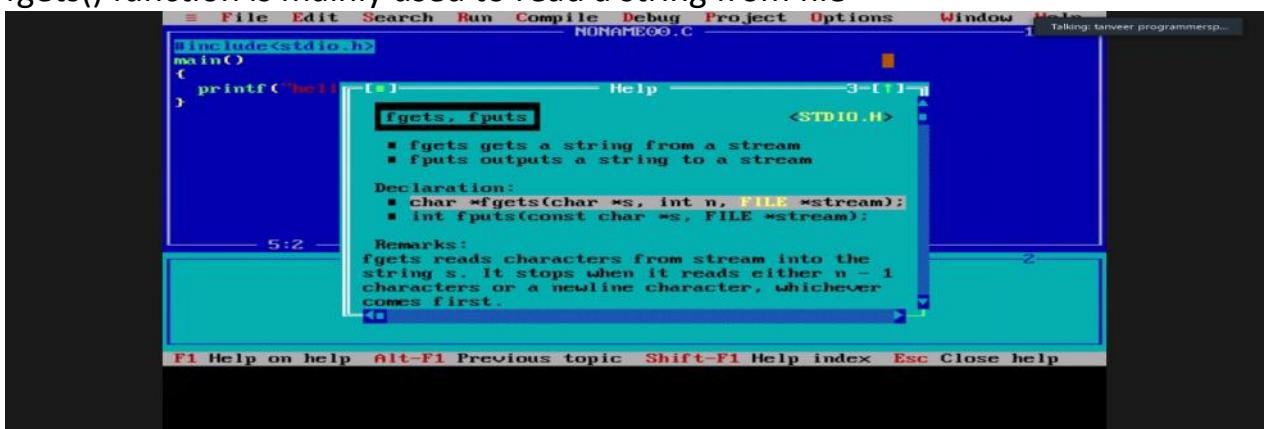


```
#include<stdio.h>
main()
{
    printf("hell");
}
```

The help menu shows the following declarations for fscanf():

- cscanf
- console
- com
- ing
- eam, using an argument list
- , using an argument list
- ing, using an argument list
- char *format [, address, ...]);
- m, const char *format [, address, ...]);
- const char *format [, address, ...]);
- buffer, const char *format [, address, ...]);
- m, const char *format, va_list arglist);

fgets() function is mainly used to read a string from file



```
#include<stdio.h>
main()
{
    printf("hell");
}
```

The help menu shows the following information for fgets():

- fgets, fputs**
- fgets gets a string from a stream
- fputs outputs a string to a stream

Declaration:

- char *fgets(char *s, int n, FILE *stream);
- int fputs(const char *s, FILE *stream);

Remarks:

- fgets reads characters from stream into the string s. It stops when it reads either n - 1 characters or a newline character, whichever comes first.

Fgetc() function is mainly used to read character from file

The screenshot shows a video conference interface with a terminal window in the foreground. The terminal displays a C program:#include<stdio.h>
main()
{
 printf("hell");
}A help window for the `fgetc` function is open, showing its declaration, description, and remarks. The remarks note that `fgetc` is a function version of the `getchar` macro.

F1 Help on help Alt-F1 Previous topic Shift-F1 Help index Esc Close help

Rewind function is mainly used to put file pointer at beginning location of file.
Ftell function returns current pointer position of file

The screenshot shows a video conference interface with a terminal window in the foreground. The terminal displays a C program:#include<stdio.h>
main()
{
 printf("hell");
}A help window for the `ftell` function is open, showing its declaration, description, and remarks. The remarks note that `ftell` returns the current file pointer for the stream. It also specifies that if the file is binary, the offset is measured in bytes from the beginning of the file.

F1 Help on help Alt-F1 Previous topic Shift-F1 Help index Esc Close help

Delete is used to delete one row ,specified row or complete row from table.

Truncate is used to delete all rows of table but it will not delete structure of a table*We can't use where with truncate.

Drop is used to delete data as well as structure of a table.

Alter is mainly used to deal with columns ,with the help of alter we can add ,drop modify columns.

Constraints are used to apply restriction/Filtration on table. We can enter filtered data with the help of constraints

JAVA DATABASE CONNECTIVITY

Tight coupling:-When a class object is dependent only on one another class object

&

Loose coupling:-When a class object doesn't depend on single class it has many classes that ie like an interface, different classes can implements that interface and interdepend ability is removed.



```
dbc - Notepad
File Edit View
}
class DriverManager
{
    static Connection getConnection(String url)
    {
        if(url.equals("mysql"))
        {
            return new Mysql();
        }
        else if(url.equals("oracle"))
        {
            return new Oracle();
        }
        else
        {
            return new Access();
        }
    }
}
class Demo
{
    public static void main(String gg[])
    {
        Connection con=DriverManager.getConnection("oracle");
        con.createStatement();
    }
}
```

Participants: 44 Chat Share Screen Record Reactions Apps Leave



```
"Untitled - Notepad
File Talking:
You are viewing tanveer programmerspoint's screen
View Options
File Edit View
interface Connection
{
    void createStatement();
}
class Mysql implements Connection
{
    public void createStatement()
    {
        System.out.println("mysql support query");
    }
}
class Oracle implements Connection
{
    public void createStatement()
    {
        System.out.println("oracle support query");
    }
}
class Access implements Connection
{
    public void createStatement()
    {
        System.out.println("access support query");
    }
}

class DriverManager
{
    static Connection getConnection(String url)
    {
        if(url.equals("mysql"))
        {
            return new Mysql();
        }
        else if(url.equals("oracle"))
        {
            return new Oracle();
        }
        else
        {
            return new Access();
        }
    }
}
```

Ln 23, Col 43 Windows (CRLF) 100% Participants Chat Share Screen Record Reactions Apps Leave



```
dbc - Notepad
File Edit View
}
class DriverManager
{
    static Connection getConnection(String url)
    {
        if(url.equals("mysql"))
        {
            return new Mysql();
        }
        else if(url.equals("oracle"))
        {
            return new Oracle();
        }
        else
        {
            return new Access();
        }
    }
}
class Demo
{
    public static void main(String gg[])
    {
        Connection con=DriverManager.getConnection("oracle");
        con.createStatement();
    }
}
```

Ln 23, Col 2 Windows (CRLF) 100% Participants Chat Share Screen Record Reactions Apps Leave

Jar files:- Compressed format of a package is known as jar(java archives)

Steps for JDBC connectivity

1st-Load the driver.

Driver is a class in java

When we will call static `forName()` method of `java run time Class` that contains full qualified name of Driver class and it will throws `ClassNotFoundException` checked Exception, to load the JDBC Driver

```
Class.forName("com.mysql.jdbc.Driver");
```

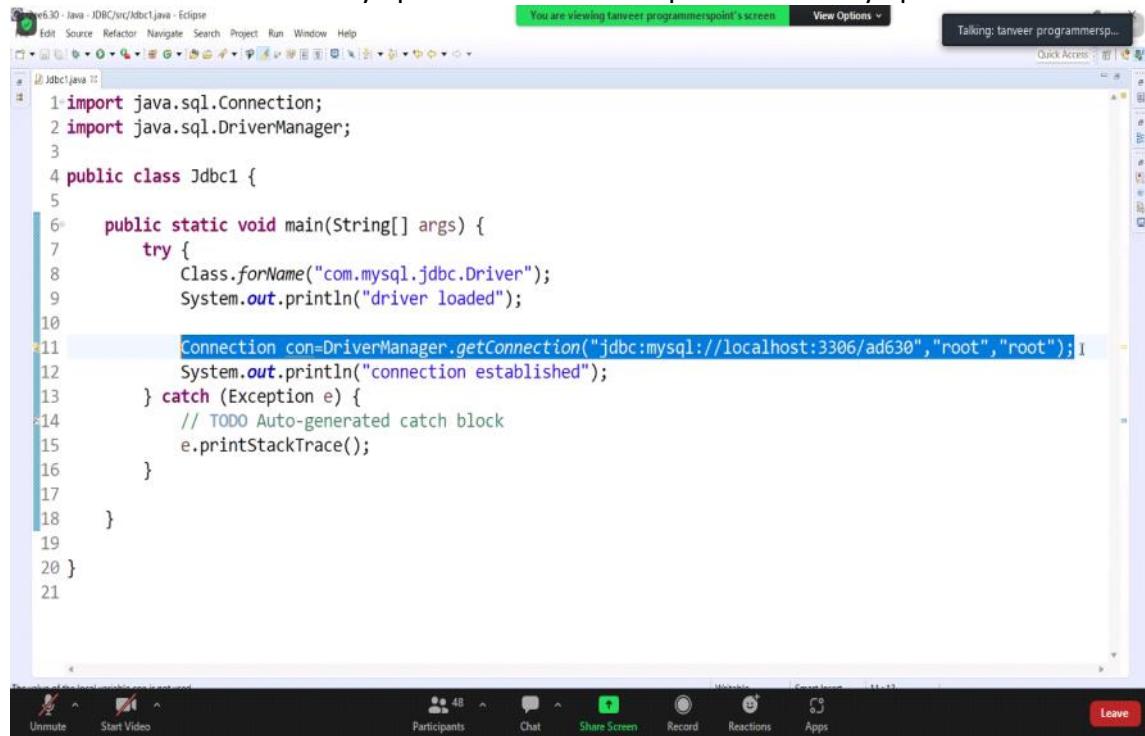
`com.mysql.jdbc.Driver` is package name and `Driver` is class name

2nd:-Establish Connection

As we know connection is an interface and when we will call `getConnection()` method of `DriverManager` class that contains three arguments , 1st database connection url ,2 nd username,3rd password of mysql.

Here localhost is servername, 3306 is port number for mysql,ad630 is database name.

1st root is username for mysql and second root is password of mysql.



The screenshot shows the Eclipse IDE interface with a Java file named `Jdbc1.java` open. The code is as follows:

```
1 import java.sql.Connection;
2 import java.sql.DriverManager;
3
4 public class Jdbc1 {
5
6     public static void main(String[] args) {
7         try {
8             Class.forName("com.mysql.jdbc.Driver");
9             System.out.println("driver loaded");
10
11             Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ad630","root","root");
12             System.out.println("connection established");
13         } catch (Exception e) {
14             // TODO Auto-generated catch block
15             e.printStackTrace();
16         }
17     }
18 }
19
20 }
21 }
```

3rd:-insert a query as a String

Statement interface

Statement is an interface which is mainly used to execute non parametrized query on database. When we will call `createStatement()` method at `con` reference variable it will return

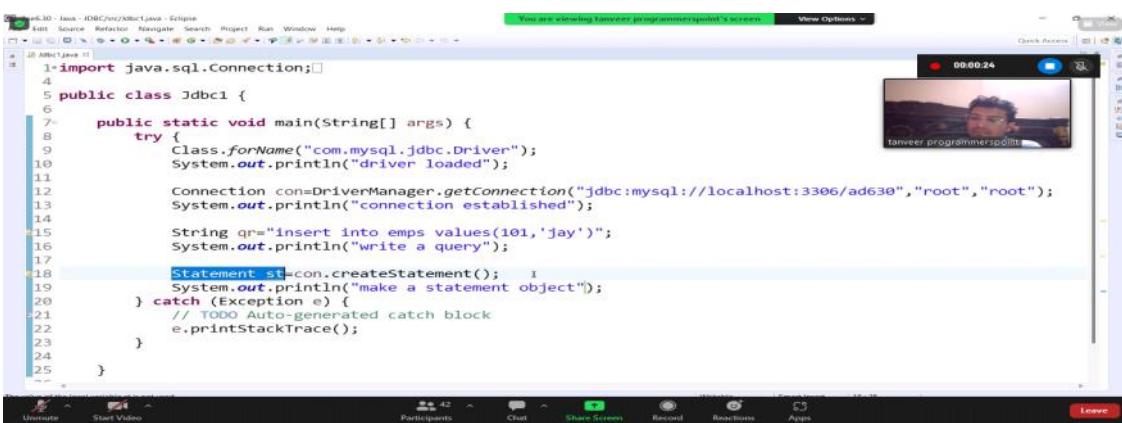
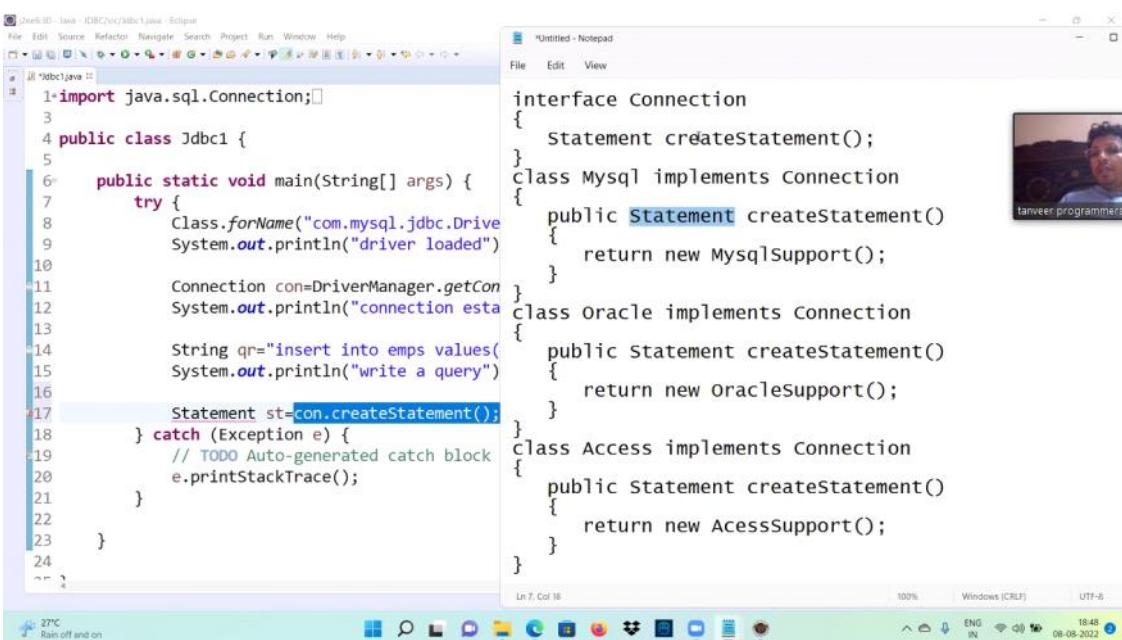
Implementations class object of Statement interface

Commonly used method of statement interface are



```
Statement st=con.createStatement();

boolean execute(String); //ddl{create table,drop table,alter table,truncate table}
int executeUpdate(String);//ddl&dml{insert,update,delete}
ResultSet executeQuery(String){select}
```





Two Notepad windows side-by-side. The left window shows code for Statement interface and its implementations for MySQL, Oracle, and Access. The right window shows code for Connection interface and its implementations for MySQL, Oracle, and Access.

```

interface Statement
{
    boolean execute();
    int executeUpdate();
    ResultSet executeQuery();
}

class MysqlSupport implements Statement
{
    its implements all methods and write
    code to support mysql queries
}

class OracleSupport implements Statement
{
    its implements all methods and write
    code to support oracle queries
}

class AccessSupport implements Statement
{
    its implements all methods and write
    code to support access queries
}

```

```

interface Connection
{
    Statement createStatement();
}

class Mysql implements Connection
{
    public Statement createStatement()
    {
        return new MysqlSupport();
    }
}

class Oracle implements Connection
{
    public Statement createStatement()
    {
        return new OracleSupport();
    }
}

class Access implements Connection
{
    public Statement createStatement()
    {
        return new AccessSupport();
    }
}

```



A single Notepad window showing the Statement interface with its methods: execute, executeUpdate, and executeQuery. It also includes a note about DDL (create, drop, alter, truncate) and DML (insert, update, delete).

```

Statement st=con.createStatement();

boolean execute(String); //ddl{create_table,drop_table,alter_table,truncate_table}
int executeUpdate(String); //ddl&dm{linsert,update,delete}
ResultSet executeQuery(String); {select}

```



A screenshot of an Eclipse IDE showing a Java file named Abc1.java. The code demonstrates JDBC connection, statement creation, and execution of an insert query. A video call interface is overlaid on the bottom right.

```

public static void main(String[] args) {
    try {
        Class.forName("com.mysql.jdbc.Driver");
        System.out.println("driver loaded");
        Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/ad630","root","root");
        System.out.println("connection established");

        String qr="insert into emps values(101,'jay')";
        System.out.println("write a query");

        Statement st=con.createStatement();
        System.out.println("make a statement object");

        int i=st.executeUpdate(qr);
        System.out.println(i+" record added");

        con.close();
        System.out.println("close the connection");
    } catch (Exception e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}

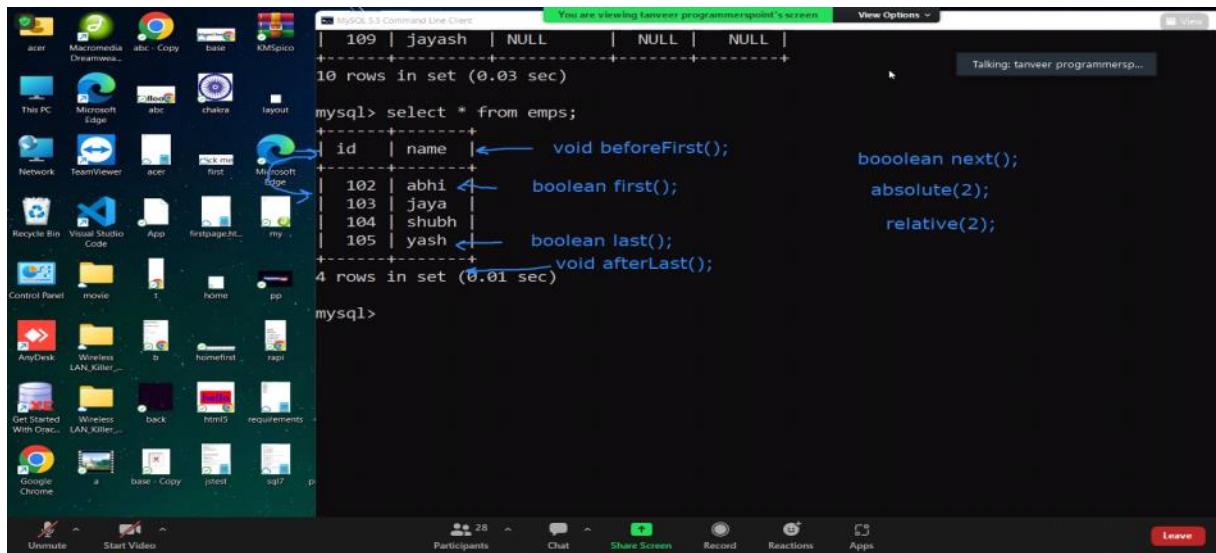
```

When we will take values from user, then query should be considered parametrized query and for parametrized query we will use prepared statement instead of statement

When we will call executeQuery method on preparedstatement or statement reference variable it will return implementation class of ResultSet. ResultSet is an interface which is used to select query on interface

It contains all the getter method corresponding to primitives and strings and some commonly used method of ResultSet interface are

Commonly used method of ResultSet interface



Construct a student table in which we will put following tables

Name ,roll no, semester, year

Add, delete, update, student

Search by rollno, search by branch ,search by year, search by sem

React

16 August 2022 11:14 AM

React is a JavaScript library which is mainly used to construct light weight easy to access fast and easy testable application . React uses Virtual Dom instead of physical Dom that's why it will create a fast application

NPM/npm :-Node package manager
NVM/nvm Node version manager

To install react:-npm install create-react-app

To install Angular- npm install@angular/cli (cli:-command line interface)

First of all we will open terminal or command prompt then we will execute following commands

Type node -v to check node version

We will install react in our project by following command

npm create-react-app

Create-react-app --version (command to check version)

We can create project by command

Create-react-app project_name

create-react-app --version

After creation of project you will get a message happy hacking

Then open folder of the created project

We can run our project with the help of following command

npm start

Folder Structure for react app:-

1st package.json file :-it is one of the most important file in which react define react version , react dom version , web version

All the supporting scripts and dependencies for our react project

Readme.md file:- in this file we will define description of our react project

.gitignore:- in this .gitignore file we will write command to ignore when we will push our project on github.

Folders that contains file

node_module package contains all the necessary modules which we can install in our project by using npm commands

Public folder:- public folder contains all the files like logo, index and manifest and we will also store all the images in public folder

index.html file in public folder:- it is the root file through which react will render everything in dom

src folder:- it contains all the important file for react project

index.js :- it is a root file for react , from this file we will call all the components. Usually we will call app component from

This file and app is parent component for all react child component

index.css:- in this index.css file we will write css content for index.html page, it means in this file we will write all the Common css contents

app.js :-App.js file contains app component and in app component we will call the child component using jsx

JavaScript xml or JavaScript extensions xml(Extensible markup language)

App.css :- in app.css file we will write css contain for app component .

App.test.js:- in this file we will write test cases

JavaScript with xml or jsx(JavaScript with extensions JSML)

React uses jsx to render a component in dom with the help of bable js we can convert jsx into js

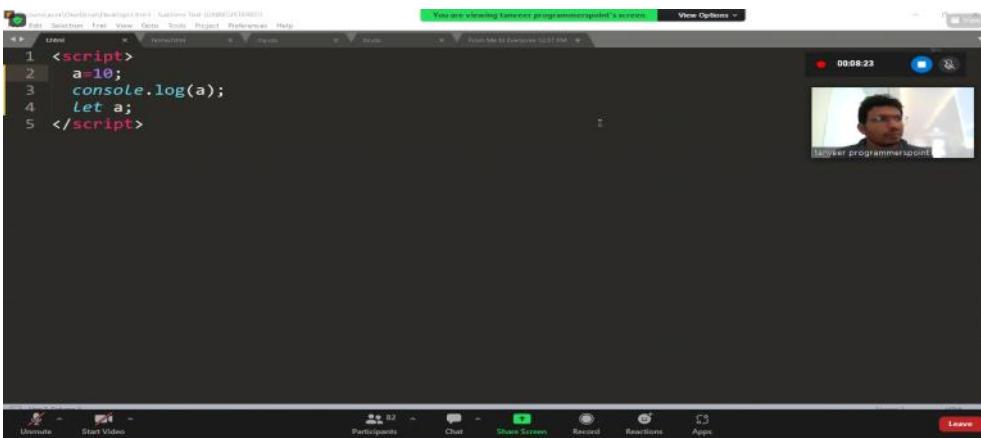
Class base component:- used in earlier version of react in which we will used some life cycle method(mounting ,updating and unmounting)

Functional component:- it is the one of the most popular technique to create component in react, now a days react uses a pure functional component and oops instead of class base components

hoisting

But when we will declare variable with var keyword

Hoisting is a memory area first of all in which JavaScript will allocate memory for those variables which is defined by Var keyword



React component:- React component is used to perform some important operations inside a component , and through which we Can perform some important operations in our react project.

A react project is collection of components and each and every react component is used to perform its own task.

In react we can construct 2 types of components

1st:- Class Based Component

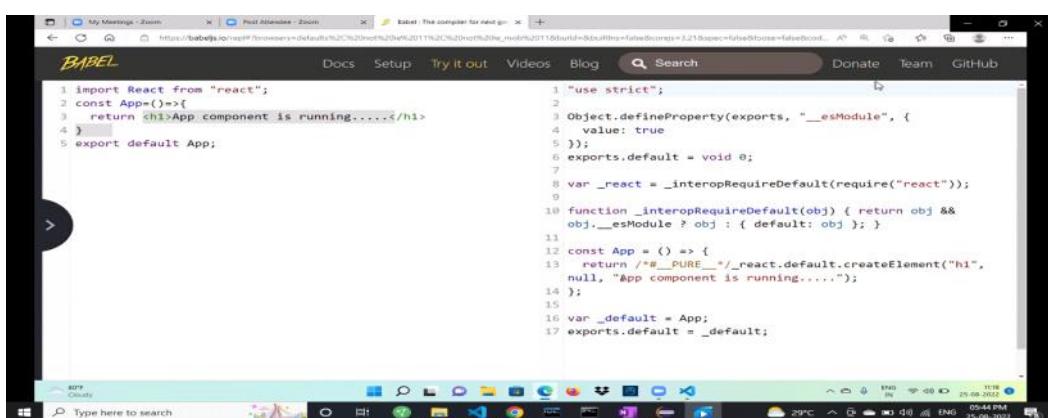
2nd:-Functional components

A Functional component is a component that uses some react hooks and return multiple jsx elements

App.js contains app component and app is parent or root component for react

In react component is generally a function. Every function is treated as component

React js with the help of bable js converts the html code written in component (which is not html code it is JSX /JSXML)into js object



App component

App component

```
return <h1>App Component is running...</h1>
```

1

```
export default App;
```

If you want to render multiple jsx elements from a react app then we will use either division or react.fragments or Syntactic sugar form of react fragments

```

File Edit Selection View Go Run Terminal Help
App.js - first - Visual Studio Code
EXPLORER JS App.js JS index.js
FIRST > node_modules > public > src # App.css JS App.js JS App.test.js # index.css JS index.js # logo.svg JS reportWebVitals.js JS setupTests.js # .gitignore () package-lock.json () package.json () README.md
src > JS App.js > (e) App
1 import React from "react";
2 const App=()=>{
3   return(
4     <div>
5       <h1>App component is running</h1>
6       <p>this is para</p>
7     </div>
8   )
9 }
10 export default App;
11

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Ln 9, Col 2 Spaces: 2 UTF-8 LF JavaScript Prettier

Type here to search

In react we don't use division as it reduce the performance of app instead we use react fragments.
React fragment is used to return multiple JSX elements from a component.

```

File Edit Selection View Go Run Terminal Help
App.js - first - Visual Studio Code
EXPLORER JS App.js JS index.js
FIRST > node_modules > public > src # App.css JS App.js JS App.test.js # index.css JS index.js # logo.svg JS reportWebVitals.js JS setupTests.js # .gitignore () package-lock.json () package.json () README.md
src > JS App.js > (e) App
1 import React from "react";
2 const App=()=>{
3   return(
4     <React.Fragment>
5       <h1>App component is running</h1>
6       <p>this is para</p>
7     </React.Fragment>
8   )
9 }
10 export default App;
11

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Ln 4, Col 20 Spaces: 2 UTF-8 LF JavaScript Prettier

Type here to search

It improves performance of our react Application, but instead of React.Fragmance we will use its syntactic sugar form to increase application or component Security

```

File Edit Selection View Go Run Terminal Help
App.js - first - Visual Studio Code
EXPLORER JS App.js JS index.js
FIRST > node_modules > public ★ favicon.ico > index.html & logo192.png & logo512.png () manifest.json & robots.txt > src # App.css JS App.js JS App.test.js # index.css JS index.js # logo.svg JS reportWebVitals.js JS setupTests.js # .gitignore () package-lock.json > OUTLINE > TIMELINE
src > JS App.js > (e) App
1 import React from "react";
2 const App=()=>{
3   return(
4     <>
5       <h1>App component is running</h1>
6       <p>this is para</p>
7     </>
8   )
9 }
10 export default App;
11

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Note that the development build is not optimized.
To create a production build, use `npm run build`.

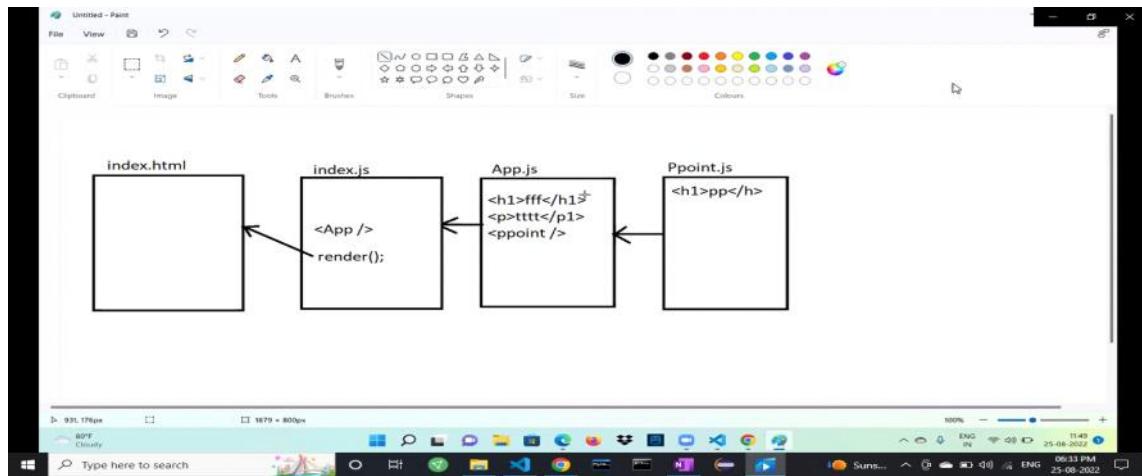
webpack compiled successfully

Ln 7, Col 8 Spaces: 2 UTF-8 LF JavaScript Prettier

Type here to search

Render Flow:-

As we know when we will construct our app component than our data will transfer to calling place of app component
We will call app component in index.js file, and from index.js file we will transfer app component and it's all child component To index.html file by using render method



Parent and child component:-

As we know component is collection of multiple jsx elements, and the root or parent component for react project is App component. We can construct n number of child components for react app and this all should be child of App component,
****Standard create new folder in src and name folder for component, n then create file either with js/jsx extension**

We can call any child component outside as tag with its name.

Import/Export:-

As we know a js/jsx file contains lot of components and if we want to access any component outside block or js file, then we will

export this component from current js file. We can export a component by using following two ways

- default export**:- when we will export a component with the help of default keyword then we can import this Component to any js file either by its original name or by any another name. **We can export only one component as a default**

- exporting component by its name**:- we can import component by name, but when we will export a component by name in any other js file we will export it by its name only. We can export more than 1 component by name.

```

// Child.jsx
import React from "react";
const Child=()=>{
  return <h1>Child Component is running</h1>
}
const Ppoint=()=>{
  return <h1>Ppoint is running</h1>
}
export default Child;
export {Ppoint};

// App.js
import React from "react";
const Child=()=>{
  return <h1>Child Component is running</h1>
}
const Ppoint=()=>{
  return <h1>Ppoint is running</h1>
}
const Ujjain=()=>{
  return <h1>Ujjain is running</h1>
}
const Indore=()=>{
  return <h1>Indore is running</h1>
}
export default Child;
export {Ppoint,Ujjain,Indore};

```

```
src > components > Child.js ...
1 import React from "react";
2 const Child=()=>{
3     return <h1>Child Component is running</h1>
4 }
5 const Ppoint=()=>{
6     return <h1>Ppoint is running</h1>
7 }
8 const Ujjain=()=>{
9     return <h1>Ujjain is running</h1>
10 }
11 const Indore=()=>{
12     return <h1>Indore is running</h1>
13 }
14 export default Child;
15 export {Ppoint,Ujjain,Indore};
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

```
src > JS App.js > [@] App
1 import React from "react";
2 //import Child from "./components/Child";
3 import Programmers from "./components/Child";
4 import { Ppoint,Ujjain,Indore } from "./components/Child";
5 const App=()=>{
6     return(
7         <>
8             <h1>App component is running</h1>
9             <p>this is para</p>
10            <Programmers/>
11            <Ppoint/>
12            <Ujjain/>
13            <Indore/>
14        </>
15    )
16 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

Here App is parent component and Ppoint,Ujjain,Indore are child component for App component

```
src > JS App.js > [@] App
1 import React from "react";
2 //import Child from "./components/Child";
3 import Programmers from "./components/Child";
4 import { Ppoint } from "./components/Child";
5 const App=()=>{
6     return(
7         <>
8             <h1>App component is running</h1>
9             <p>this is para</p>
10            <Programmers/>
11            <Ppoint/>
12        </>
13    )
14 }
15 export default App;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Session contents restored from 8/26/2022 at 8:18:27 PM

Microsoft Windows [Version 10.0.22000.856]
(c) Microsoft Corporation. All rights reserved.

D:\react11\first>

Here Ppoint is a child component of App component

```

src > components > Child.jsx >...
1 import React from "react";
2 const Child=(...){
3   return <h1>Child Component is running</h1>
4 }
5 const Ppoint=()=>{
6   return(
7     <>
8       <h1>Point is running</h1>
9       <Indore/>
10      <Ujjain/>
11    </>
12  )
13 }
14 const Ujjain=()=>{
15   return <h1>Ujjain is running</h1>
16 }

```

Indore and Ujjain are child component of Ppoint component and grandchild component of App component

Props:

Props is used to define properties or attributes of a component . When a component will receive any Props then react will store all the props in form of JavaScript Object

A component is mainly used to reduce code in application and it is also used to reuse our code many times and in many places in our application.

Use JavaScript in Jsx :-we can't directly use JavaScript in Jsx. If we want to use JavaScript code in Jsx then we will use curly Brackets {}

1st method as variables

```

src > components > One.jsx > Lang
1 import React from "react";
2 const Lang=()=>{
3   var lname="Java Script";
4   var duration="6 months";
5   var details="sldkjsdj dpsjdsks dskjdlsjdl s";
6   var fees=60000;
7   var cls="programmers point";
8   return(
9     <>
10       <h1>Language Name : {lname}</h1>
11       <h2>Duration : {duration}</h2>
12       <h4>Details: {details}</h4>
13       <h3>Fees : {fees}</h3>
14       <h3>Class : {cls}</h3>
15       <hr/>
16     </>
17   )
18 }
19 export default Lang;

```

webpack compiled successfully

2nd method as Object

```

src > components > Onejsx > ...
1 import React from "react";
2 const Lang=()=>{
3   var data={
4     lname:'java script',
5     duration:'6 months',
6     details:'aldjhakjdhada',
7     fees:120000,
8     cls:'programmers point'
9   };
10  return(
11    <>
12      <h1>Language Name : {data.lname}</h1>
13      <h2>Duration : {data.duration}</h2>
14      <h4>Details: {data.details}</h4>
15      <h3>Fees : {data.fees}</h3>
16      <h3>Class : {data.cls}</h3>
17      <hr/>
18    </>
19  )
20}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
webpack compiled successfully

```

Props

```

src > components > Onejsx > ...
1 import React from "react";
2 const Lang=(Props)=>{
3   console.log(Props);
4   return(
5     <>
6       <h1>Language Name : {Props.lname}</h1>
7       <h2>Duration : {Props.duration}</h2>
8       <h4>Details: {Props.details}</h4>
9       <h3>Fees : {Props.fees}</h3>
10      <h3>Class : {Props.cls}</h3>
11      <hr/>
12    </>
13  )
14}
15 export default Lang;

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
D:\react11\first>

```

```

src > JS App.js > ...
1 import React from "react";
2 import Lang from "./components/One";
3 //import Child from "./components/Child";
4 const App=()=>{
5   return(
6     <>
7       <h1>App component is running</h1>
8       <Lang lname="java" duration="1 year" details="java is a best language" fees="240000" c>
9         <Lang lname="python" duration="1 year" details="python is a best language" fees="240000" c>
10        </>
11      </>
12    )
13  export default App;
14

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
webpack compiled successfully

```

Here above is the second method using props, here props variables is declared in curly brackets so that only desired attributes are sent

```

JS App.js      One.jsx  X  # One.css      Child.jsx
src > components > Onejsx > ...
1  import React from "react";
2  import './One.css';
3  const Lang = ({lname,duration,details,fees,cls})=>{
4      return(
5          <>
6          <h1>Language Name : {lname}</h1>
7          <h2>Duration : {duration}</h2>
8          <h4>Details: {details}</h4>
9          <h3>Fees : {fees}</h3>
10         <h3>Class : {cls}</h3>
11         <hr/>
12     </>
13   )
14 }
15 export default Lang;

```

we will import external css in any component by using following import

Index.css file is used to apply styling for whole react application.

App.css:- in App.css file we will write styling code for app component

Inside the app.css file we will write external css content like

```

}
h1{
  color: blue;
}

```

Inline Css:- If we want to use inline css in our react application, we will use style prop and inside a style prop we will pass a JavaScript object.

```

Go Run Terminal Help You are viewing tanveer.programmerspoint's screen View
JS App.js      One.jsx  X  # One.css      Child.jsx  JS index.js
src > components > Onejsx > Lang > color
1  import React from "react";
2  import './One.css';
3  const Lang = ({lname,duration,details,fees,cls})=>{
4      return(
5          <>
6          <h1>Language Name : {lname}</h1>
7          <h2>Duration : {duration}</h2>
8          <h4 style={{color:'green'}}>Details: {details}</h4>
9          <h3>Fees : {fees}</h3>
10         <h3>Class : {cls}</h3>
11         <hr/>
12     </>
13   )
14 }
15 export default Lang;

```

```

<>
<h1>Language Name : {lname}</h1>
<h2>Duration : {duration}</h2>
<h4 style={{color:'green',backgroundColor:'grey'}}>Details: {details}</h4>
<h3>Fees : {fees}</h3>
<h3>Class : {cls}</h3>
<hr/>
</>

```

If we want to apply internal css on a react project or application , then we will define a object at top of component and we can use this object in many component by using style props

The screenshot shows a Visual Studio Code interface with multiple tabs open. The active tab is 'App.js' which contains the following code:

```
src > components > One.jsx > # One.css > Child.jsx > JS index.js > # index.css
1 import React from "react";
2 import './One.css';
3 const Lang = ({lname,duration,details,fees,cls})=>{
4     var styling={color:'green',backgroundColor:'grey'};
5     return(
6         <>
7             <h1>Language Name : {lname}</h1>
8             <h2>Duration : {duration}</h2>
9             <h4 style={styling}>Details: {details}</h4>
10            <h3>Fees : {fees}</h3>
11            <h3>Class : {cls}</h3>
12            <hr/>
13        </>
14    )
15 }
16 export default Lang;
```

Conditional rendering is one of the most important concept of rendering any jsx element according to specified condition

Either we will use conditional or ternary operator for conditional rendering , this conditional or ternary works only on single line codes not with multi line commands in conditional statement

The screenshot shows a Visual Studio Code interface with multiple tabs open. The active tab is 'App.js' which contains the following code:

```
App.js 1 X One.jsx # One.css Child.jsx JS index.js # index.css
c > JS App.js > ...
1 import React from "react";
2 import Lang from "./components/One";
3 //import Child from "./components/Child";
4 import './App.css';
5 const App=()=>{
6     var d=false;
7     return(
8         <>
9             {d}<h1>True block is running</h1>:<h1>False block is running</h1>
10            </>
11        )
12 }
13 export default App;
14
```

```

src > JS App.js > [o]App
1 import React from "react";
2 import Lang from "./components/One";
3 //import Child from "./components/Child";
4 import './App.css';
5 const App=()=>{
6     var d="ppoint";
7     return(
8         <>
9         | {d==="ppoint"?<h1>True block is running</h1>:<h1>False block is running</h1>}
10        </>
11    )
12 }
13 export default App;
14

```

2nd Method:-

```

src > JS App.js > ...
1 import React from "react";
2 import Lang from "./components/One";
3 //import Child from "./components/Child";
4 import './App.css';
5 const App=()=>{
6     var d="ujjain";
7     if(d==="indore")
8     {
9         return(
10            <>
11            | <h1>Programmers Point</h1>
12            | <p>vijaynagar | bhanwarkua</p>
13        </>
14    )
15 }
16 else{
17     return(
18        <>
19        | <h1>Programmers Point</h1>
20        | <p>freeganj ujjain</p>
21        </>
22    )
23 }
24 }
25 export default App;
26

```

```

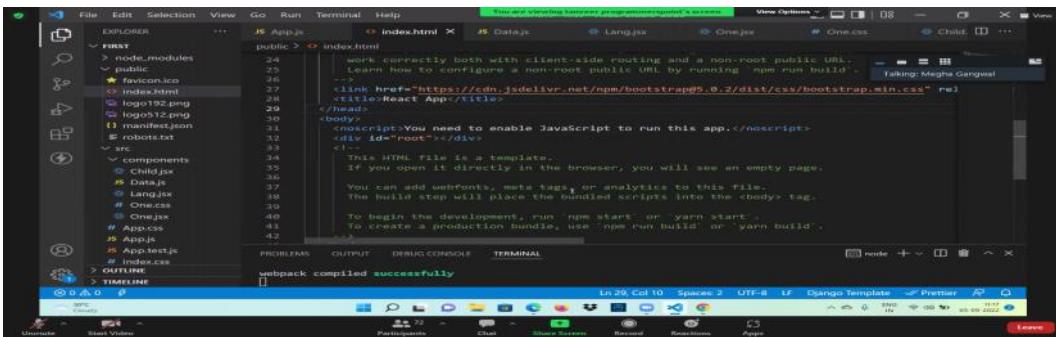
src > JS App.js > ...
13     </>
14 }
15 }
16 else{
17     return(
18        <>
19        | <h1>Programmers Point</h1>
20        | <p>freeganj ujjain</p>
21        </>
22    )
23 }
24 }
25 export default App;
26

```

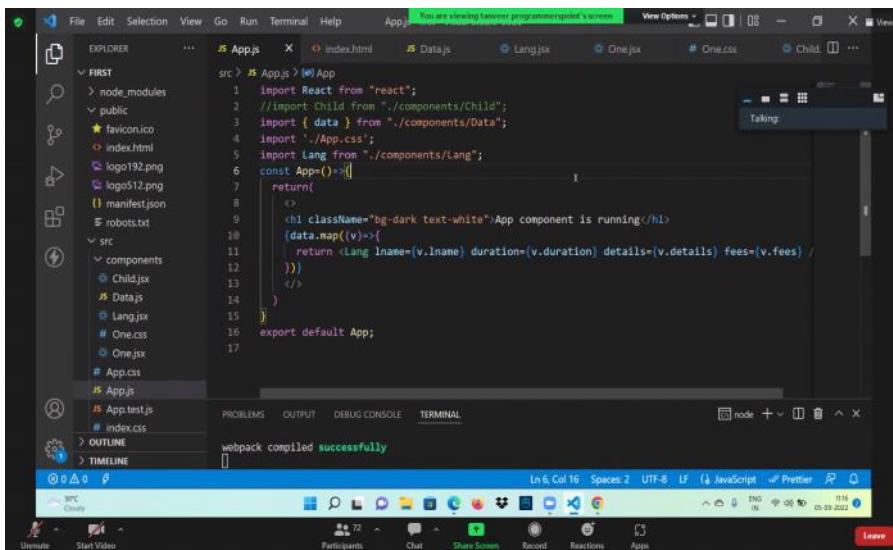
Bootstrap in react project:-

If you want to implement Bootstrap in React.js project then we will use, then we will use following npm command in our project. We can use Bootstrap in our project either by CDN or by npm commands.

1st method to add CDN in index.html file of react



We use classname in react instead of class



We can check all the dependencies of our project in package.json file, and in this file we can add script also.

npm install bootstrap : -command to add dependencies of bootstrap that is to use bootstrap in our project

```
src > JS App.js > ...
1 import React from "react";
2 //import Child from "./components/Child";
3 import { data } from "./components/Data";
4 import './App.css';
5 import Lang from "./components/Lang";
6 import '../node_modules/bootstrap/dist/css/bootstrap.min.css';
7 const App=()=>{
8   return(
9     <>
10     <h1 className="bg-dark text-primary">App component is running</h1>
11     {data.map((v)=>{
12       | return <Lang lname={v.lname} duration={v.duration} details={v.details} fees={v.fees}>
13     })}
14   )
15 }
16
17 export default App;
```

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `index.html`, `package.json`, `Data.js`, `Lang.jsx`, and `One.css`.
- Code Editor:** Displays the `App.js` file containing JSX code to render cards based on data from `Data.js`.
- Terminal:** Shows the command `webpack` was run with 1 warning.
- Bottom Status Bar:** Shows the current date and time (05-09-2022) and system status (Cloudy).

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `index.html`, `package.json`, `Data.js`, `Lang.jsx`, and `One.css`.
- Code Editor:** Displays the `Card.jsx` file containing JSX code for a card component.
- Terminal:** Shows the command `webpack` was run with 1 warning.
- Bottom Status Bar:** Shows the current date and time (05-09-2022) and system status (Cloudy).

Image in keys and values, pass URL as an value of the key

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure with files like `index.html`, `package.json`, `Lang.jsx`, `One.css`, and `One.jsx`.
- Code Editor:** Displays the `Data.js` file containing code to define a `data` array with objects having `lname`, `duration`, `details`, `fees`, and `img` properties.
- Terminal:** Shows the command `webpack` was run with 1 warning.
- Bottom Status Bar:** Shows the current date and time (06-09-2022) and system status (Cloudy).

```

src > components > Card.js > Card
1 import React from "react";
2 const Card = ({lname,duration,details,fees,img})=>{
3   return(
4     <>
5       <div class="card" style={{'width': "18rem"} >
6         <img src=[img] class="card-img-top" alt="..." />
7         <div class="card-body">
8           <h1 className="card-title">{lname} {duration}</h1>
9           <h2 className="card-subtitle">{fees}</h2>
10          <p class="card-text">{details}</p>
11        </div>
12      </div>
13    </>
14  )
15 }
16 }
17 export default Card;

```

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL

Line 5:8: 'Lang' is defined but never used [no-unused-vars](#)

webpack compiled with 1 warning

In 6 Col 16 Spaces: 2 UTF-8 CRLF (JavaScript React) ↗ Prettier Leave

If we want to put images in our react project ,then we will construct a images folder inside public folder and we will store all the images inside images folder
Path is as follows

`img: '/images/java.jpg'`,

```

src > components > Data.js > data > img
1
2   'year',details:'java is a top rated language',fees:240000,img:'/images/java.jpg'),
3   '1 year',details:'python is a top rated language',fees:240000,img:'/images/python.jpg'),
4   '3 months',details:'mern is a top rated language',fees:240000,img:'/images/java.jpg'),
5   '3 months',details:'react is a top rated language',fees:240000,img:'/images/python.jpg'),
6   '6 months',details:'angular is a top rated language',fees:240000,img:'/images/java.jpg')
7
8

```

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL

Line 5:8: 'Lang' is defined but never used [no-unused-vars](#)

webpack compiled with 1 warning

In 6 Col 101 Spaces: 4 UTF-8 CRLF (JavaScript React) ↗ Prettier Leave

```

src > components > Card.js > Card
1 import React from "react";
2 const Card = ({lname,duration,details,fees,img})=>{
3   return(
4     <>
5       <div class="card" style={{'width': "18rem"} >
6         <img src=[img] class="card-img-top" alt="..." />
7         <div class="card-body">
8           <h1 className="card-title">{lname} {duration}</h1>
9           <h2 className="card-subtitle">{fees}</h2>
10          <p class="card-text">{details}</p>
11        </div>
12      </div>
13    </>
14  )
15 }
16 }
17 export default Card;

```

PROBLEMS ① OUTPUT DEBUG CONSOLE TERMINAL

Line 5:8: 'Lang' is defined but never used [no-unused-vars](#)

webpack compiled with 1 warning

In 6 Col 16 Spaces: 2 UTF-8 CRLF (JavaScript React) ↗ Prettier Leave

Hooks:-

React hooks are mostly used with functional component , it reduces large code into small codes. hooks are mainly used to convert component methods into short codes and easy use.

All react hooks should be declared at top of components . Some commonly used react oops are useState ,useDispatch,useSelector,useEffect,createContext,useMemo etc

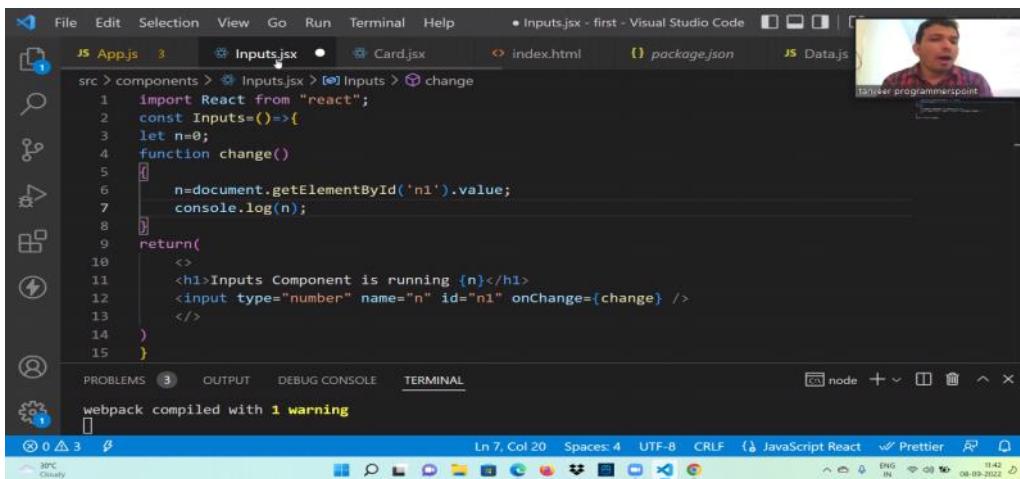
useState hook is mainly used to maintain state of a variable of a component or variable life cycle, with the help of the useState we can use a control component in react js

When we will construct a variable using use state then it will return array and by using destructuring we will store this array into 2 variables 1st one is state variable and second one is updated function.

In state variable we will store initial state by using useState and we can change and modify state by using updated function

Uncontrolled component:-That is if we change value of any component with the help of JavaScript then it will not reflect change in our jsx element of our component . If you want to modify along with reflection in jsx element we use useState

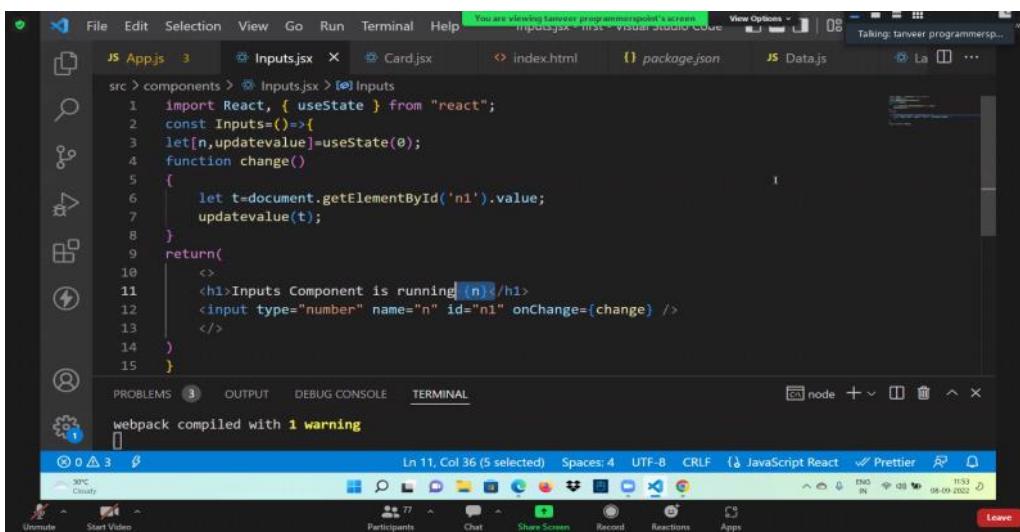
We can modify a state variable according to javaScript events ,but it will not reflect its value in jsx, because javaScript will modify value of variable it will not change the state of jsx



A screenshot of Visual Studio Code showing the code for a component named 'Inputs'. The code uses a standard JavaScript function-based approach to handle state and events:

```
src > components > Inputs.jsx > Inputs > change
1 import React from "react";
2 const Inputs=()=>{
3 let n=0;
4 function change()
5 {
6   n=document.getElementById('n1').value;
7   console.log(n);
8 }
9 return(
10   <>
11   <h1>Inputs Component is running {n}</h1>
12   <input type="number" name="n" id="n1" onChange={change} />
13   </>
14 )
15 }
```

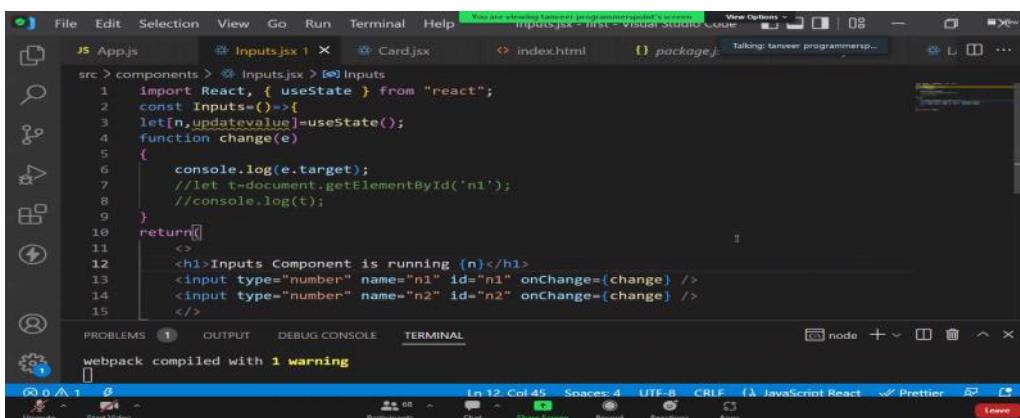
The terminal shows a warning from webpack: "webpack compiled with 1 warning".



A screenshot of Visual Studio Code showing the same component 'Inputs' but using the useState hook. The code is more concise and follows functional programming principles:

```
src > components > Inputs.jsx > Inputs
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n,updatevalue]=useState(0);
4 function change()
5 {
6   let t=document.getElementById('n1').value;
7   updatevalue(t);
8 }
9 return(
10   <>
11   <h1>Inputs Component is running {n}</h1>
12   <input type="number" name="n" id="n1" onChange={change} />
13   </>
14 )
15 }
```

The terminal shows a warning from webpack: "webpack compiled with 1 warning".



A screenshot of Visual Studio Code showing the component 'Inputs' using useState with event handling. It logs the target of the event to the console:

```
src > components > Inputs.jsx > Inputs
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n,updatevalue]=useState();
4 function change(e)
5 {
6   console.log(e.target);
7   //let t=document.getElementById('n1');
8   //console.log(t);
9 }
10 return(
11   <>
12   <h1>Inputs Component is running {n}</h1>
13   <input type="number" name="n1" id="n1" onChange={change} />
14   <input type="number" name="n2" id="n2" onChange={change} />
15   </>
)
```

The terminal shows a warning from webpack: "webpack compiled with 1 warning".

A screenshot of the Visual Studio Code interface. The code editor shows a file named App.js with the following content:

```
src > components > Inputs.jsx > Inputs > change
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n,updatevalue]=useState();
4 function change(e)
5 {
6   updatevalue(e.target.value);
7   //console.log(e.target.id);
8   //console.log(e.target.name);
9   //console.log(e.target.value);
10  //let t=document.getElementById('n1');
11  //console.log(t);
12 }
13 return(
14   <>
15   <h1>Inputs Component is running {n}</h1>
16   <input type="number" name="n1" id="n1" onChange={change} />
17 )
18 }
19 
```

The terminal below shows the command "webpack compiled with 1 warning".

A screenshot of a web browser window titled "React App" at "localhost:3000". The page displays the text "Inputs Component is running" above a text input field containing the value "5". The browser's developer tools are open, showing the "Console" tab with the following log entries:

```
[webpack-dev-server] Server index.js:51
started: Hot Module Replacement enabled, Live
Reloading enabled, Progress disabled, Overlay
enabled.
n1           Inputs.jsx:6
n1           Inputs.jsx:7
5            Inputs.jsx:8
```



A screenshot of the Visual Studio Code interface. The code editor shows the same App.js file as before, but the terminal now displays the message "webpack compiled successfully".

Screenshot of Visual Studio Code showing code in App.js:

```
src > components > Inputs.jsx > [e]Inputs > ⚡ change
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n1,updatevalue1]=useState(0);
4 let[n2,updatevalue2]=useState(0);
5 function change(e)
6 {
7 if(e.target.name==="n1")
8 [
9 updatevalue1(e.target.value);
10 ]
11 else
12 [
13 updatevalue2(e.target.value);
14 ]
15 }
```

Terminal output: webpack compiled successfully

Screenshot of Visual Studio Code showing code in App.js:

```
src > components > Inputs.jsx > [e]Inputs > ⚡ change
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n1,updatevalue1]=[useState([0,0]);
4 function change(e)
5 {
6 if(e.target.name==="n1")
7 [
8 n1[0]=e.target.value;
9 updatevalue1(n1);
10 ]
11 else
12 [
13 n1[1]=e.target.value;
14 updatevalue1(n1);
15 }
```

Terminal output: Unmute Start Video Participants Chat Share Screen Record Reactions Apps Leave

Screenshot of Visual Studio Code showing code in App.js:

```
src > components > Inputs.jsx > [e]Inputs > ⚡ change > ⚡ [e.target.name]
1 import React, { useState } from "react";
2 const Inputs=()=>{
3 let[n,updatevalue1]=useState({n1:0,n2:0});
4 function change(e)
5 [
6 updatevalue1({...n,[e.target.name]:e.target.value});
7 ]
8 return(
9 <>
10 <h1>Inputs Component is running {n.n1+n.n2}</h1>
11 <input type="number" name="n1" id="n1" onChange={change} />
12 <input type="number" name="n2" id="n2" onChange={change} />
13 </>
14 )
15 }
```

Terminal output: webpack compiled successfully

S:\acer\OneDrive\Desktop\pp.html - Sublime Text (UNREGISTERED)

```

<script>
  var d={n1:0,n2:0};
  d={...d,n1:5};
  console.log(d);
</script>

```

The Simplified explanation of the state handling of n number
Of inputs can be explained by following destructuring of object

Handling various inputs in calci

```

JS App.js Inputs.jsx Card.jsx index.html package.json JS Data.js La ...
src > components > Inputs.jsx > Inputs
1 import React, { useState } from "react";
2 const Inputs=()=>{
3   let[n,updatevalue]=useState({n1:0,n2:0,r:0});
4   function change(e)
5   {
6     updatevalue({...n,[e.target.name]:e.target.value});
7   }
8   function add(e)
9   {
10    updatevalue({...n,r:(parseInt(n.n1)+parseInt(n.n2))});
11  }
12  return(
13    <>
14    <h1>Inputs Component is running {n.r}</h1>
15    <input type="number" name="n1" id="n1" onChange={change} />

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

webpack compiled successfully

1 Answers

This is to dynamically update object property (when the name of the property is unknown upfront but runtime). This way you could have multiple React inputs having a different name property and using the same onChange handler to update part of the state.

**Square brackets in [e.target.name] is used to set priority of the key first that is first the value of the key should be updated than its value

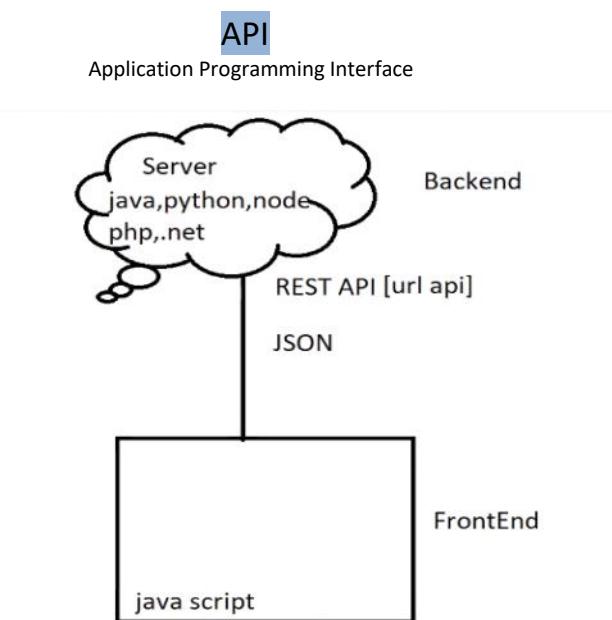
```

App.js 4 AddProduct.jsx Inputs.jsx Card.jsx index.html package.json
components > AddProduct.jsx > AddProduct > submit
}
const submit=(e)=>{
  e.preventDefault();
  console.log(data);
}
return(
  <>
  <form onSubmit={submit}>
    Name<input type="text" name="name" value={data.name} onChange={change}>/>
    Price<input type="number" name="price" value={data.price} onChange={change}>/>
    Category<input type="text" name="cat" value={data.cat} onChange={change}>/>
    Company<input type="text" name="cmp" value={data.cmp} onChange={change}>/>
    <button>Add</button>
  </form>
</>

```

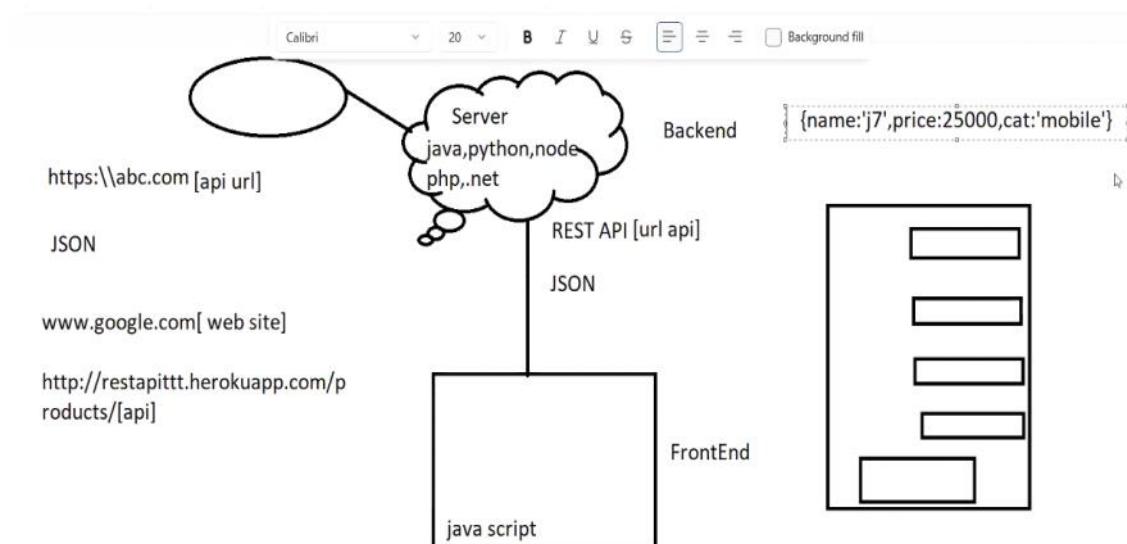
Whenever we submit our form ,the onSubmit event will send data in console in this case, but it get erased from console after a short duration this is because on submitting a form the page will be reload or refreshed , to prevent this we call preventDefault() method on event object e preventDefault() is a method which we will use at the time of submitting a form, it will not reload or refresh page

data



JavaScript connectivity with database is not possible directly but indirectly can be done with the help of JSON and REST API

A REST API (also known as RESTful API) is an application programming interface (API or web API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services. REST stands for **representational state transfer** and was created by computer scientist Roy Fielding



Thunder Client or postman are client tool, which is mainly used to test a rest API
Common Methods of REST API

show	GET
search	GET
add	POST
delete	DELETE
update	PUT

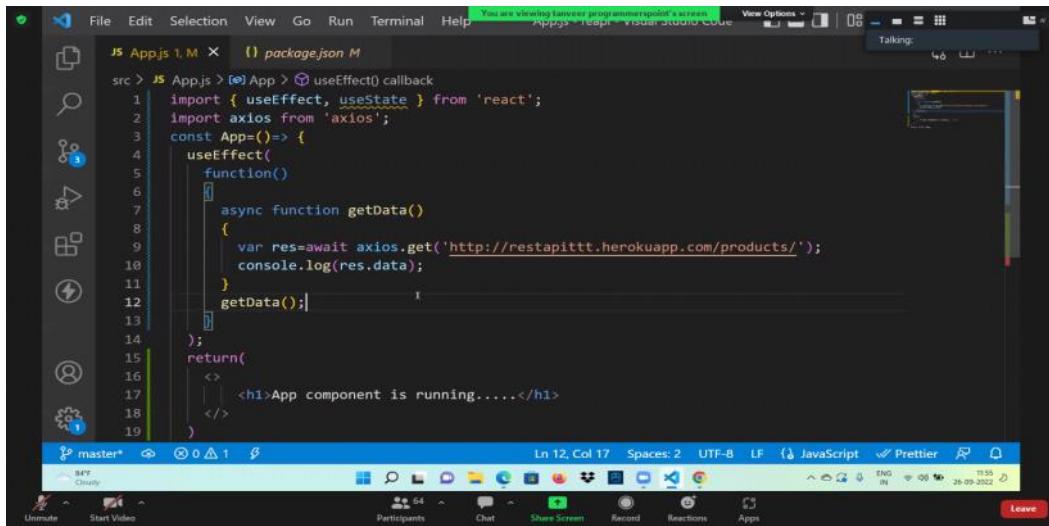
<http://restapittt.herokuapp.com/products/>
<https://regres.in/>

Axios:- axios is a package through which we will/can call rest api in react, with the help of axios we can call get,post,put and delete method, when we will call axios methods then it will always return response in form of

promises, so either we will manually resolve promises or we call axios methods in async await function

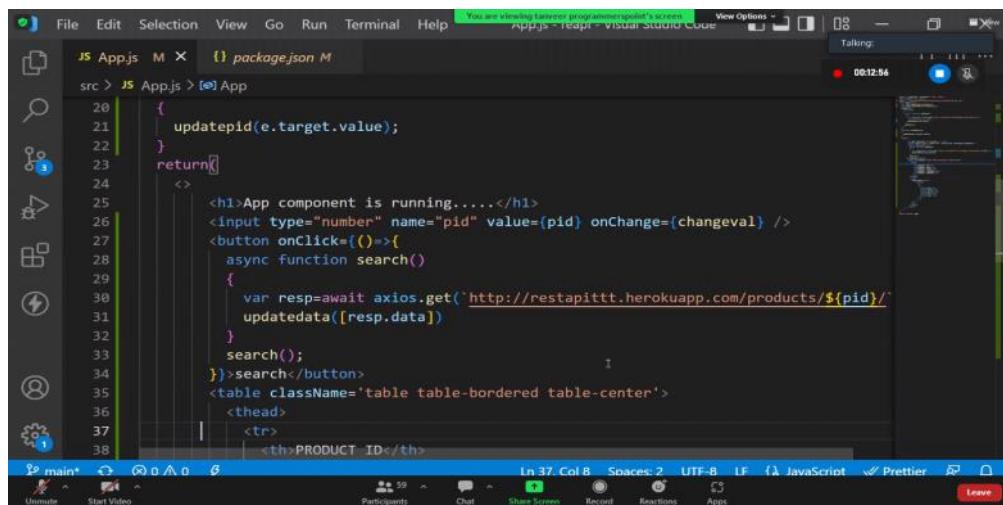
We can install axios with the help of following npm command

```
npm install axios
```

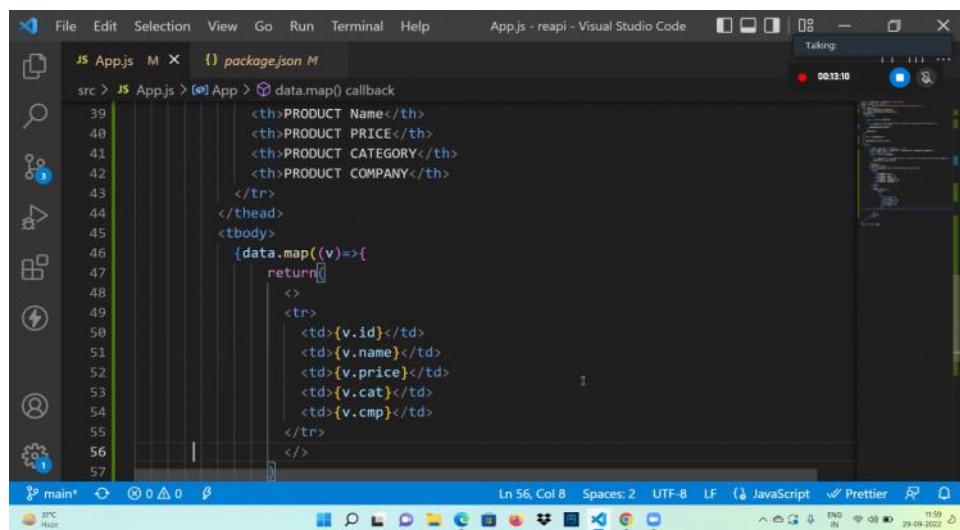


```
JS App.js 1.M ✘ {} package.json M
src > JS App.js > (e) App > useEffect() callback
1 import { useEffect, useState } from 'react';
2 import axios from 'axios';
3 const App=()=> {
4   useEffect(
5     function()
6     {
7       async function getData()
8       {
9         var res=await axios.get('http://restapitt.herokuapp.com/products/');
10        console.log(res.data);
11      }
12      getData();
13    }
14  );
15  return(
16    <>
17      <h1>App component is running.....</h1>
18    </>
19  )
}
```

Get method with a specific id:-



```
JS App.js M ✘ {} package.json M
src > JS App.js > (e) App
20   {
21     updatepid(e.target.value);
22   }
23   return(
24     <>
25       <h1>App component is running.....</h1>
26       <input type="number" name="pid" value={pid} onChange={changeval} />
27       <button onClick={()=>{
28         async function search()
29         {
30           var resp=await axios.get(`http://restapitt.herokuapp.com/products/${pid}`);
31           updatedata([resp.data]);
32         }
33         search();
34       }}>search</button>
35       <table className='table table-bordered table-center'>
36         <thead>
37           <tr>
38             <th>PRODUCT_ID</th>
```



```
JS App.js M ✘ {} package.json M
src > JS App.js > (e) App > data.map() callback
39   <th>PRODUCT_NAME</th>
40   <th>PRODUCT PRICE</th>
41   <th>PRODUCT CATEGORY</th>
42   <th>PRODUCT COMPANY</th>
43   </tr>
44   </thead>
45   <tbody>
46     {data.map((v)=>{
47       return(
48         <>
49           <tr>
50             <td>{v.id}</td>
51             <td>{v.name}</td>
52             <td>{v.price}</td>
53             <td>{v.cat}</td>
54             <td>{v.cmp}</td>
55           </tr>
56         </>
57       )
58     })
59   }
60 }
```

```
48 |             <>
49 |             <tr>
50 |                 <td>{v.id}</td>
51 |                 <td>{v.name}</td>
52 |                 <td>{v.price}</td>
53 |                 <td>{v.cat}</td>
54 |                 <td>{v.cmp}</td>
55 |             </tr>
56 |         )
57 |     )
58 | )
59 | </tbody>
60 | </table>
61 |
62 | }
63 |
64 | export default App;
65 |
```

Ln 65, Col 1 Spaces: 2 UTF-8 LF JavaScript Prettier

```
src > JS App.js > ...
1 import { useEffect, useState } from 'react';
2 import axios from 'axios';
3 import './node_modules/bootstrap/dist/css/bootstrap.min.css';
4 const App=()=>{
5     var [pid,updatepid]=useState();
6     var [data,updatedata]=useState([]);
7     var [pro,updatepro]=useState({name:'',price:0,cat:'',cmp:''});
8     useEffect(
9         function()
10     {
11         async function getData()
12         {
13             var res=await axios.get('http://restapittt.herokuapp.com/products/');
14             //console.log(res.data);
15             updatedata(res.data);
16         }
17         getData();
18     }
19 );
```

Ln 1, Col 1 Spaces: 2 UTF-8 LF JavaScript Prettier

```
src > JS App.js > @App > @<function> > @addpro
19 );
20 function changeval(e)
21 {
22     updatepid(e.target.value);
23 }
24 const prochange=(e)=>{
25     updatepro({...pro,[e.target.name]:e.target.value});
26 }
27 return(
28     <>
29         <h1>App component is running.....</h1>
30         <form onSubmit={(e)=>{
31             e.preventDefault();
32             async function addpro()
33             {
34                 var res=await axios.post("http://restapittt.herokuapp.com/products/",pro)
35                 if(res.status==201)
36                 {
37                     alert('product added sucessfully!');
```

Ln 36, Col 5 Spaces: 2 UTF-8 LF JavaScript Prettier

A screenshot of a video conferencing interface. At the top, a toolbar includes File, Edit, Selection, View, Go, Run, Terminal, Help, Talking, and a timestamp of 00:40:43. Below the toolbar is a code editor window titled "JS App.js M X". The code is as follows:

```
src > JS App.js > [e] App > ↗ <function> > ↗ addpro
  19
  20
  21
  22
  23
  24
  25
  26
  27
  28
  29
  30
  31
  32
  33
  34
  35
  36
  37
```

The code continues with several function definitions and component logic. The status bar at the bottom shows Ln 36, Col 1, Spaces: 2, UTF-8, LF, JavaScript, Prettier, and a timestamp of 00:40:43.

A screenshot of a video conferencing interface. At the top, a toolbar includes File, Edit, Selection, View, Go, Run, Terminal, Help, Talking, and a timestamp of 00:40:59. Below the toolbar is a code editor window titled "JS App.js M X". The code is as follows:

```
src > JS App.js > [e] App > ↗ <function> > ↗ search
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
  56
```

The code continues with component logic, including input fields and a button. The status bar at the bottom shows Ln 55, Col 2, Spaces: 2, UTF-8, LF, JavaScript, Prettier, and a timestamp of 00:40:59.

A screenshot of a video conferencing interface. At the top, a toolbar includes File, Edit, Selection, View, Go, Run, Terminal, Help, Talking, and a timestamp of 00:41:15. Below the toolbar is a code editor window titled "JS App.js M X". The code is as follows:

```
src > JS App.js > [e] App > ↗ <function> > ↗ search
  37
  38
  39
  40
  41
  42
  43
  44
  45
  46
  47
  48
  49
  50
  51
  52
  53
  54
  55
```

The code continues with component logic, including input fields and a button. The status bar at the bottom shows Ln 54, Col 2, Spaces: 2, UTF-8, LF, JavaScript, Prettier, and a timestamp of 00:41:15.

Screenshot of Visual Studio Code showing the code editor with package.json open. The file path is src > JS App.js > [e] App > data.map() callback. The code is a part of a function mapping over an array of objects (v) to generate rows for a table. The table has columns for PRODUCT ID, Name, Price, Category, and Company.

```
src > JS App.js > [e] App > data.map() callback
  ...
  <tbody>
    {data.map((v)=>{
      return[

        <tr key={v.id}>
          <td>{v.id}</td>
```

Screenshot of Visual Studio Code showing the continuation of the code editor with package.json open. The file path is src > JS App.js > [e] App > data.map() callback > <function>. The code adds more columns to the table row: name, price, category, and company. It then defines a delete button's onClick event which uses axios to delete a product from a specified URL. If successful (status 204), it shows an alert message.

```
src > JS App.js > [e] App > data.map() callback > <function>
  ...
  <td>{v.id}</td>
  <td>{v.name}</td>
  <td>{v.price}</td>
  <td>{v.cat}</td>
  <td>{v.cmp}</td>
  <td><button className='btn btn-danger' onClick={()=>{
    async function delpro()
    {
      var res=await axios.delete('http://restapittt.herokuapp.com/pro
      if(res.status==204)
      {
        alert("product deleted sucessfully");
      }
    }
    delpro();
  }}>Delete</button></td>
  <td><button className='btn btn-warning' onClick={()=>[
    prochange({name:v.name,price:v.price,cat:v.cat,cmp:v.cmp});
```

Screenshot of Visual Studio Code showing the final state of the code editor with package.json open. The file path is src > JS App.js > The code continues the table row definition, adding update and delete buttons. It then closes the tbody and table tags. Finally, it exports the default App component.

```
src > JS App.js > ...
  ...
  <td><button className='btn btn-warning' onClick={()=>[
    prochange({name:v.name,price:v.price,cat:v.cat,cmp:v.cmp});
```

Async await:-

As we know JavaScript is synchronous it means it will skip those code ,that will take some extra time in execution so JavaScript introduced a concept of promises, through which we can execute JavaScript code in asynchronous way. But we have to write some lengthy codes to resolve a promise so ECMAScript 6 introduced a

```
<h1>Rest Api Using Java Script</h1>
<script>
  async function show() {
    var res=await fetch("http://restapittt.herokuapp.com/products/");
    var data=await res.json();
    for(var i of data)
    {
      var d=document.createElement("div");
      d.textContent=i.id+" "+i.name+" "+i.price+" "+i.cat+" "+i.cmp;
      var q=document.querySelector("body");
      q.appendChild(d);
    }
  }
  show();
</script>
```

Concept of async and await keyword in which we will define a function with the help of async keyword and we can only use await keyword inside a async function. await keyword mostly used with promises, and await automatically resolve a promise and return data. We will call a rest API then it will always return response in from of promise , So usually we will handle this by using async and await

```
<script>
  var a=10;
  console.log(a);
  var p=new Promise(function(resolve,reject)
  {
    setTimeout(function(){
      resolve(a=20);
    },3000);
  });
  async function show()
  {
    var v=await p;
    console.log(v);
  }
  show();
  a=30;
  console.log(a);
</script>
```

Fetch Method:- Fetch method is mainly used to call REST API in JavaScript, with the help of fetch method we can call get, post, put and delete methods . fetch method always return a promise so we can extract data by resolving a promise or we can call fetch method inside async function and await keyword.

```
<script>
  async function show()
  {
    var d=await fetch("http://restapittt.herokuapp.com/products/");
    d=await d.json();
    console.log(d);
  }
  show();
</script>
```

C:\Users\user\OneDrive\Desktop\pp.html - Sublime Text (UNREGISTERED)

You are viewing Tanveer programmerspoint's screen

```

1 <h1>Rest Api Using Java Script</h1>
2 <input type="text" name="" id="pid" placeholder="please enter id">
3 <button onclick="search()">serach</button>
4 <button onclick="deletes()">Delete</button>
5 <button onclick="show()">show all data</button>
6 <script>
7   async function deletes() {
8     let options = {
9       method: 'DELETE',
10      headers: {
11        'Content-Type': 'application/json; charset=utf-8'
12      }
13    }
14    var pid=document.getElementById('pid').value;
15    var res=await fetch(`http://restapittt.herokuapp.com/products/${pid}`);
16    if(res.status==204)
17    {
18      alert("product sucessfully deleted")
19    }
20  }
21
22  async function show() {
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42

```

Participants: 66 Chat Share Screen Record Reactions Apps Leave

C:\Users\user\OneDrive\Desktop\pp.html - Sublime Text (UNREGISTERED)

You are viewing Tanveer programmerspoint's screen

```

21   async function show() {
22     var res=await fetch("http://restapittt.herokuapp.com/products/");
23     var data=await res.json();
24     for(var i of data)
25     {
26       var d=document.createElement("div");
27       d.textContent=i.id+ " "+i.name+ " "+i.price+ " "+i.cat+ " "+i.cmp;
28       var q=document.querySelector("body");
29       q.appendChild(d);
30     }
31   }
32   async function search() {
33     var pid=document.getElementById('pid').value;
34     var res=await fetch(`http://restapittt.herokuapp.com/products/${pid}`);
35     var i=await res.json();
36     var d=document.createElement("div");
37     d.textContent=i.id+ " "+i.name+ " "+i.price+ " "+i.cat+ " "+i.cmp;
38     var q=document.querySelector("body");
39     q.appendChild(d);
40   }
41 </script>
42

```

Participants: 66 Chat Share Screen Record Reactions Apps Leave

task

<http://restp.herokuapp.com/language/>
<http://restp.herokuapp.com/student/>

useEffect() hook

useEffect hook is one of the most important hook of react, It will call automatically before rendering it means when a component contains n number of state variables and when we will update value of a state variable before this useEffect will be called automatically .

useEffect contains either 1 or 2 arguments, 1st argument should always be a call back function which is automatically called every time before rendering. And 2nd argument is an array in which we will pass state variables on which we have to call useEffect

```

import { useEffect, useState } from 'react';
const App=()=>{
  let [n1,updaten1]=useState(0);
  let [n2,updaten2]=useState(0);
  useEffect(()=>
  {
    alert("useeffect running.....");
  },[n1]);
}

```

A screenshot of a video conference interface. At the top, a status bar says "You are viewing tanveer programmer's screen". Below it is a file tree labeled "EXPLORER" showing a "src" folder containing "App.css", "App.js", "App.test.js", "index.css", "index.js", "logo.svg", "reportWebVitals.js", and "setupTests.js". The "App.js" file is open in the main editor area, showing the following code:

```

import { useEffect, useState } from 'react';
const App=()=>{
  useEffect(()=>
  {
    alert("useeffect running.....");
  },[n1]);
  const show=(e)=>{
    if(e.target.name=="n1")
    {
      updaten1(n1++);
    }
    else
    {
      updaten2(n2++);
    }
  }
  return(
    <>
    <h1>{n1}</h1>
    <h1>{n2}</h1>
    <button name='n1' onClick={show}>click me first</button>
    <button name='n2' onClick={show}>click me second</button>
    </>
  )
}
export default App;

```

The code editor has syntax highlighting and a status bar at the bottom indicating "Ln 10, Col 2" and "Spaces: 2".

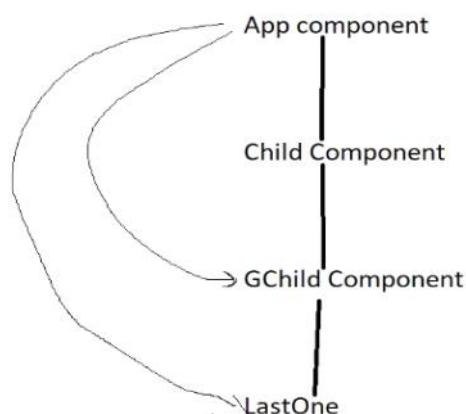
```

    updaten1(n1++);
  }
  else
  {
    updaten2(n2++);
  }
}
return(
  <>
  <h1>{n1}</h1>
  <h1>{n2}</h1>
  <button name='n1' onClick={show}>click me first</button>
  <button name='n2' onClick={show}>click me second</button>
  </>
)
}
export default App;

```

Context API/useContext Hook

It is a concept through which, We will transfer data from parent component to bottom hierarchy of child component . Any child component can fetch data of parent component .we can use create context in parent component and it's an hook so we will on top of component hierarchy, so with the help of useContext we will use data of parent component in Child component



createContext hook is used to create a context and with the help of provider we will provide its data to bottom hierarchy of child component

```
import logo from './logo.svg';
import './App.css';
import First from './components/first';
import { createContext } from 'react';
let Data;
const App=()=>{
  Data=createContext();
  return(
    <>
    <h1>App Component is Running.....</h1>
    <Data.Provider value={'programmers point'}>
      <First/>
    </Data.Provider>
    </>
  )
}
export {Data};
export default App;
```

```
import Second from "./Second";

const First=()=>{
  return(
    <>
    <h1>First Component is Running....</h1>
    <Second/>
    </>
  )
}
export default First;
import Fourth from "./Fourth";

const Third=()=>{
  return(
    <>
    <h1>Third Component is Running....</h1>
    <Fourth/>
    </>
  )
}
export default Third;
```

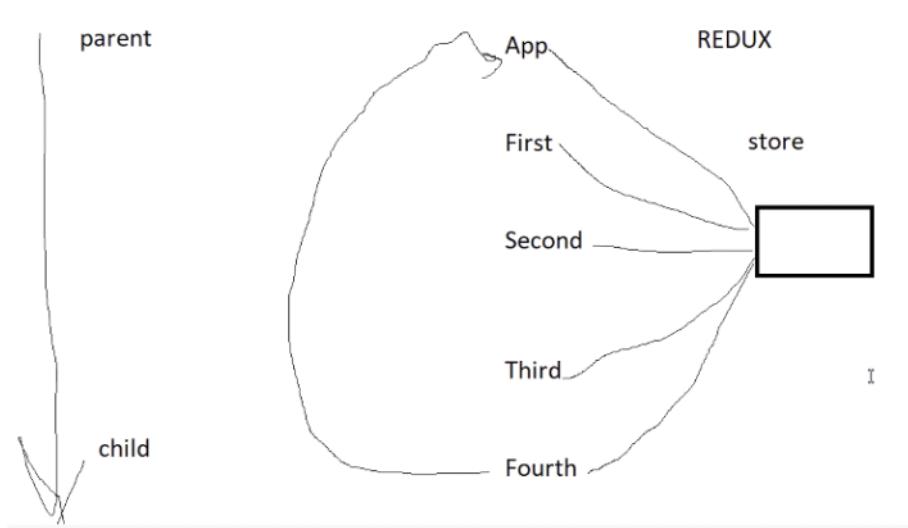
```
import { useContext } from "react";
import { Data } from "../App";
const Fourth=()=>{
  let t=useContext(Data);
  return(
    <>
      <h1>Fourth Component is Running....{t}</h1>
    </>
  )
}
export default Fourth;
```

Sending Object in createContext

```
import First from './components/first';
import { createContext } from 'react';
let Data;
let Datas;
const App=()=>{
  Data=createContext();
  Datas=createContext();
  return(
    <>
      <h1>App Component is Running.....</h1>
      <Data.Provider value={'programmers point'}>
        <Datas.Provider value={{name:'ajay',address:'indore'}}>
          <First/>
        </Datas.Provider>
      </Data.Provider>
    </>
  )
}
export {Data,Datas};
```

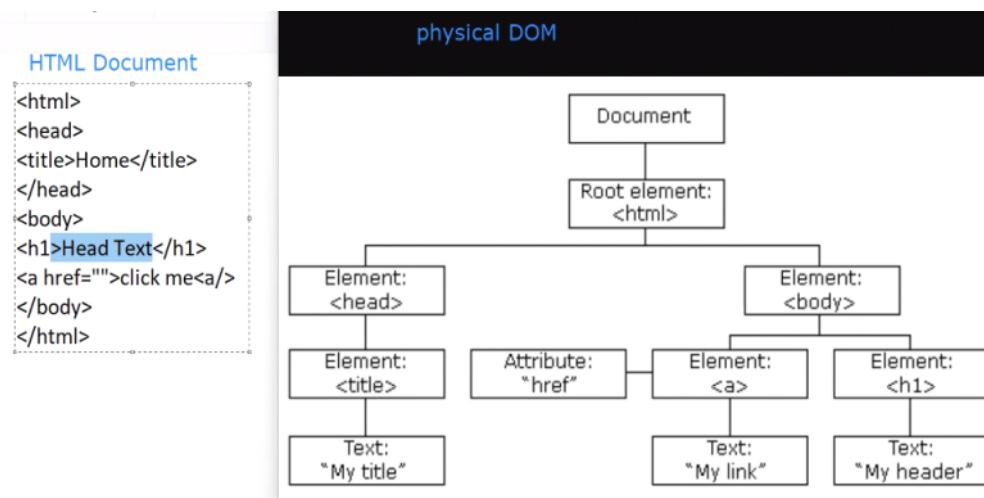
```
import { useContext } from "react";
import { Data,Datas } from "../App";
const Fourth=()=>{
  let t=useContext(Data);
  let tt=useContext(Datas);
  return(
    <>
      <h1>Fourth Component is Running....{t}</h1>
      <h4>Name : {tt.name} Address : {tt.address}</h4>
    </>
  )
}
export default Fourth;
```

Redux



Physical DOM vs Virtual Dom

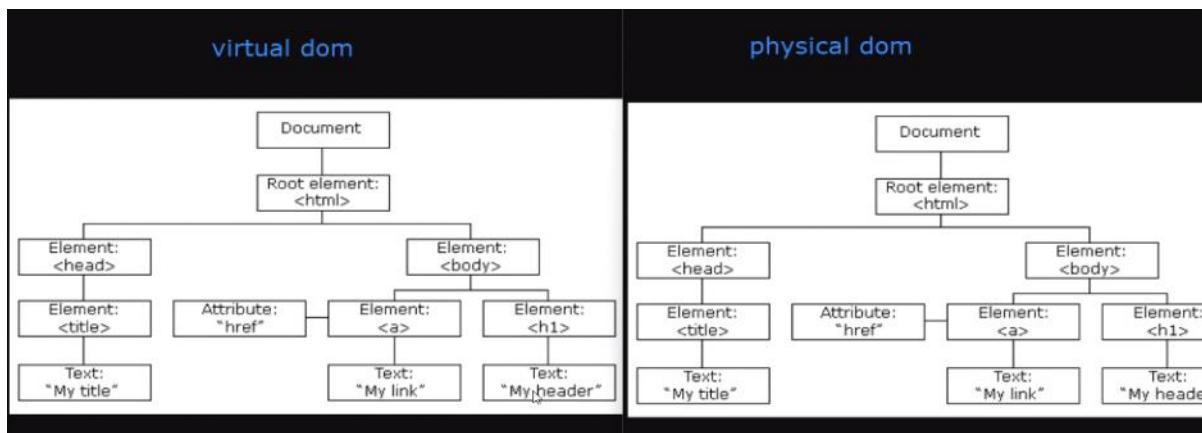
As we know DOCUMENT OBJECT MODEL is used to render a html document into browser, when we will use only javaScript then it will use physical DOM , the major drawback of javaScript is that when we will update a component or some piece of code in our html document then it will reconstruct whole DOM and render it into browser.



Virtual DOM

It is a DOM that doesn't exist physically

And virtual DOM will compare its data with physical DOM, and if any changes in virtual DOM doesn't present in physical DOM then virtual DOM will update only those changes in physical DOM



Routing

Routing is a process through which we can achieve or construct single page application in react js. We will install react router DOM with the help of following command

```
npm install react-router-dom
```

We will use routes and route component of react router dom through we which we will define roots for our application and with the help of link we will define all the paths for roots

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import './index.css';
import App from './App';
import reportWebVitals from './reportWebVitals';
import { BrowserRouter } from 'react-router-dom';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
reportWebVitals();
```

```
import logo from './logo.svg';
import './App.css';
import { Route, Routes } from 'react-router-dom';
import Home from './components/Home';
import Service from './components/Service';
import Contact from './components/Contact';
import PageNotFound from './components/PageNotFound';
const App=()=>{
  return(
    <>
    <Routes>
      <Route path='/' element={<Home/>}></Route>
      <Route path='/service/:val' element={<Service/>}></Route>
      <Route path='/contact' element={<Contact/>}></Route>
      <Route path='*' element={<PageNotFound/>}></Route>
    </Routes>
    </>
  )
}
```

```

import React from "react";
import { useParams } from "react-router-dom";
const Service=()=>{
    let v=useParams();
    return <h1>{v.val} Service Component is running</h1>
}
export default Service;

```

UseParam hook is mainly use to use or read parameters from a URL. By using routing when we will pass URL with the help of /and parameter name then we can read this parameter in any component with the help of useParam, it will always return a object in which parameter name should be considered as key and the parameter name should be pass at the time of URL calling

```

import React from "react";
import { useParams } from "react-router-dom";
const Service=()=>{
    let v=useParams();
    return <h1>{v.val} {v.name} Service Component is runn
}
export default Service;
<>
<Routes>
    <Route path='/' element={<Home/>}></Route>
    <Route path='/service/:val/:name' element={<Service/>}></Route>
    <Route path='/contact' element={<Contact/>}></Route>
    <Route path='*' element={<PageNotFound/>}></Route>

```

If you want to construct a single page application ,then we will use nav tag and inside the nav we will use link tags

```

</Routes>
<a href="/">Home</a>
<a href="/contact">Contact</a>
<a href="/service/programmers/point">Service</a>
<nav>
    <Link to="/">Home</Link>
    <Link to="/contact">Contact</Link>
    <Link to="/service/programmers/point">Service</Link>
    <Link to="/about">About</Link>
</nav>
</>

```

useNavigate hook mainly used to navigate to a specific URL ,it is mainly used to redirect current component to any other component by using its URL

```

const App=()=>{
  return(
    <>
    <Routes>
      <Route path='/' element={<Home/>}></Route>
      <Route path='/service/:val/:name' element={<Service/>}></Route>
      <Route path='/contact' element={<Contact/>}></Route>
      <Route path='/login/:uid/:pwd' element={<Login/>}></Route>
      <Route path='/userhome' element={<UserHome/>}></Route>
      <Route path='*' element={<PageNotFound/>}></Route>
    </Routes>
    <nav>
      <Link to="/">Home</Link>
      <Link to="/contact">Contact</Link>
      <Link to="/service/programmers/point">Service</Link>
      <Link to="/about">About</Link>
    </nav>
  </>|
)

```

```

src > components > Service.jsx > ...
1 import React from "react";
2 import { useParams } from "react-router-dom";
3 const Service=()=>{
4   let v=useParams();
5   return <h1>{v.val} {v.name} Service Component is running</h1>
6 }
7 export default Service;

```

```

Login.jsx 1, U X UserHome.jsx U Java.jsx U Python.jsx U
src > components > Login.jsx > ...
1 import React from "react";
2 import { useNavigate, useParams } from "react-router-dom";
3 const Login=()=>{
4   let obj=useParams();
5   let navigate=useNavigate();
6   if(obj.uid==='admin'&&obj.pwd==='12345')
7   {
8     navigate("/userhome");
9   }
10 else
11 {
12   alert("Invalid id and password")
13 }
14
15 }
16 export default Login;

```

useLocation hook
useLocation hook is mainly used to show current URL location as a String

```

import React from "react";
import { useLocation } from "react-router-dom";
const UserHome=()=>{
    let loc=useLocation();
    //console.log(loc.pathname);
    return <h1>{loc.pathname} UserHome Component is running</h1>
}
export default UserHome;

```

useHistory hook

It was mainly used in react-router-dom version 5 to find out previous URL and by using go back we can render to previous URL

Redux

Redux is one of the most important concept of react js through which we can achieve bidirectional data flow between react components , with the help of react we will store data in a centralized store, any component can store data inside this centralized store and we can also retrieve centralized data in any other component. We can implement Redux with the help of following 2 ways

- a) By using createStore (old technique)
- b) By using configureStore(new technique)

Redux App using createStore technique:-if we want to implement redux with the help of createStore. First of all we will construct an action

Action-action is a function through which we will return an object that contains a key whose name is type and a value in which we will pass a String as action which is used to define an operation which we want to perform, but in action we will not define how can we perform this action. We will define how to perform action into reducer function

Reducer:-reducer is a function in which we will define initial state and action in arguments and we will use switch case (for multiple action) through which we will return a updated or new state. It means reducer function always return a updated state. With the help of combineReducers function we will combine all the reducer and construct a root reducer

Store:-with the help of createStore Hook we will construct a store, and in this store we will use root reducer to configure our store with all reducers .With the help of useDispatch we can store value in centralized store by any component.

Now we can use centralized store in any component with the help of some Hooks and by using action and reducer function we can modify and get values in a centralized store. We can also use Redux Developer to configure Redux in our browser .And with the help of redux developer tool google chrome extension we can install or add extension in our chrome browser

useSelector:-with the help of useSelector hook we can find out the initial state corresponding to a reducer function,

and with the help of useDispatch hook we will change state corresponding to an action function

Steps in redux

We will create actions inside Actions.jsx

```

Actions.jsx > ...
1   function square()
2   {
3       return {
4           type:'SQUARE'
5       }
6   }
7   function cube()
8   {
9       return{
10           type:'CUBE'
11       }
12   }
13   export [square,cube]

```

We can construct reducers with the help of following function

```
> Reducers.jsx > CalculateSquare
let initialState=5;
function CalculateSquare(state=initialstate,action)
{
    switch(action.type)
    [
        case 'SQUARE':
            return state**2;
        case 'CUBE':
            return state**3;
        default:
            return state;
    ]
}
```

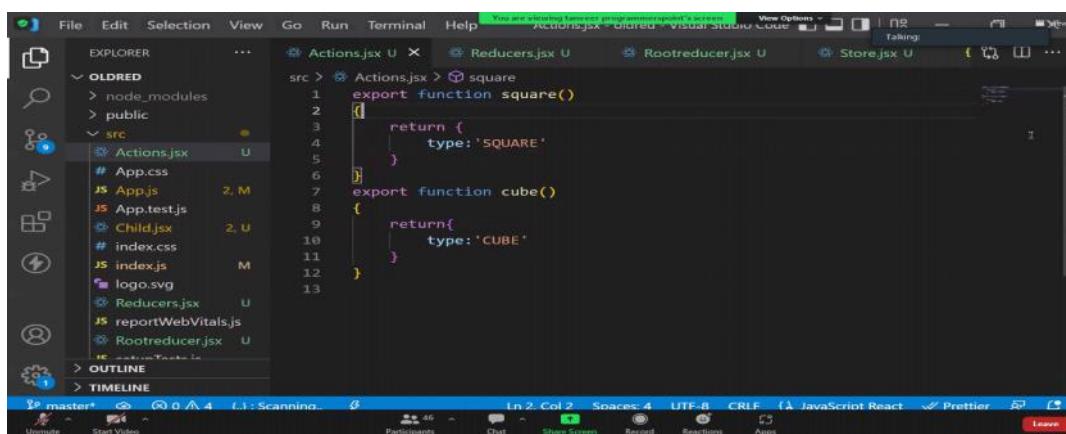
We can install redux with the help of following command

Npm install redux

Npm install

We can combine all reducers with the help of combineReducers function

```
> Rootreducer.jsx > ...
import { combineReducers } from "redux";
import CalculateSquare from './Reducers';
const rootreducer=combineReducers({
    CalculateSquare
})
export {rootreducer}
```



File Edit Selection View Go Run Terminal Help Reducers.jsx - oldred - Visual Studio Code Taking: 0 ms

EXPLORER OLDRED src > Actions.jsx U Reducers.jsx U Rootreducer.jsx U Store.jsx U

```
src > Reducers.jsx > calculateSquare
1 let initialstate=5;
2 export function calculateSquare(state=initialstate,action)
3 {
4     switch(action.type)
5     {
6         case 'SQUARE':
7             return state**2
8         case 'CUBE':
9             return state**3
10        default:
11            return state
12    }
13}
14
```

Ln 11, Col 28 Spaces: 4 UTF-8 CRLF JavaScript React Prettier

File Edit Selection View Go Run Terminal Help Rootreducer.jsx - oldred - Visual Studio Code Taking: 0 ms

EXPLORER OLDRED src > Rootreducer.jsx > ...
1 import { combineReducers } from "redux";
2 import { calculateSquare } from "./Reducers";
3 const rootreducer=combineReducers({
4 calculateSquare
5 })
6 export [rootreducer]

Ln 6, Col 21 Spaces: 4 UTF-8 CRLF JavaScript React Prettier

File Edit Selection View Go Run Terminal Help Store.jsx - oldred - Visual Studio Code Taking: 0 ms

EXPLORER Actions.jsx U Reducers.jsx U Rootreducer.jsx U Store.jsx X package.json M

```
src > Store.jsx > store
1 import { createStore } from 'redux';
2 import {rootreducer} from './Rootreducer';
3 let store=createStore(rootreducer,
4   | window.__REDUX_DEVTOOLS_EXTENSION__ && window.__REDUX_DEVTOOLS_EXTENSION__());
5 export default store;
```

Ln 4, Col 5 Spaces: 4 UTF-8 CRLF JavaScript React Prettier

File Edit Selection View Go Run Terminal Help index.js - oldred - Visual Studio Code Taking: 0 ms

EXPLORER OLDRED src > index.js > ...
1 import './index.css';
2 import App from './App';
3 import reportWebVitals from './reportWebVitals';
4 import { Provider } from 'react-redux';
5 import store from './Store';
6 const root = ReactDOM.createRoot(document.getElementById('root'));
7 root.render(
8 <React.StrictMode>
9 <Provider store={store}>
10 <App />
11 </Provider>
12 </React.StrictMode>
13);
14
15
16 // If you want to start measuring performance in your app, pass a f...

```

File Edit Selection View Go Run Terminal Help indexjs - oldred - Visual Studio Code
EXPLORER ... JS index.js M package.json M JS App.js 2, M Child.jsx 2, U JS index.js M ...
src > JS index.js > ...
  import './index.css';
  import App from './App';
  import reportWebVitals from './reportWebVitals';
  import { Provider } from 'react-redux';
  import store from './Store';
  const root = ReactDOM.createRoot(document.getElementById('root'));
  root.render(
    <React.StrictMode>
      <Provider store={store}>
        <App />
      </Provider>
    </React.StrictMode>
  );
// If you want to start measuring performance in your app, pass a f
// to log results (for example: reportWebVitals(console.log))
// or send to an analytics endpoint. Learn more: https://bit.ly/CRA
reportWebVitals();

```

Ln 13, Col 16 Spaces: 2 UTF-8 LF JavaScript Prettier

```

File Edit Selection View Go Run Terminal Help App.js - oldred - Visual Studio Code
EXPLORER ... JSX U package.json M JS App.js 2, M Child.jsx 2, U JS index.js M ...
src > JS App.js > [App]
  import { square, cube } from './Actions';
  import Child from './Child';
  const App = () => {
    store.subscribe(() => console.log(store.getState()));
    let init = useSelector((state) => state.calculateSquare);
    let dispatch = useDispatch();
    return (
      <h1>{init} App component is running</h1>
      <button onClick={() => {
        dispatch(square())
      }}>Square</button>
      <input type="number" value={init} />
      <button onClick={() => {
        dispatch(cube())
      }}>Cube</button>
      <Child/>
    )
  }
  export default App;

```

Ln 23, Col 2 Spaces: 2 UTF-8 LF JavaScript Prettier

```

File Edit Selection View Go Run Terminal Help Child.jsx - oldred - Visual Studio Code
EXPLORER ... JSX U package.json M JS App.js 2, M Child.jsx 2, U JS index.js M ...
src > JS Child.jsx > [Child]
  import { useDispatch, useSelector } from 'react-redux';
  import store from './Store';
  import { calculateSquare } from './Reducers';
  import { square, cube } from './Actions';
  const Child = () => {
    let init = useSelector((state) => state.calculateSquare);
    let dispatch = useDispatch();
    return (
      <h1>{init} Child component is running</h1>
      <button onClick={() => {
        dispatch(square())
      }}>Square</button>
      <input type="number" value={init} />
      <button onClick={() => {
        dispatch(cube())
      }}>Cube</button>
    )
  }

```

Ln 16, Col 19 Spaces: 4 CRLF JavaScript React Prettier

NEW Technique of Redux

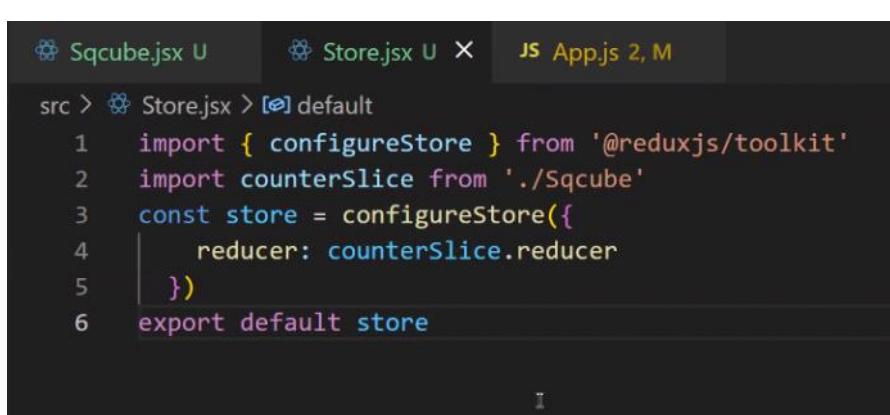
In new technique we will define action and inside the action we will use some functions, that returns type and we will use this action inside createSlice()

1st step we can construct redux by using createSlice and createSlice function contains slice name, initial state and reducer including actions and with the help of configureStore we will construct Store. configureStore contains an object in which we will pass our reducers and by using dispatch we will perform actions.

Installing redux using toolkit
We can use following command
Npm install @reduxjs/toolkit
npm install redux

```
import { createSlice } from '@reduxjs/toolkit'
const counterSlice = createSlice({
  name: 'squarecube',
  initialState: 5,
  reducers: {
    square: state => {
      state=state**2
    },
    cube: state => {
      state=state**3
    }
  }
})
export const { square, cube } = counterSlice.actions
```

```
import { createSlice } from '@reduxjs/toolkit'
const counterSlice = createSlice({
  name: 'squarecube',
  initialState: 5,
  reducers: {
    square: state => {
      return state=state**2
    },
    cube: state => {
      return state=state**3
    }
  }
})
export const { square, cube } = counterSlice.actions
export default counterSlice
```



```
src > Store.jsx > [o] default
  1 import { configureStore } from '@reduxjs/toolkit'
  2 import counterSlice from './Sqcube'
  3 const store = configureStore({
  4   reducer: counterSlice.reducer
  5 })
  6 export default store
```

```
src > JS App.js > [e] App
1 import logo from './logo.svg';
2 import './App.css';
3 import store from './Store';
4 import { useState } from 'react';
5 import {square,cube} from './Sqcube';
6 const App=()=>{
7     store.subscribe(() => console.log(store.getState()))
8     let[st,updatest]=useState(store.getState());
9     return(<>
10         <h1>App component {st}</h1>
11         <button onClick={()=>{
12             store.dispatch(square())
13             updatest(store.getState())
14         }}>square</button>
15         <input type="number" value={st} />
16     </>)
17 }
18 export default App;
```

3

Local storage is a place in browser memory which is used to store our data temporary or it's a predefined object in which we can call some functions, with the help of `setItem()` method we can store elements in local storage and with the help of `getItem()` method we can fetch items from local storage. With the help of `removeItem` method we can delete items

```
<h1 id="i"></h1>
<button onclick="show()">Delete</button>
<script>
    localStorage.setItem('name','ppoint');
    var t=localStorage.getItem('name');
    document.getElementById('i').innerHTML=t;
    function show()
    {
        localStorage.removeItem('name');
        //localStorage.setItem('name','');
        var t=localStorage.getItem('name');
        document.getElementById('i').innerHTML=t;
    }
</script>
```

```
import { useEffect, useState } from 'react';
const App=()=>{
  let [name,updatename]=useState();
  let [names,updatenames]=useState([]);
  useEffect(()=>{
    localStorage.setItem('nm',names);
  },[names])
  return(
    <>
    <h1>App component is running</h1>
    <input type="text" name="name" value={name} onChange={(e)=>{
      updatename(e.target.value);
    }} />
    <button onClick={()=>{
      let t=localStorage.getItem('nm');
      updatenames([...t,name]);
    }}>Add Name</button>
    </>
  )
}
```

Stringify a JavaScript Array

It is also possible to stringify JavaScript arrays:

Imagine we have this array in JavaScript:

```
const arr = ["John", "Peter", "Sally", "Jane"];
```

Use the JavaScript function `JSON.stringify()` to convert it into a string.

```
const myJSON = JSON.stringify(arr);
```

```
JS App.js M X
src > JS App.js > (x) default
1 import React from "react";
2 import { useState, useEffect } from "react";
3 const App=()=>{
4   let [name, updateName] = useState();
5   let [names, updateNames] = useState([]);
6   useEffect(()=>{
7     localStorage.setItem('nm', JSON.stringify(names));
8   },[names])
9   return(
10   <>
11   <h1>App component is running</h1>
12   <input type="text" name="name" value={name} onChange={(e)=>{
13     updateName(e.target.value);
14   }} />
15   <button onClick={()=>{
16     let t=JSON.parse(localStorage.getItem('nm'));
17     updateNames([...t, name]);
18   }}>Add Name</button>
19   </>
20   )
21 }
22 export default App;
```

Props type mainly used for defining data types of props, through which we can restrict props to store specific types value. We will import `propTypes` from `prop-types` and we can apply `propTypes` on a component with the help of following syntax `defaultProps` we can assign default props to a component ,through which we can assign default values in props when we will not pass any value at the time of component value then it will always take values from default props.it means it is not mandatory to pass props values at the time of component calling

```

Child.propTypes={

}

Child.defaultProps={

}

```

Python
Machine learning
{Ai
Deep learning }
{Analysis tubule power VI}
{SQL:- Mongo DB, Oracle, MySQL }

```

import React from "react";
import PropTypes from 'prop-types';
const Child={({roll,name,fees,subjects})=>{
    return (
        <>
        <h1>Child Component is running</h1>
        <h2>ROLL : {roll}</h2>
        <h2>NAME : {name}</h2>
        <h2>FEES : {fees*0.90}</h2>
        <h2>SUBJECTS : {subjects}</h2>
        </>
    )
}
Child.propTypes={
    roll:PropTypes.number,
    name:PropTypes.string,
    fees:PropTypes.number,
    subjects:PropTypes.array
}

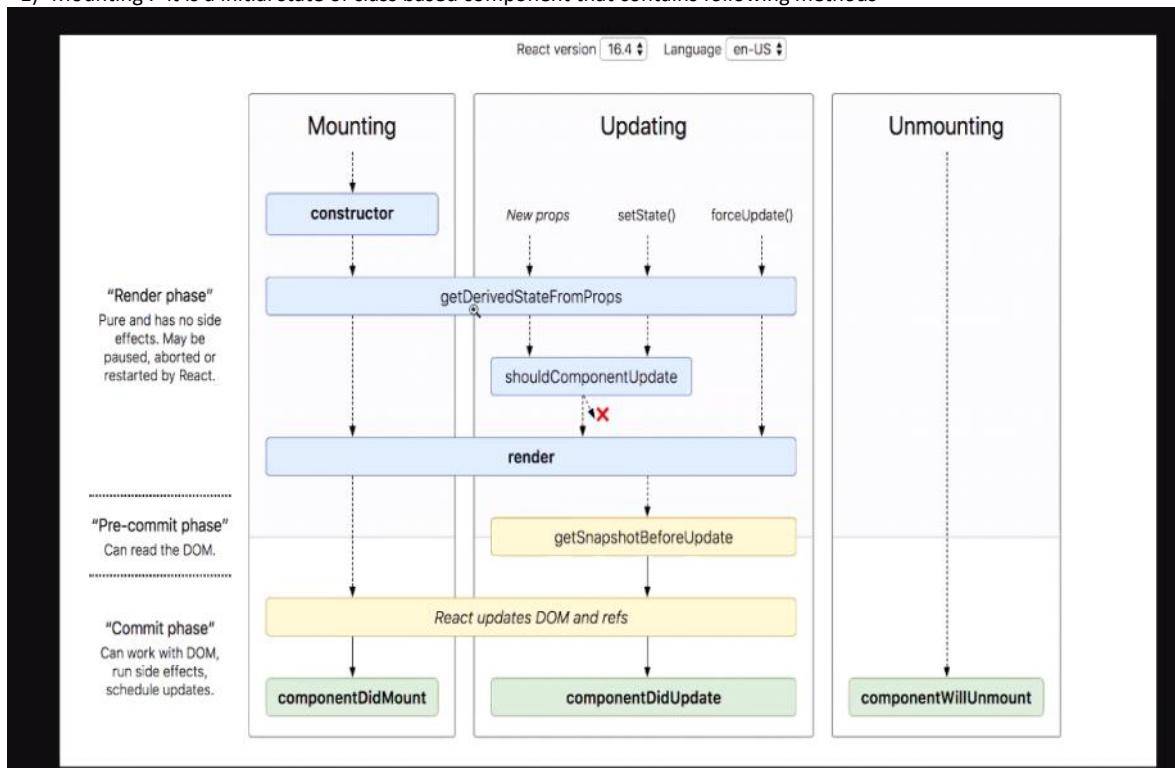
```

Screen clipping taken: 08-11-2022 11:39 AM

A class based component is a component in which we will use a class that inherit component class of React Module and we will write some methods through which we will perform all the operations with classes.

Life cycle of class based components contains following three states

- 1) Mounting :- it is a initial state of class based component that contains following methods



- The commonly used methods of mounting phase are
- Constructor:- as we know constructor always used to initialize the initial state for components
 - getDerivedStateFromProps:- it is used to identify values of props for a component
 - Render method:- this method is used to transfer Jsx elements from component to Dom
 - componentDidMount:-this method is used to update values of initial state with the help of setState() function
 - Rafce ctrl+space+enter shortcut key for creating component

```
import React from 'react'

class App extends React.Component
{
  constructor()
  {
    super();
    this.state={name:"ppoint"}
    console.log('constructor is running')
  }
  render()
  {
    console.log('render is running')
    return <h1>App component is running {this.state.name}</h1>
  }
}
export default App
```

setState method is used to update, modify or change value of a state variable

```
import React from 'react'
class App extends React.Component
{
  constructor()
  {
    super();
    this.state={name:"ppoint"}
    console.log('constructor is running')
  }
  static getDerivedStateFromProps(props,state)
  {
    console.log('get derived state from props is running'+props.course);
    return null;
  }
  render()
  {
    console.log('render is running'+this.state.name)
    return <h1>App component is running {this.state.name}</h1>
  }
  componentDidMount()
  {
    this.setState({name:'programmers'});
    console.log('componentDidMount is running')
  }
}
export default App
```

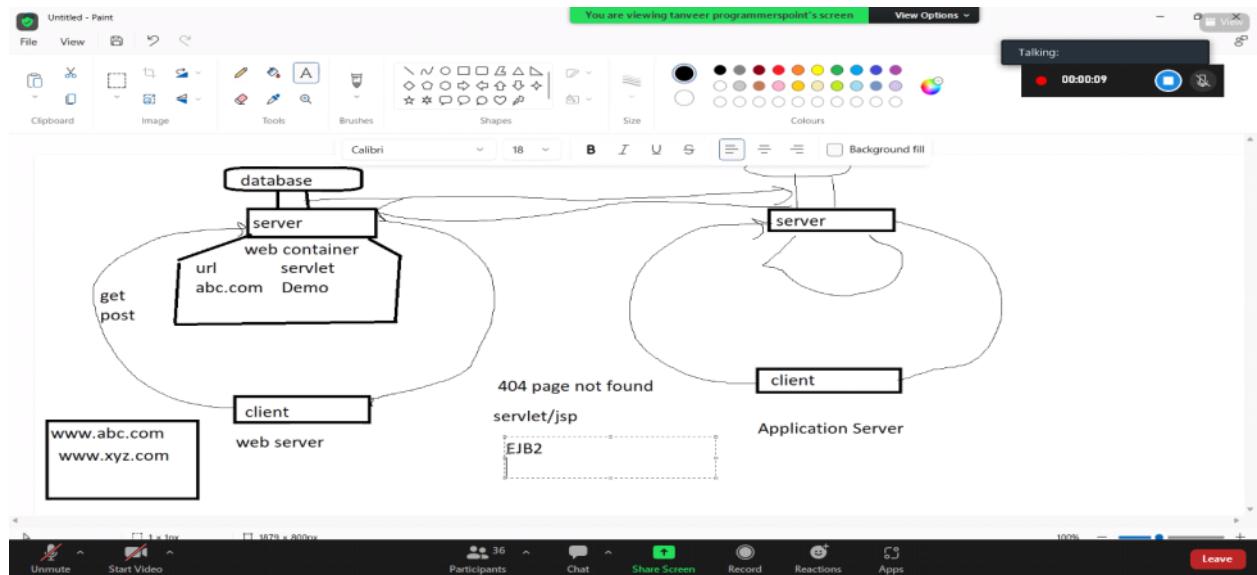
```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render([
  <React.StrictMode>
    | <App course="react js" />
  </React.StrictMode>
]);

```

Screen clipping taken: 21-11-2022 11:57 AM

```
| return (<>
|   <h1>App component is running {this.state.name}</h1>
|   <button onClick={()=>{this.setState({show:false})}}>click me</button>
|   {this.state.show?<Child/>:null}
| </> )
|
componentDidMount()
{
  this.setState({name:'programmers'});
  console.log('componentDidMount is running')
}
shouldComponentUpdate()
{
  console.log('should component update is running');
  return true;
}
getSnapshotBeforeUpdate(nextProps,prevState)
{
  console.log('getsnapshotbeforeupdate is running');
  console.log(nextProps);
  console.log(prevState);
  return {}
}
componentDidUpdate()
{
  console.log('component did update is running');
}
}
export default App|
```

Web terminology for servers



Server:- server is a place where all the upcoming request will be handled and it will generate response for all upcoming request.

Web container:- Web container is a part of server which is used to generate dynamic web pages according to request.

Get type request	Post type request
Get type request will show content from URL bar	Post type request will not show content from URL bar
We can construct bookmark for get type request	We can't construct bookmark for post type request
Get is reliable	Post is secure
We can transfer limited amount of data by using get request(approx. 2000)	We can transfer large amount of data by using post type request

Servlet:- servlet is a java program/servlet is a technology used to create dynamic web pages supported by server/Servlet is an interface

Javax.servlet package contains all the classes and interfaces for servlet technology

Classes like HttpServlets ,GenericServlets

Interfaces like HttpServletRequest, HttpServletResponse,HttpSessions or RequestDispatcher

Life cycle of servlet contains 3 stage

Page white that is loading ,2nd data display , 3rd page closed connection closed

And we will use 2 methods

Init() //

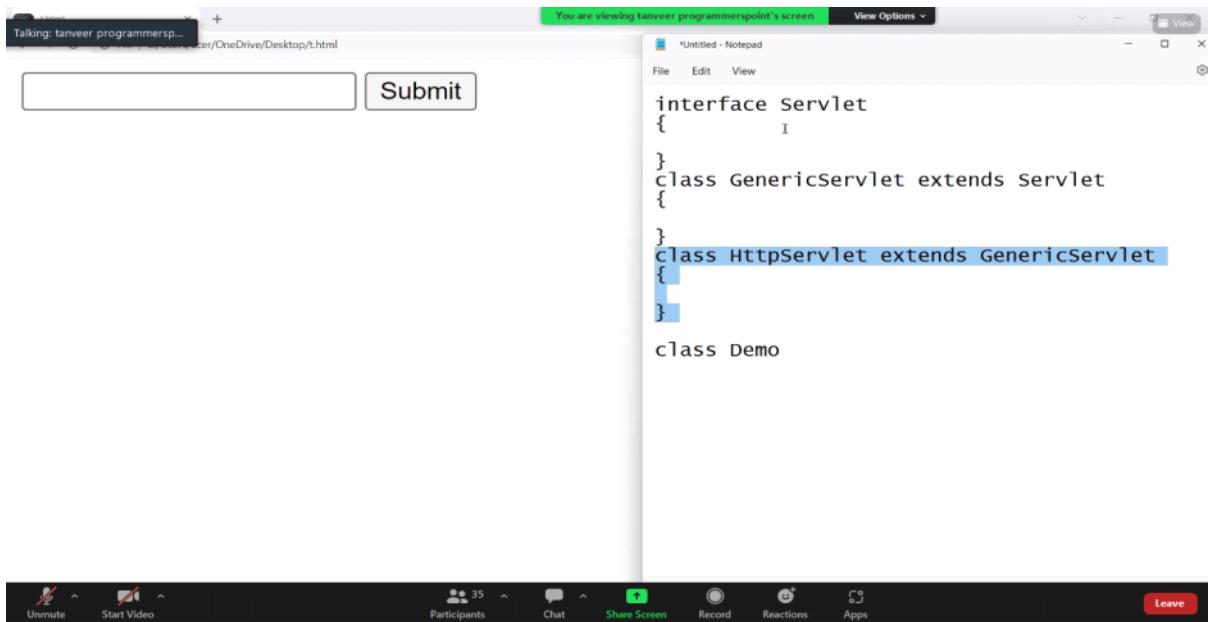
service()//Mainly we will use this

destroy()

Ways to create a servlet

We can create a servlet with the help of following 3 ways

By extend HttpServlet class , By extends GenericServlet class , By implements Servlet interface

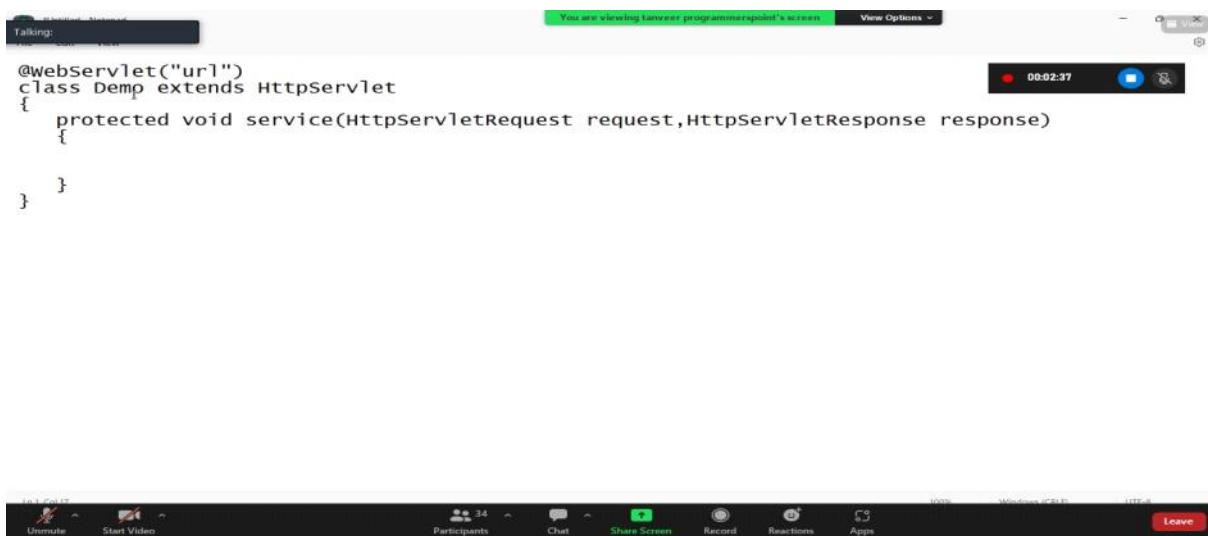


We can map a servlet URL with the help of following two ways

1st annotation mapping

We will represent annotation by @ symbol

And for servlet mapping we will use Webservlet annotation



Printing content in servlet:-We can print only text or html content in response which is generated by a servlet, when we will call println method of printStream class than it will return object type content which is supported by console but when we want to return text type content then we will call println method of PrintWriter class, it will return text type content which is supported by browser/client.

When we will call getWriter() method at response variable it will return object of PrintWriter class

```

1
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 /**
13 * Servlet implementation class Demo
14 */
15 @WebServlet("/dummy")
16 public class Demo extends HttpServlet {
17     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
18         //System.out.println("servlet is running.....");
19         PrintWriter out=response.getWriter();
20         out.println("Demo servlet is running");
21     }
22 }
23

```

Response content by servlet:- As we know `println` method of `PrintWriter` class return text type content, but if We want to display html content in response then we will call `setContentType()` method at response reference variable

In `setContentType()` method we will pass content type as String, usually we will pass `text/html` in arguments of `setContentType()`

```

1
2
3 import java.io.IOException;
4 import java.io.PrintWriter;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 /**
13 * Servlet implementation class Demo
14 */
15 @WebServlet("/dummy")
16 public class Demo extends HttpServlet {
17     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
18         //System.out.println("servlet is running.....");
19         PrintWriter out=response.getWriter();
20         response.setContentType("text/html");
21         out.println("Demo servlet is running");
22         out.println("<h1>helloooo</h1>");
23     }
24 }

```

1st program
For calculator

iZee6.30 - Java EE - Demo/WebContent/index.html - Eclipse

```

1<html>
2<head>
3<title>Insert title here</title>
4</head>
5<body>
6<h1>Project is running</h1>
7<a href="dummy">go to Demo</a>
8<form action="Calc">
9<input type="number" name="n1" /><br/>
10<input type="number" name="n2" /><br/>
11<input type="submit" name="ch" value="Add" />
12<input type="submit" name="ch" value="Sub" />
13<input type="submit" name="ch" value="Mul" />
14<input type="submit" name="ch" value="Div" />
15</form>
16</body>
17</html>

```

html/body/form/input/value

82°F Cloudy

Writable Smart Insert 14:42 ENG IN 18:17 26-08-2022

iZee6.30 - Java EE - Demo/src/CalcJava - Eclipse

You are viewing tanveer programmerspoint's screen View Options

```

8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 @WebServlet("/Calc")
12 public class Calc extends HttpServlet {
13     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,
14         PrintWriter out=response.getWriter();
15         response.setContentType("text/html");
16         String n1=request.getParameter("n1");
17         String n2=request.getParameter("n2");
18         String ch=request.getParameter("ch");
19         int a=Integer.parseInt(n1);
20         int b=Integer.parseInt(n2);
21         switch(ch)
22         {
23             case "Add":
24                 out.println("add is "+(a+b));
25                 break;
26             case "Sub":
27                 out.println("sub is "+(a-b));
28                 break;
29             case "Mul":
30                 out.println("mul is "+(a*b));
31                 break;
32             case "Div":
33                 try
34                 {
35                     out.println("div is "+(a/b));
36                 }
37                 catch(Exception e)
38                 {
39                     out.println("can't divide by zero");
40                 }
41                 break;
42         }
43     }
44 }

```

Unmute Start Video Participants Chat Share Screen Record Reactions Apps Leave

iZee6.30 - Java EE - Demo/src/CalcJava - Eclipse

Talking:

```

24         out.println("add is "+(a+b));
25         break;
26     case "Sub":
27         out.println("sub is "+(a-b));
28         break;
29     case "Mul":
30         out.println("mul is "+(a*b));
31         break;
32     case "Div":
33         try
34         {
35             out.println("div is "+(a/b));
36         }
37         catch(Exception e)
38         {
39             out.println("can't divide by zero");
40         }
41         break;
42     }
43 }
44 }

```

Writable Smart Insert 46:1 ENG IN 18:19 26-08-2022

```

1 import javax.servlet.annotation.WebServlet;
2 import javax.servlet.http.HttpServlet;
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 @WebServlet("/Alogin")
13 public class Alogin extends HttpServlet {
14     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,
15         PrintWriter out=response.getWriter();
16         response.setContentType("text/html");
17         String id=request.getParameter("id");
18         String pwd=request.getParameter("pwd");
19         if(id.equals("admin")&&pwd.equals("12345"))
20         {
21             response.sendRedirect("adminhome.html");
22         }
23         else
24         {
25             out.println("Invalid id and password");
26         }
27     }
28 }
29 
```

Project Started

sendRedirect method is used to transfer request or response from one servlet or jsp page to another servlet/jsp/html page

```

1 import javax.servlet.annotation.WebServlet;
2 import javax.servlet.http.HttpServlet;
3 import javax.servlet.http.HttpServletRequest;
4 import javax.servlet.http.HttpServletResponse;
5
6
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11 import javax.servlet.http.HttpServlet;
12
13 @WebServlet("/Alogin")
14 public class Alogin extends HttpServlet {
15     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException,
16         PrintWriter out=response.getWriter();
17         response.setContentType("text/html");
18         String id=request.getParameter("id");
19         String pwd=request.getParameter("pwd");
20         if(id.equals("admin")&&pwd.equals("12345"))
21         {
22             //response.sendRedirect("Ahomes");
23             RequestDispatcher rd=request.getRequestDispatcher("Ahomes");
24             rd.forward(request, response);
25         }
26         else
27         {
28             out.println("Invalid id and password");
29         }
30     }
31 
```

RequestDispatcher Interface is mainly used to transfer request and response from one servlet or jsp page to another servlet/jsp page/html page including current page content.

When we will call getRequestDispatcher at request reference variable it will return implementation class object of RequestDispatcher interface , we will use following two methods of RequestDispatcher interface

1st:-

forward() forward method is used to transfer request and response from one servlet or jsp page to another servlet or Jsp page including current page content.

It contains 2 arguments request and response

2nd:-

Include() method is used to transfer request and response from one servlet or jsp page to another servlet or jsp page including current page content and user message

The screenshot shows the Eclipse IDE interface with the Java EE - Demo project open. The Alogin.java file is displayed in the editor, containing Java code for a HttpServlet. The Tomcat log window shows startup messages and context reloads. The system tray at the bottom indicates it's 87°F Cloudy, and the taskbar shows various application icons.

```

12
13 @WebServlet("/Alogin")
14 public class Alogin extends HttpServlet {
15     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16         PrintWriter out=response.getWriter();
17         response.setContentType("text/html");
18         String id=request.getParameter("id");
19         String pwd=request.getParameter("pwd");
20         if(id.equals("admin")&&pwd.equals("12345"))
21         {
22             response.sendRedirect("adminhome.html");
23             //response.sendRedirect("Ahome");
24             //RequestDispatcher rd=request.getRequestDispatcher("Ahome");
25             //rd.forward(request, response);
26         }
27         else
28         {
29             out.println("Invalid id and password");
30         }
31     }
32 }
33 }
34 
```

The screenshot shows the Eclipse IDE interface with the Java EE - Demo project open. The Alogin.java file is displayed in the editor, with the include() method highlighted. A video call overlay shows a person named tanveer programmerspoint. The system tray at the bottom indicates it's 87°F Cloudy, and the taskbar shows various application icons.

```

12
13 @WebServlet("/Alogin")
14 public class Alogin extends HttpServlet {
15     protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
16         PrintWriter out=response.getWriter();
17         response.setContentType("text/html");
18         String id=request.getParameter("id");
19         String pwd=request.getParameter("pwd");
20         if(id.equals("admin")&&pwd.equals("12345"))
21         {
22             response.sendRedirect("adminhome.html");
23             //response.sendRedirect("Ahome");
24             //RequestDispatcher rd=request.getRequestDispatcher("Ahome");
25             //rd.forward(request, response);
26         }
27         else
28         {
29             RequestDispatcher rd=request.getRequestDispatcher("admin.html");
30             rd.include(request, response);
31             out.println("Invalid id and password");
32         }
33     }
34 
```

Sessions

As we know each and every request received by server should be https specific , http is a state less protocol it means it doesn't maintains state between client and server, and if we want to maintain state between client and server then we will use following 4 methods of sessions

1st:- URL rewriting

It is first and very common technique of session through which we can transfer values of url parameters

```

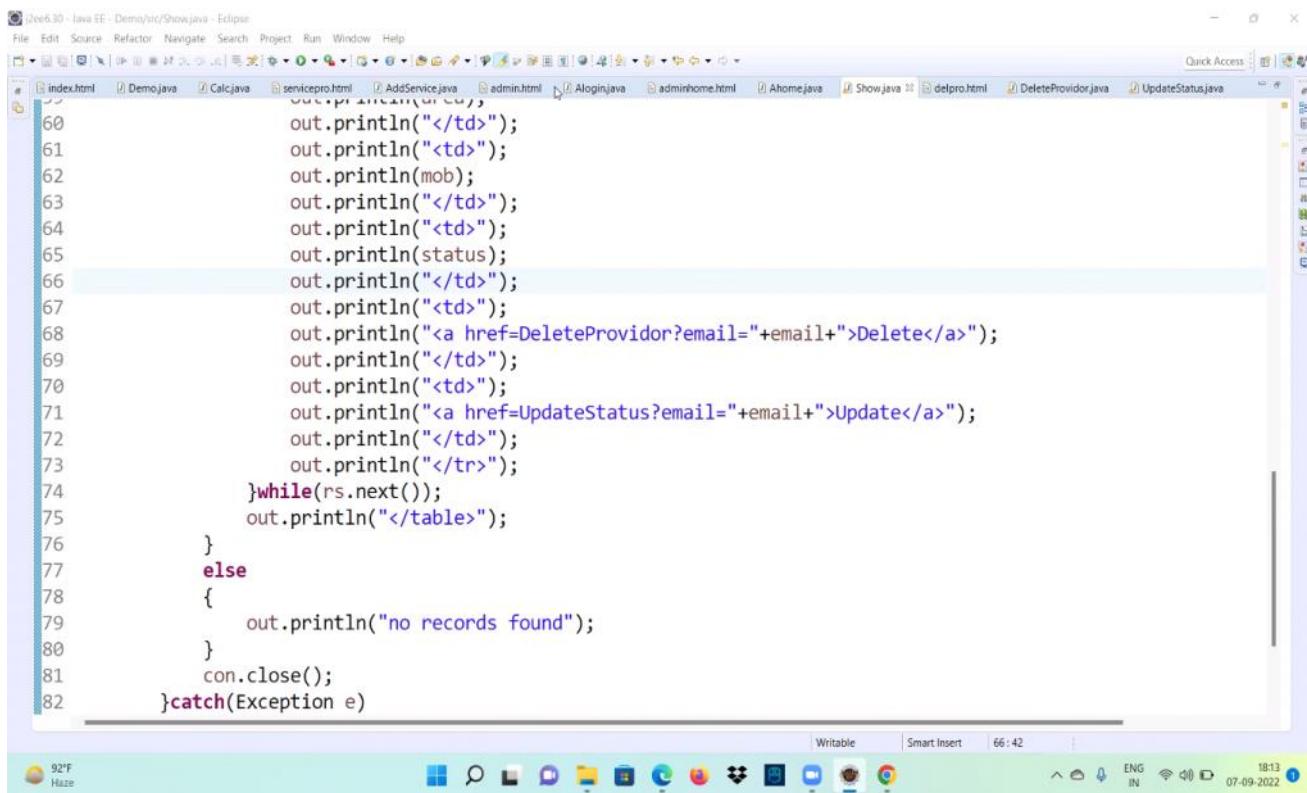
out.println("<td>");
out.println("<a href=DeleteProvider?email="+email+">Delete</a>");
out.println("</td>");
out.println("</tr>"); 
```

2nd:- Hidden Form Filled

it is another way to maintain session in java through which we can hiddenly transfer parameters and its value at the time of submit of form

We will use input type="hidden" to achieve hidden form filled technique in java

Update any service provider status can be done same as delete any service provider by creating a Servlet for it



The screenshot shows the Eclipse IDE interface with the following details:

- Title Bar:** Jee6.30 - Java EE - Demo/src>Show.java - Eclipse
- Menu Bar:** File, Edit, Source, Refactor, Navigate, Search, Project, Run, Window, Help
- Toolbar:** Standard Eclipse toolbar with various icons.
- Project Explorer:** Shows files like index.html, Demo.java, Calcjava, servicepro.html, AddService.java, admin.html, Alogin.java, adminhome.html, Ahome.java, Show.java, delpro.html, DeleteProvider.java, and UpdateStatus.java.
- Code Editor:** Displays Java code for a servlet named Show.java. The code prints a table from a database result set, including links for Delete and Update actions.
- Status Bar:** Shows Writable, Smart Insert, 66:42, ENG IN, and a date/time stamp of 07-09-2022 18:13.
- System Tray:** Shows system icons including battery level (92%), network, volume, and date/time.

```
60         out.println("</td>");
61         out.println("<td>");
62         out.println(mob);
63         out.println("</td>");
64         out.println("<td>");
65         out.println(status);
66         out.println("</td>");
67         out.println("<td>");
68         out.println("<a href=DeleteProvider?email=" + email + ">Delete</a>");
69         out.println("</td>");
70         out.println("<td>");
71         out.println("<a href=UpdateStatus?email=" + email + ">Update</a>");
72         out.println("</td>");
73         out.println("</tr>");
74     }while(rs.next());
75     out.println("</table>");
76 }
77 else
78 {
79     out.println("no records found");
80 }
81 con.close();
82 }catch(Exception e)
```

3rd HttpSession interface:-

It is 3rd and very common technique of session, which is mainly used to maintain session from login to log out. When we will getSession() method at request reference variable it will return implementation classes object of HttpSession interface commonly used methods of Http

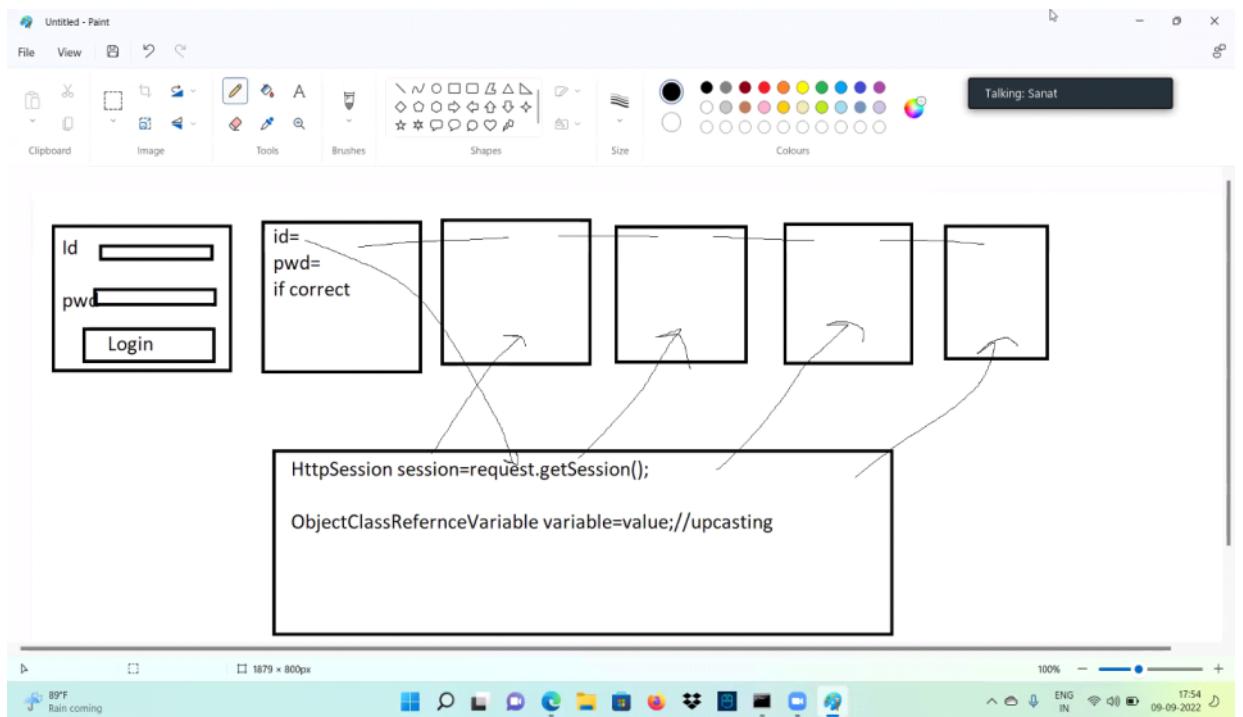
setAttribute():- this method is used mainly to set values in session, it contains 2 arguments 1st one is session variable name and 2nd one is its value

Syntax:-

getAttribute():- this method is use to fetch values from session , it will always return upcasted value.

setCurly() & getCurly()

Void invalidate():- invalidate is used to remove all the values from session ,we will call it at the time of logout



Cookies:-it is another way to maintain session in java, the major drawback of cookies is it is stored at client side that is browser. So when user will disable or remove cookies from browser than it will not work. In java Cookie is a class which is stores in javax.servlet package and Cookie class constructor contains 2 arguments 1st one is name of cookie and 2nd one is value of cookie ,we can transfer any cookie with response , with the help of response.addCookie() and we can fetch all the cookies from upcoming request with the help of getCookies() method and getCookie() ,method will return a cookie array from which we can get name and values of cookies

```

15 public class Alogin extends HttpServlet {
16     protected void service(HttpServletRequest request, HttpServletResponse response) throws
17         PrintWriter out=response.getWriter();
18         response.setContentType("text/html");
19         String id=request.getParameter("id");
20         String pwd=request.getParameter("pwd");
21         if(id.equals("admin")&&pwd.equals("12345"))
22         {
23             //response.sendRedirect("adminhome.html");
24             Cookie ck=new Cookie("id",id);
25             response.addCookie(ck);
26             response.sendRedirect("Ahome");
27             //RequestDispatcher rd=request.getRequestDispatcher("Ahome");
28             //rd.forward(request, response);
29         }
30     else
31     {
32         RequestDispatcher rd=request.getRequestDispatcher("admin.html");
33         rd.include(request, response);
34         out.println("Invalid id and password");
35     }
36 }
37

```

```
import java.io.IOException;
import java.io.PrintWriter;

import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/Ahome")
public class Ahome extends HttpServlet {
    protected void service(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        PrintWriter out=response.getWriter();
        response.setContentType("text/html");
        //String id=request.getParameter("id");
        //String pwd=request.getParameter("pwd");
        //out.println(id+" "+pwd);
        Cookie ck[]=request.getCookies();
        out.println(ck[0].getName()+" "+ck[0].getValue());
    }
}
```

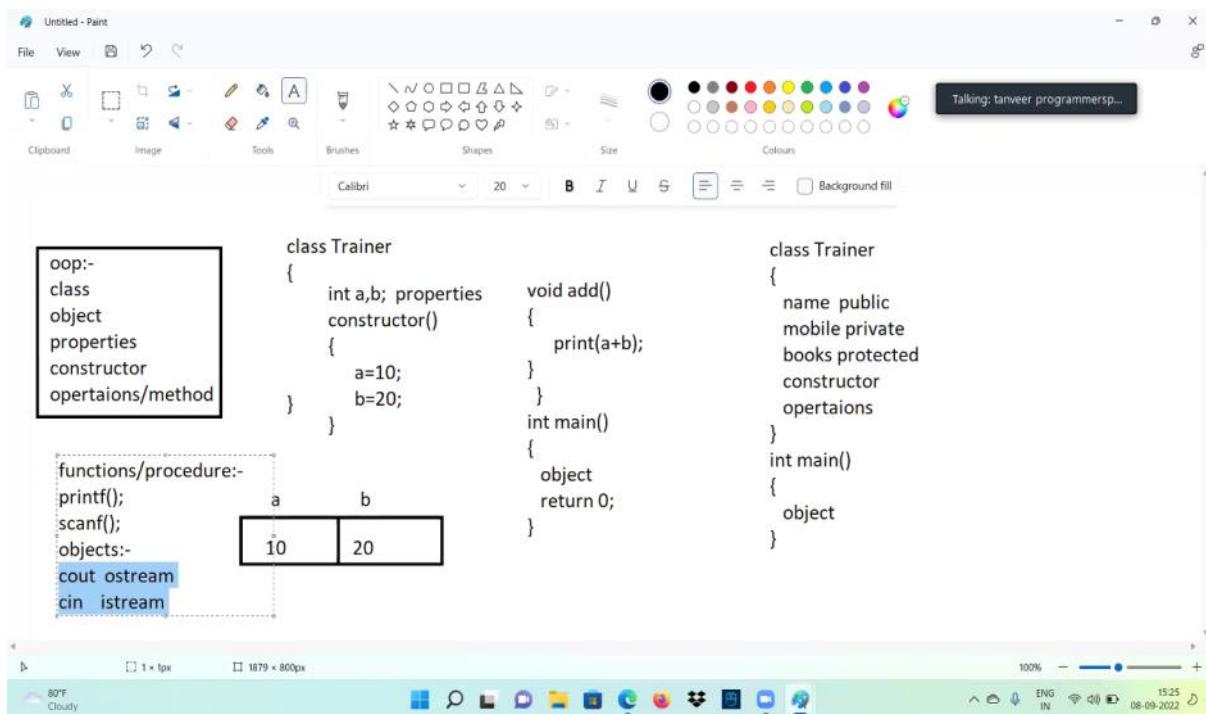
C++

Object:-

Os/Operating system/Disk Operating system

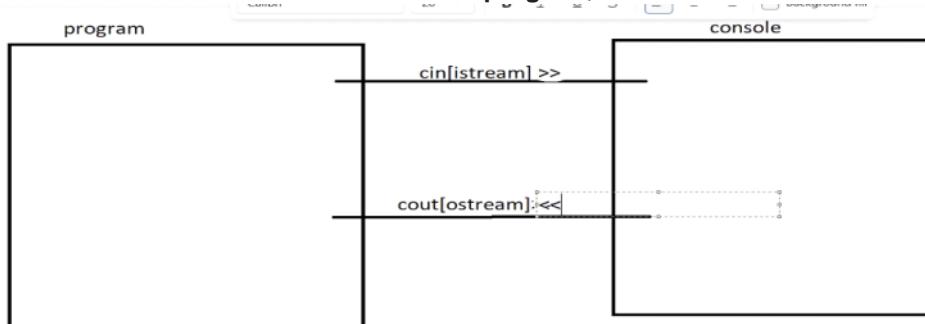
Command line argument in c is another way to take input from user at runtime in c, we can write a main method in c program that Can take 2 arguments 1st one is integer that is used to count number of arguments and 2nd one is pointer array of character array of pointers which is used to store values inside an array. We can take elements in array at the time of run by command prompt. Elements should be separated by spaces

```
#include<stdio.h>
int main(int n,char *args[])
{
    printf("number of arguments are %d\n",n);
    for(int i=1;i<n;i++)
    {
        printf("%s ",args[i]);
    }
    return 0;
}
```



A class is a blueprint of Object,

A C++ stream is a flow of data into or out of a program, such as the data written to cout or read from cin.



```
pro *first - Notepad
File Edit View
#include<iostream.h>
int main()
{
    cout<<"hello we are learning cpp";
    return 0;
}

files
ea pp & tt namespace tanveer
ab & dk namespace indore
+ #include<iostream>
<
```

main.cpp		Run	Output
			/tmp/CSAZpexzGi.o
? #include <iostream>			Hello world!
}+ int main() { I			
// Write C++ code here			
; std::cout << "Hello world!";			
;			
? return 0;			
} }			

scope resolution operator is ::

```
#include<iostream>
//using namespace std;
int main()
{
    //cout<<"hello this is our first c++ program";
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    int a=1245;
    cout<<"hello\n" << "cpp";
    return 0;
}
```

```
#include<iostream>
//using namespace std;
int main()
{
    //cout<<"hello this is our first c++ program";
    std::cout<<"hello this is our first c++ program";
    return 0;
}
```

endl:-used to change line

```
#include<iostream>
using namespace std;
int main()
{
    int a=1245;
    cout<<"hello"<<endl<<"hii"<<10<<endl<<12.5<<endl<<a;
    return 0;
}
```

Escape character can work In c++

```
#include<iostream>
using namespace std;
int main()
{
    int a=1245;
    cout<<"hello\n"=><"cpp";
    return 0;
}
```

```
#include<iostream>
using namespace std;
int main()
{
    int a;
    float b;
    char c;
    cout<<"enter a int number"<<endl;
    cin>>a;
    cout<<"enter a float number"<<endl;
    cin>>b;
    cout<<"enter a char"<<endl;
    cin>>c;
    cout<<"value of a is "<<a<<" and b is "<<b<<" and c is "<<c;
    return 0;
}
```

```

#include<iostream>
using namespace std;
int main()
{
    int a;
    float b;
    char c;
    cout<<"enter a int then float number then charctors" << endl;
    cin>>a>>b>>c;
    cout<<"value of a is "<<a<<" and b is "<<b<<" and c is "<<c;
    return 0;
}

```

Class :- class is a blueprint of an object in which we will write data members and member function. The basic structure of a c++ class is

```

class Name
{
    acces modifiers:
        properties
        constructor
        method
};
int main()
{
    object
}

```

By default data members of a class and member function in C++ are private.

Constructor are those member function of class, whose name is same as that of class. Constructor doesn't have any

Return type not even void, when we will create object of class then constructor will automatically called

Constructor are mainly uses to initialize properties of an object

In c++ we will use following types of constructor

Default constructor:- constructor that doesn't contains any arguments .When we will not write any constructor inside a c++ class then compiler will write default constructor inside a c++ class

Parametrized constructor:- a constructor that contains parameters in arguments is called parametrized constructor

```

#include<iostream>
using namespace std;
class Emp
{
public:
    int id;
    int sal;
    Emp(int i,int s)
    {
        id=i;
        sal=s;
    }
};
int main()
{
    Emp e(101,2500);
    cout<<e.id<<" "<<e.sal<<endl;
    Emp e1(102,2800);
    cout<<e1.id<<" "<<e1.sal<<endl;
    return 0;
}

```

Call by Value

```
#include<iostream>
using namespace std;

void show(int a,int b)
{
    a=a+b;
    b=a-b;
    a=a-b;
    cout<<" a is "<<a<<" and b is "<<b<<endl;
}
int main()
{
    int i=10,j=20;
    cout<<"before show calling i is "<<i<<" and j is "<<j<<endl;
    show(i,j);
    cout<<"after show calling i is "<<i<<" and j is "<<j<<endl;
    return 0;
}
```



Call By reference

```
#include<iostream>
using namespace std;
void display(int *p1,int *p2)
{
    *p1=*p1+*p2;
    *p2=*p1-*p2;
    *p1=*p1-*p2;
    cout<<" p1 is "<<*p1<<" and p2 is "<<*p2<<endl;
}
int main()
{
    int i=10,j=20;
    cout<<"before show calling i is "<<i<<" and j is "<<j<<endl;
    display(&i,&j);
    cout<<"after show calling i is "<<i<<" and j is "<<j<<endl;
    return 0;
}
```



Copy Constructor:- It is mainly used to construct exact copy of object, it mean in copy constructor we will write code to copy properties from old object to new object

```
#include<iostream>
using namespace std;
class Emp
{
};

int main()
{
    Emp e;
    cout<<"address of e is "<<&e<<endl;
    Emp e1=e;//copy constructor
    cout<<"address of e1 is "<<&e1<<endl;
    return 0;
}
```

```

#include<iostream>
using namespace std;
class Emp
{
public:
    int id;
    Emp(int i)
    {
        id=i;
    }
    Emp(Emp &o)
    {
    }
};
int main()
{
    Emp e(101);
    Emp e2=e;
    return 0;
}

```

Pointers doesn't work with copy constructor but address of memory does.

```

#include<iostream>
using namespace std;
class Emp
{
public:
    int id;
    int sal;
    //parametrized constructor
    Emp(int i,int s)
    {
        id=i;
        sal=s;
    }
    //copy constructor
    Emp(const Emp &old)
    {
        id=old.id;
        sal=old.sal;
    }
};
int main()
{
    Emp e(101,2500);
    Emp e1=e;//calling of copy constructor
    cout<<e.id<<" "<<e.sal<<endl;
    cout<<e1.id<<" "<<e1.sal<<endl;
}

```

When we will not write any constructor at the time of c++ class declaration then c++ compiler always write default constructor, but if user will write any constructor(default or parametrized) then compiler will not write default constructor

As we know constructor is used to initialise properties of an object and allocate a memory while destructor will be called automatically when object memory should be released. **Destructor is never parametrized**

```

#include<iostream>
using namespace std;
class Emp
{
public:
    Emp()
    {
        cout<<"memory sucessfully alocated" << endl;
    }
    ~Emp()
    {
        cout<<"memory sucessfully realsed" << endl;
    }
};
int main()
{
    Emp e;
    return 0;
}

```

Constructor overloading or compile time polymorphism

We can write same constructor with different arguments in a c++ class, this process is called overloading of constructor. But when we will write two or more constructor with exactly same syntax then compiler will produce an error

```

#include<iostream>
using namespace std;
class Emp
{
public:
    Emp()
    {
        cout<<"default constructor is called" << endl;
    }
    Emp(int a)
    {
        cout<<"1 parameter constructor called" << endl;
    }
    Emp(int a,int b)
    {
        cout<<"2 parameter constructor called" << endl;
    }
};
int main()
{
    Emp e;
    Emp e1(5);
    Emp e2(5,10);
    return 0;
}

```

Method are member functions that can be repeatedly used in our c++ program. We will use 4 types of methods

```
#include<iostream>
using namespace std;
class Emp
{
public:
    void show()
    {
        cout<<"show is running" << endl;
    }
    void cube(int a)
    {
        cout<<"cube is "<<a*a*a << endl;
    }
};
int main()
{
    Emp e;
    e.show();
    e(cube(5));
    return 0;
}
```

```
#include<iostream>
using namespace std;
class Demo
{
public:
    int getCode()
    {
        return 101;
    }
    int factorial(int n)
    {
        int f=1;
        for(int i=1;i<=n;i++)
        {
            f=f*i;
        }
        return f;
    }
};
int main()
{
    Demo d;
    cout<<d.getCode()<< endl;
    cout<<"factorial of 5 is "<<d.factorial(5)<< endl;
    int c=d.getCode();
    cout<<c<< endl;
```

```

        return 1;
    }
};

int main()
{
    Demo d;
    cout<<d.getCode()<<endl;
    cout<<"factorial of 5 is "<<d.factorial(5)<<endl;
    int c=d.getCode();
    cout<<c<<endl;
    c=d.factorial(6);
    cout<<c<<endl;
    return 0;
}

#include<iostream>
using namespace std;
class Addition
{
private:
    int a,b;//properties of an object
public:
    Addition()
    {
        cout<<"enter first number"<<endl;
        cin>>a;
        cout<<"enter second number"<<endl;
        cin>>b;
    }
    void show()
    {
        cout<<"add is "<<a+b<<endl;
    }
};
int main()
{
    Addition t1;
    t1.show();
    return 0;
}

```

Method overloading or compile time polymorphism

```

#include<iostream>
using namespace std;
class Demo
{
public:
    void show()
    {
        cout<<"show is running" << endl;
    }
    void show(int a)
    {
        cout<<"cube is "<<a*a*a<< endl;
    }
    void show(int a,int b)
    {
        cout<<"mul is "<<a*b<< endl;
    }
};
int main()
{
    Demo d;
    d.show();
    d.show(5);
    d.show(10,20);
    return 0;
}

```

Is it possible to write two main methods in c++ program, yes it is possible in condition where it is written inside a class

```

#include<iostream>
using namespace std;
class Demo
{
public:
    void main()
    {
        cout<<"main is running" << endl;
    }
    void main(int a)
    {
        cout<<"cube is "<<a*a*a<< endl;
    }
    void main(int a,int b)
    {
        cout<<"mul is "<<a*b<< endl;
    }
};
int main()
{
    Demo d;
    d.main();
    d.main(5);
    d.main(10,20);
    return 0;
}

```

Scope Resolution Operator ::

scope resolution operator is used to access data members and member function outside the class or block scope

```

#include<iostream>
using namespace std;
class Demo
{

```

```
#include<iostream>
using namespace std;
class Demo
{
public:
    Demo(); //declaration
    void show(); //declaration
};
Demo::Demo()
{
    cout<<"default constructor is running" << endl;
}
void Demo::show()
{
    cout<<"show is running" << endl;
}
int main()
{
    Demo d;
    d.show();
    return 0;
}
```

Namespace

```
#include<iostream>
using namespace std;
namespace pp1
{
    void show()
    {
        cout<<"show is running" << endl;
    }
    void display()
    {
        cout<<"display is running" << endl;
    }
}
int main()
{
    display();
    pp1::show();
    return 0;
}
```

```

#include<iostream>
using namespace std;
namespace pp1
{
    void show()
    {
        cout<<"show is running" << endl;
    }
    class Temp
    {
        public:
        void welcome()
        {
            cout<<"welcome to my temp class" << endl;
        }
    };
    Temp t;
}
void display()
{
    cout<<"display is running" << endl;
}
//using namespace pp1;
int main()
{
    display();
    pp1::show();
    pp1::t.welcome();
    return 0;
}

```

Object type pointers or reference

A object type pointer is mainly used to store address of object memory, if we want to access data members and members function by using object type pointer or reference object

```

#include<iostream>
using namespace std;
class Demo
{
    public:
    int a;
    int b;
    void show()
    {
        cout<<"show is running" << endl;
    }
};
int main()
{
    Demo d;
    d.a=10;
    d.b=20;
    cout<<d.a<<" "<<d.b<< endl;
    Demo *p;
    p=&d;
    p->a=50;
    p->b=80;
    cout<<p->a<<" "<<p->b<< endl;
    d.show();
}

```

```

p->b=80;
cout<<p->a<<" "<<p->b<<endl;
d.show();
p->show();
return 0;

```

THIS POINTER

This is a pointer which is mainly used to point current class object, we can use this inside member function of class. If we will use local or global both variables with same name and we want to transfer value from local to global we will use this pointer

```

#include<iostream>
using namespace std;
class Emp
{
public:
    int id,sal;//global variable or object properties
    Emp(int id,int sal)
    {
        this->id=id;
        this->sal=sal;
    }
    void show()
    {
        cout<<this->id<<" "<<this->sal<<endl;
    }
};
int main()
{
    Emp e(101,2500);
    e.show();

    return 0;
}

```

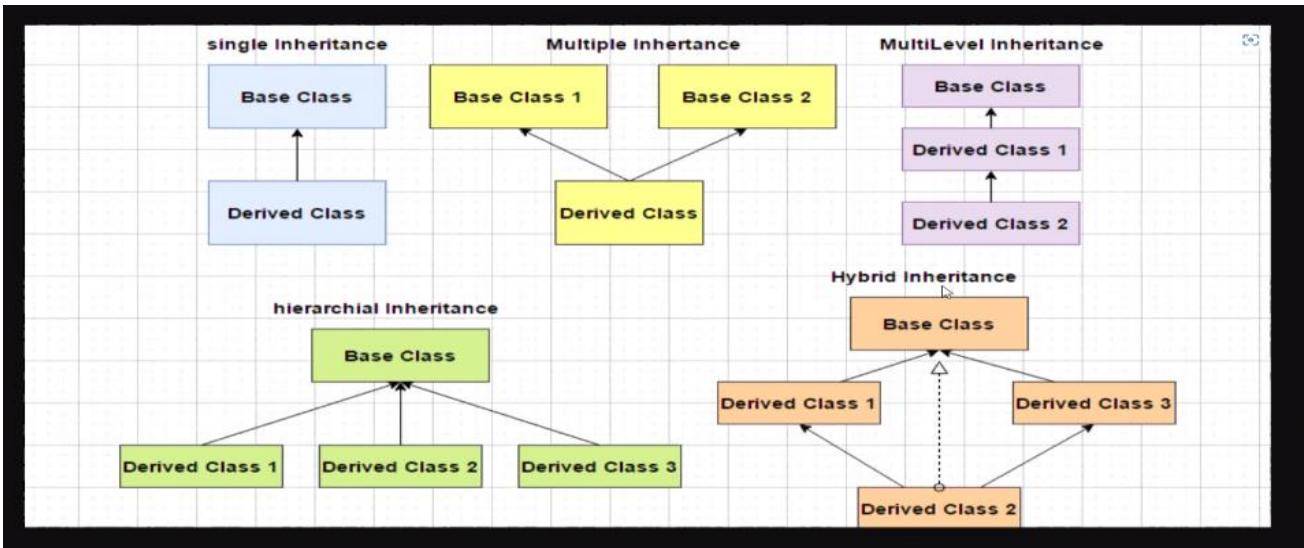
```

#include<iostream>
using namespace std;
class Emp
{
public:
    int id,sal;//global variable or object properties
    Emp(int id,int sal)
    {
        this->id=id;
        this->sal=sal;
        cout<<"address of a object e in constructor "<<this<<endl;
    }
    void show()
    {
        cout<<this->id<<" "<<this->sal<<endl;
        cout<<"address of a object e in method "<<this<<endl;
    }
};
int main()
{
    Emp e(101,2500);
    e.show();
    Emp *p;
    p=&e;
    cout<<"address of a object e in main "<<p<<endl;
    return 0;
}

```

Inheritance

Inheritance is mainly used for code reusability in c++, we can inherit data members and members functions of one class into another class with the help of inheritance, in c++ we will use following types of inheritance



When we will inherit a class with private access modifier then public, private ,protected data members should be

When we will inherit a class with protected access modifier then private data members of parent class should be private in child class

When we will inherit a class with public access modofiers then public private protected data members of class should be public private and protected in child class

Multilevel Inheritance

```
#include<iostream>
using namespace std;
class Calculator
{
public:
    void add(int a,int b)
    {
        cout<<"add is "<<a+b<<endl;
    }
    void sub(int a,int b)
    {
        cout<<"sub is "<<a-b<<endl;
    }
    void mul(int a,int b)
    {
        cout<<"mul is "<<a*b<<endl;
    }
    void div(int a,int b)
    {
        cout<<"div is "<<a/b<<endl;
    }
};
class Scalculator:public Calculator
{
public:
    void square(int n)
    {
        cout<<"square of "<<n<<" is "<<n*n<<endl;
    }
    void cube(int n)
    {
        cout<<"cube of "<<n<<" is "<<n*n*n<<endl;
    }
}
```

```

public:
    void square(int n)
    {
        cout<<"square of "<<n<<" is "<<n*n<<endl;
    }
    void cube(int n)
    {
        cout<<"cube of "<<n<<" is "<<n*n*n<<endl;
    }
};

class Demo:public Scalculator
{
public:
public:
    void show()
    {
        cout<<"show is running from Demo"<<endl;
    }
};

int main()
{
    Demo d;
    d.show();
    d.square(5);
    d.cube(5);
    d.add(20,30);
    d.sub(50,30);
    d.mul(20,30);
    d.div(90,30);
    return 0;
}

```

Screen clipping taken: 30-09-2022 03:22 PM

HEIRARICAL Inheritance

```

#include<iostream>
using namespace std;
class Base
{
public:
    void welcome()
    {
        cout<<"welcome to base"<<endl;
    }
};

class Demo:public Base
{
public:
    void show()
    {
        cout<<"show is running from Demo"<<endl;
    }
};

class Dummy:public Base
{
public:
    void show()
    {
        cout<<"show is running from Dummy"<<endl;
    }
};

```

```

        }
    };
int main()
{
    Demo d;
    d.show();
    d.welcome();
    Dummy d1;
    d1.show();
    d1.welcome();
}

```

Multiple Inheritance

```

#include<iostream>
using namespace std;
class Base
{
public:
    void welcome()
    {
        cout<<"welcome to base"=<endl;
    }
};
class Dummy
{
public:
    void display()
    {
        cout<<"display is running from Dummy"=<endl;
    }
};
class Demo:public Base,public Dummy
{
public:
    void show()
    {
        cout<<"show is running from Demo"=<endl;
    }
};
int main()
{
    Demo d;
    d.show();
    d.welcome();
    d.display();
}

```

Hybrid Inheritance

```

#include<iostream>
using namespace std;
class BaseSuper
{
public:
    void greetings()
    {
        cout<<"greetings from BaseSuper"=<<endl;
    }
};
class Base:public BaseSuper
{
public:
    void welcome()
    {
        cout<<"welcome to base"=<<endl;
    }
};
class Dummy:public Base
{
public:
    void display()
    {
        cout<<"display is running from Dummy"=<<endl;
    }
};
class Demo:public Base
{
public:
    void show()
    {
        cout<<"show is running from Demo"=<<endl;
    }
};
int main()
{
    Demo d;
    d.show();
    d.welcome();
    d.greetings();
    Dummy d1;
    d1.display();
    d1.welcome();
    d1.greetings();
}

```

Ambiguity problem arrives in case of multiple inheritance when both parent classes contains exact same method, the solution for ambiguity problem is virtual function

```

#include<iostream>
using namespace std;
class Base
{
public:
    void display()
    {
        cout<<"display is running from Base" << endl;
    }
};
class Dummy
{
public:
    void display()
    {
        cout<<"display is running from Dummy" << endl;
    }
};
class Demo:public Base,public Dummy
{
public:
    void show()
    {
        cout<<"show is running from Demo" << endl;
    }
};
int main()
{
    Demo d;
    d.show();
    d.display();
}

```

Method Overriding or run time polymorphism:-

If parent and child classes both contains exact same method with same signature and parameters then at run time content of parent class method will replace by child class method , this process is known as method overriding or run time polymorphism

```

#include<iostream>
using namespace std;
class Trainer
{
public:
    void intro()
    {
        cout<<"intro of trainer" << endl;
    }
    void skills()
    {
        cout<<"skills of a language" << endl;
    }
};
class Student:public Trainer
{
public:
    void intro()
    {
        cout<<"intro of student" << endl;
    }
};

```

```

int main()
{
    Student s;
    s.intro();
    s.skills();
}

```

Access modifiers are mainly used to define accessibility and life time of data members and member function in c++. In c++ there are 3 types of access modifiers(private,protected,public) by default data members and member function in c++ should be private but we can also define data members and member function as private and protected

Private data members are accessible within class only, we can't access private data members outside or derived class

	Within class	Outside class	Inheritance
Private	yes	No	No
Protected	Yes	No	Yes
Public	Yes	Yes	Yes

1st Private:-

```

#include<iostream>
using namespace std;
class Demo
{
private:
    int a;
public:
    void show()
    {
        a=10;
        cout<<"value of a is "<<a<<endl;
    }
};
class Dummy:public Demo
{
public:
    void display()
    {
        //a=10;
        //cout<<"value of a is "<<a<<endl;
    }
};
int main()
{
    Demo d;
    //d.a=20;
    d.show();
}

```

2nd:-Protected

```

#include<iostream>
using namespace std;
class Demo
{
protected:
    int a;
public:
    void show()
    {
        a=10;
        cout<<"value of a is "<<a<<endl;
    }
};
class Dummy:public Demo
{
public:
    void display()
    {
        a=10;
        cout<<"value of a is "<<a<<endl;
    }
};
int main()
{
    Demo d;
    d.show();
    //d.a=20;
    Dummy d1;
    d1.display();
}

```

3rd:-Public

```

#include<iostream>
using namespace std;
class Demo
{
public:
    int a;
    void show()
    {
        a=10;
        cout<<"value of a is "<<a<<endl;
    }
};
class Dummy:public Demo
{
public:
    void display()
    {
        a=10;
        cout<<"value of a is "<<a<<endl;
    }
};
int main()
{
    Demo d;
    d.a=20;
    d.show();
    Dummy d1;
    d1.display();
}

```

Encapsulation

It is a process of data Binding, we can bind data members and members of class in single unit is called encapsulation, we can achieve encapsulation in c++ class when we will define private properties and either a public constructor or getter methods or setter or getter methods to store or retrieve values in private properties

```
#include<iostream>
using namespace std;
class Emp
{
    int id,sal;
public:
    void setId(int id)
    {
        this->id=id;
    }
    void setSal(int sal)
    {
        this->sal=sal;
    }
    int getId()
    {
        return id;
    }
    int getSal()
    {
        return sal;
    }
};
int main()
{
    Emp e;
    //e.id=101;
    e.setId(101);
    e.setSal(25000);
    cout<<e.getId()<<" "<<e.getSal();
}
```

```

#include<iostream>
using namespace std;
class Emp
{
    int id,sal;
public:
    Emp(int id,int sal)
    {
        this->id=id;
        this->sal=sal;
    }
    int getId()
    {
        return id;
    }
    int getSal()           I
    {
        return sal;
    }
};
int main()
{
    Emp e(101,36000);
    cout<<e.getId()<<" "<<e.getSal();
}

```

Virtual Keyword and Function

As we know we can store, Base class object's address in Base class pointer variable and Derived class object's address in Derived class pointer variable. But in c++ we have a special concept through which we can store Derived class object's address into Base class pointer variable but when we will call same method(Base & Derived class same method) on a pointer variable then it will call only parent class method, but if we want to call same child method with the help of pointer variable then we will define parent class method with virtual keyword . It means virtual function is a function which we will define in parent class and again we will redefine it in child class.

```

#include<iostream>
using namespace std;
class Dummy
{
public:
    virtual void show()
    {
        cout<<"show is running from Dummy"<<endl;
    }
};
class Demo:public Dummy
{
public:
    void show()
    {
        cout<<"show is running from Demo"<<endl;
    }
};
int main()

```

```

{
    //Dummy d;
    //Dummy *p;
    //Demo d;
    //Demo *p;
    Demo d;
    Dummy *p;
    p=&d;
    p->show();
    return 0;
}           I

```

```

#include<iostream>
using namespace std;
class Trainer
{
public:
    void skills()
    {
        cout<<"skills of it"<<endl;
    }
    virtual void intro()
    {
        cout<<"intro of trainer"<<endl;
    }
};
class Student:public Trainer
{
public:
    void intro()
    {
        cout<<"intro of student"<<endl;
    }
};
int main()
{
    Trainer *t;
    Student s;
    t=&s;
    t->intro();
    t->skills();
    return 0;
}

```

Pure Virtual Function

A virtual function that contains declaration=0 is known as Pure Virtual Function , we can declare a pure virtual function with the help of following syntax

```
virtual return_type name()=0;
```

If a class contains at least a pure virtual function we will consider that class as Abstract class, a abstract class is a special class through which we can achieve abstraction in c++, an abstract class always contains a pure virtual function.

"We can't create an object of abstract class but we can create a pointer of abstract class"

If any class will inherit an abstract class then we have to override all the abstract function in child class otherwise child class should also be considered as an abstract

```

#include<iostream>
using namespace std;
class Rbi
{
    public:
        void rules()
        {
            cout<<"rules by Rbi"<<endl;
        }
        virtual void deposit()=0;//pure virtual function
        virtual void withdraw()=0;//pure virtual function
};
class Sbi:public Rbi
{
    public:
        void deposit()
        {
            cout<<"deposit in Sbi"<<endl;
        }
        void withdraw()
        {
            cout<<"withdraw amount from Sbi"<<endl;
        }
};
int main()
{
    //Rbi r;//error
    Sbi s;
    s.rules();
    s.deposit();
    s.withdraw();
    return 0;
}

```

Friend class

Friend class is a class through which we can access private and protected data members and member functions of another class, we can make a class as friend of another with the help of friend keyword.

```

#include<iostream>
using namespace std;
class Dummy
{
    private:
        int a;
    protected:
        int b;
    public:
        void setvalue()
        {
            this->a=10;
            this->b=20;
        }
        void display()
        {

        }
    friend class Demo;
};
class Demo:public Dummy
{
    public:
        void show()
        {

```

```

        cout<<a<<b<<endl;
    }
};

int main()
{
    Demo d;
    d.setValue();
    d.show();
    return 0;
}

```

Friend function

Friend function is a not a member function of class but it is used to access private and protected members of class. We can construct friend function with the help of friend keyword

```

#include<iostream>
using namespace std;
class Demo
{
    private:
        int a;
    protected:
        int b;
    public:
        void setValue()
        {
            this->a=10;
            this->b=20;
        }
        friend void show();
};

void show()
{
    Demo d;
    d.setValue();
    cout<<d.a<<d.b<<endl;
}

int main()
{
    show();
    return 0;
}

```

```

#include<iostream>
using namespace std;
class Demo
{
    private:
        int a;
    protected:
        int b;
    public:
        void setValue()
        {
            this->a=10;
            this->b=20;
        }
        friend int main();
};

int main()
{
    Demo d;
    d.setValue();
    cout<<d.a<<d.b<<endl;
    return 0;
}

```

```

void main()
{
    int puts(const char *);
    puts("hello all");
}

```

Operator overloading as we know C++ is an object oriented programming language that supports of overloading, we can overload function as well as operator in C++. We will use following syntax for operator overloading

```

classname operator operator_name(class const &obj)
{

}

#include<iostream>
using namespace std;
class Demo
{
public:
    int a,b;
    void show()
    {
        cout<<a<<" "<<b<<endl;
    }
    Demo operator +(Demo const &obj)
    {
        Demo t;
        t.a=a+obj.a;
        t.b=b+obj.b;
        return t;
    }
};

int main()
{
    Demo d1;
    d1.a=10;
    d1.b=20;
    cout<<"value in d1 object are a="<<d1.a<<"and b="<<d1.b<<endl;
    Demo d2;
    d2.a=30;
    d2.b=40;
    cout<<"value in d2 object are a="<<d2.a<<"and b="<<d2.b<<endl;
    Demo d=d1+d2;
    cout<<"value in d object are a="<<d.a<<"and b="<<d.b<<endl;
}

```

New Keyword in C++

It is mainly used to allocate dynamic memory it will return address of object and we will store address of object in pointer variable. But the dynamic memory allocated by new keyword will not release automatically, we will use delete keyword to release dynamic memory which is allocated by new keyword

```

#include<iostream>
using namespace std;
class Demo
{
public:
    int a,b;
    Demo(int a,int b)
    {
        this->a=a;
        this->b=b;
    }
    void show()
    {
        cout<<a<<" "<<b<<endl;
    }
};
int main()
{
    //Demo d(10,20);
    //Demo *p;
    //p=&d;
    Demo *p=new Demo(20,30);
    cout<<p->a<<" "<<p->b<<endl;
    delete p;
    cout<<p->a<<" "<<p->b<<endl;
}

```

Template is a concept

At the time of calling we will pass data type ,then function and class will work according to data type, in c++ we will use 2 types of templates

1st:- function template

We will use following syntax for

```

template<typename SYMBOL> SYMBOL function_name()
{
    return statement;
}

```

```

#include<iostream>
using namespace std;
template<typename T> T show(T a,T b)
{
    return a+b;
}
int main()
{
    cout<<show<int>(10,20)<<endl;
    cout<<show<float>(10.2,20.5)<<endl;
    cout<<show<long>(25457,2000)<<endl;
    return 0;
}

```

2nd:- class template

We will use following syntax

```

template<typename symbol> class classname
{
    I
};

};
```

```

#include<iostream>
using namespace std;
template<typename T>class Demo
{
public:
    T a;
    void setValue(T a)
    {
        this->a=a;
    }
    T getValue()
    {
        return a;
    }
}; I
int main()
{
    Demo<int> d;
    d.setValue(10);
    cout<<d.getValue()<<endl;
    Demo<float> d1;
    d1.setValue(12.5);
    cout<<d1.getValue()<<endl;
    return 0;
}

```

C++ file handling

File is placed at secondary storage, which is used to store data permanently we can read and write stream(group of characters or sequence of data) type data

c++ provides stream classes through which we can perform read write and append operations with file

Stream classes like

ofstream use to write and append data in a file
ifstream use to read data from file
fstream use to read/write/append

Open function :-it is used to open a file in following mode

It contains 2 arguments 1st one is file name and 2nd one is opening mode, we will use following opening mode

ios::out write mode
ios::in read mode
ios::app append mode

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
    ofstream fout;
    string data;
    fout.open("abc.txt");
    //fout.open("abc.txt",ios::app);
    while(fout)
    {
        getline(cin,data);
        if(data=="-1")
        {
            break;
        }
        fout<<data<<endl;
    }
    cout<<"data sucessfully write in file"<<endl;
    fout.close();
    ifstream fin;
    fin.open("abc.txt");
    while(fin)
    {
        getline(fin,data);
        cout<<data<<endl;
    }
    fin.close();
    return 0;
}
```

Screen clipping taken: 28-10-2022 03:47 PM

Jsp

14 September 2022

It is a tag support language

1.Scriptlet tag- "if we want to write java code inside a JSP page we will write that code inside scriptlet tag. and this code will put by JSP loader inside service method of servlet."

<% java code %> - syntax.

2.Declaration tag- "if we want to declare a global variable or a method with in a JSP page we will use declaration tag. if we write any code inside declaration tag it will automatically transfer in the class of servlet by JSP loader."

<%! java code %>- syntax.

3.Expression tag- "if we want to print something within JSP page we will use expression tag. it is substitution of out.println in JSP ."

<%= java code %>- syntax.

```
1 <h1>hello</h1>
2 <h2>
3 <%
4 String s="hello";
5 out.println(s);
6 <%
7 </h2>
8 <%! String b="global variable b"; %>
9 <%=b%>
```

Exception handling by jsp By JSP Directives ie @page and @include

```
<%@page errorPage="error.jsp" %>
<%
String n1=request.getParameter("n1");
String n2=request.getParameter("n2");
String ch=request.getParameter("ch");
int a=Integer.parseInt(n1);
int b=Integer.parseInt(n2);
switch(ch)
{
    case "Add":
        out.println("add is "+(a+b));
        break;
    case "Sub":
        out.println("sub is "+(a-b));
        break;
    case "Mul":
        out.println("mul is "+(a*b));
        break;
    case "Div":
        out.println("add is "+(a/b));
}
%>
```

```
1 <%@page isErrorPage="true"%>
2 <h1>something went wrong</h1>
3 <%=exception%>
```

Jsp action elements

```

1<%>
2 String id=request.getParameter("id");
3 String pwd=request.getParameter("pwd");
4 if(id.equals("admin")&&pwd.equals("12345"))
5 {
6     session.setAttribute("id", id);
7     //response.sendRedirect("adminhome.jsp");
8 %>
9 <jsp:forward page="adminhome.jsp"></jsp:forward>
.0<%>
.1 }
.2 else
.3 {
.4     out.println("Invalid id and password");
.5 }
.6 %>

```

To club different html pages and jsp into one page

```

1 <h1>First jsp page is running</h1>
2 <jsp:include page="index.html"></jsp:include>
3 <h1>admin page</h1>
4 <jsp:include page="admin.html"></jsp:include>
5 <h1>servicepro</h1>
6 <jsp:include page="servicepro.html"></jsp:include>

```

Steps in Image API/image about
 We will define stored location of file in web.xml .
 we

```

+-----+
| veer@gmail.com | ajay@gmail.com | 2022-09-23T20:30 | pending |
| veer@gmail.com | ajay@gmail.com | 2022-09-26T22:50 | pending |
+-----+
2 rows in set (0.00 sec)

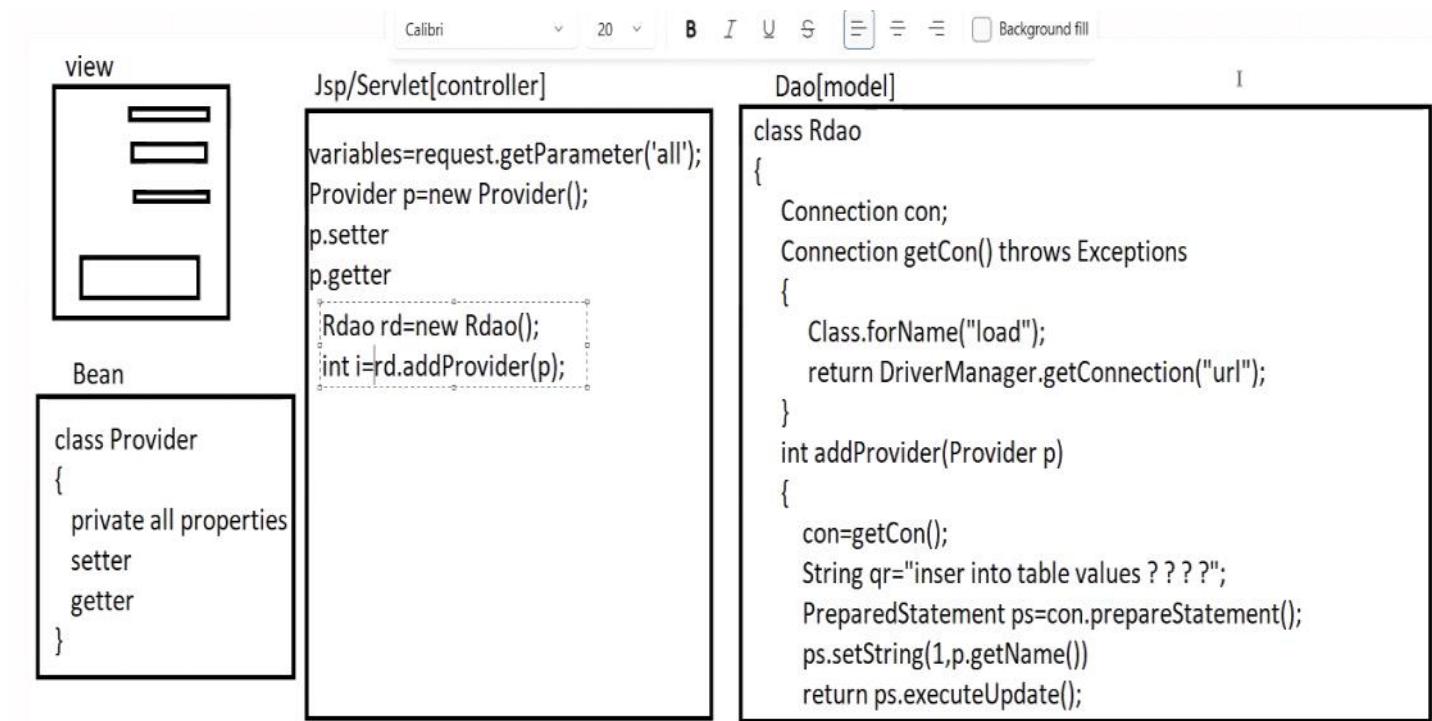
mysql> select s.name,s.email,mob,charges,u.name,uemail,dt,bstatus from serviceprovider s,user u,booking where booking.pemail=s.email and booking.uemail=u.email;
+-----+
| name | email | mob | charges | name | uemail | dt | bstatus |
+-----+
| veer | veer@gmail.com | 7878989845 | 350 | ajay | ajay@gmail.com | 2022-09-23T20:30 | pending |
| veer | veer@gmail.com | 7878989845 | 350 | ajay | ajay@gmail.com | 2022-09-26T22:50 | pending |
+-----+
2 rows in set (0.01 sec)

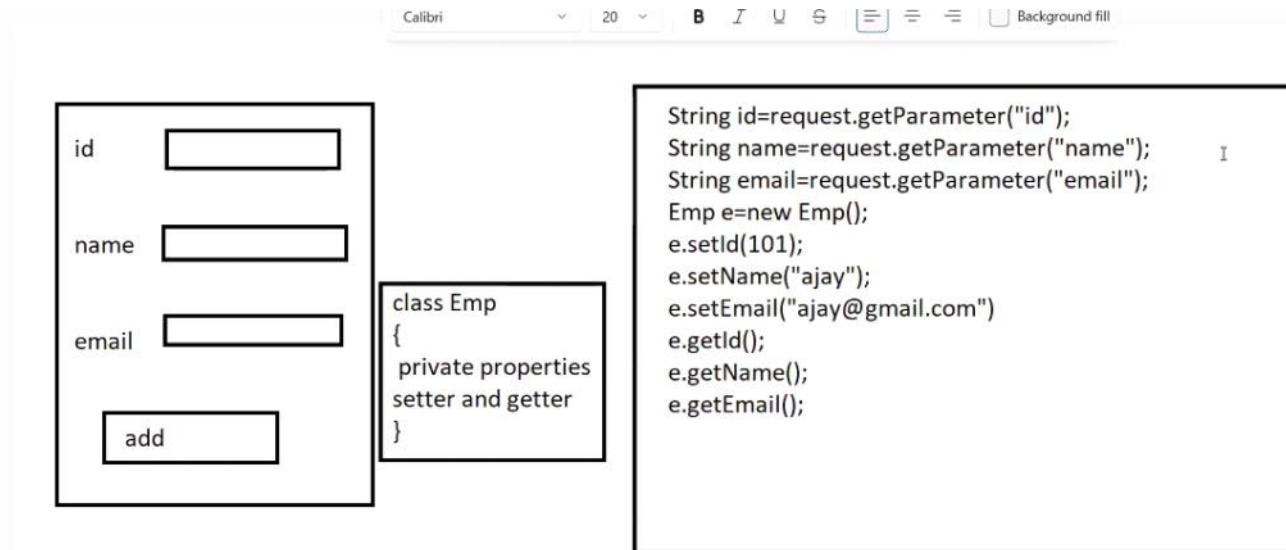
mysql> select s.name,s.email,mob,charges,u.name,uemail,dt,bstatus from serviceprovider s,user u,booking where pemail=u.email and uemail=u.email;
+-----+
| name | email | mob | charges | name | uemail | dt | bstatus |
+-----+
| veer | veer@gmail.com | 7878989845 | 350 | ajay | ajay@gmail.com | 2022-09-23T20:30 | pending |
| veer | veer@gmail.com | 7878989845 | 350 | ajay | ajay@gmail.com | 2022-09-26T22:50 | pending |
+-----+
2 rows in set (0.00 sec)

mysql>

```

Java Beans are classes that contains private properties that contains setter and getter method.





Web.xml is one of the most important file of our project through which we will control all functionalities of our project, with the help of web.xml we can define welcome file of our project and we can also map a servlet using web.xml
Webcontent/web inf/ web.xml

Web.xml is also used to map a servlet using xml tags, in java servlet is also a tag and we can use a servlet mapping tag to map a servlet

```

<welcome-file-list>
    <welcome-file>first.html</welcome-file>
</welcome-file-list>
<servlet>
    <servlet-name>one</servlet-name>
    <servlet-class>Demo</servlet-class>
</servlet>
<servlet-mapping>
    <servlet-name>one</servlet-name>
    <url-pattern>/first</url-pattern>
</servlet-mapping>

```

Live Project Heroku
`mysql://b23c435ce3e477:lad1f069@us-cdbr-east-06.cleardb.net/heroku_c7c89a5171d1d0a?reconnect=true`
`mysql://`
`b23c435ce3e477 username`
`:`
`lad1f069 password`
`@`
`us-cdbr-east-06.cleardb.net host`
`/`
`heroku_c7c89a5171d1d0a database`
`?reconnect=true`

Screen clipping taken: 19-10-2022 03:50 PM

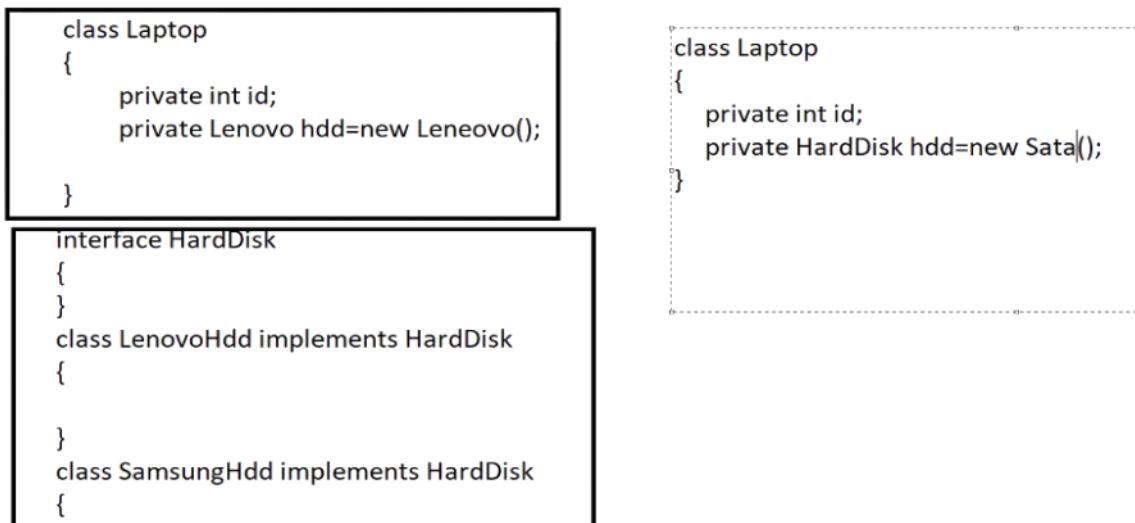
MVC (Model View Controller)

Tight coupling-When a class object is dependent only on one another class object

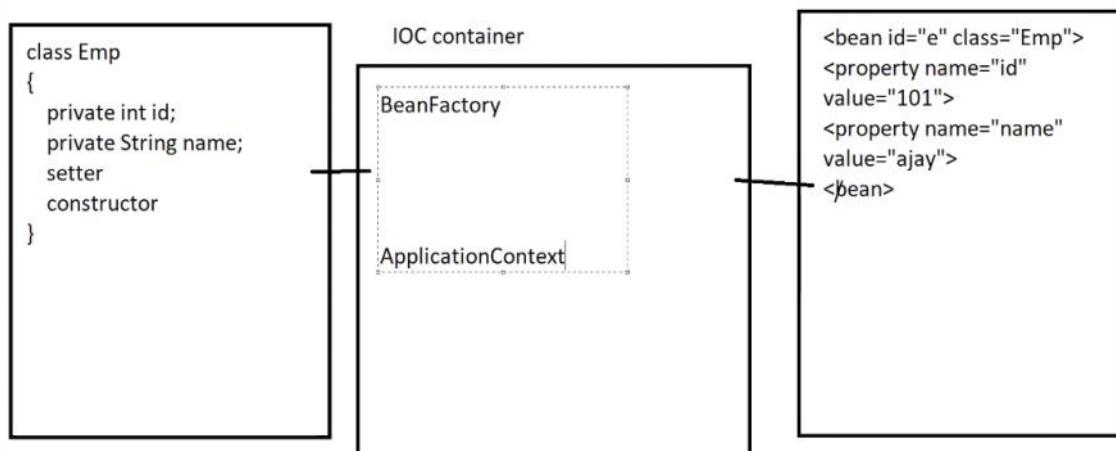
&

Loose coupling-When a class object doesn't depend on single class it has many classes that ie like an interface, different classes can implements that interface and interdepend ability is removed.

Spring uses loose coupling



Inversion of control (IOC container)



Spring

Spring is framework of framework in which we will use some frameworks like SpringCore, SpringJdbc, SpringBoot, SpringMvc, SpringSecurity, SpringMvc etc.

In Spring Core we will construct a bean object without using new keyword, with the help of xml and we will configure xml and bean with the help of **IOC (inversion of control)** we will use 2 IOC container to construct being using xml

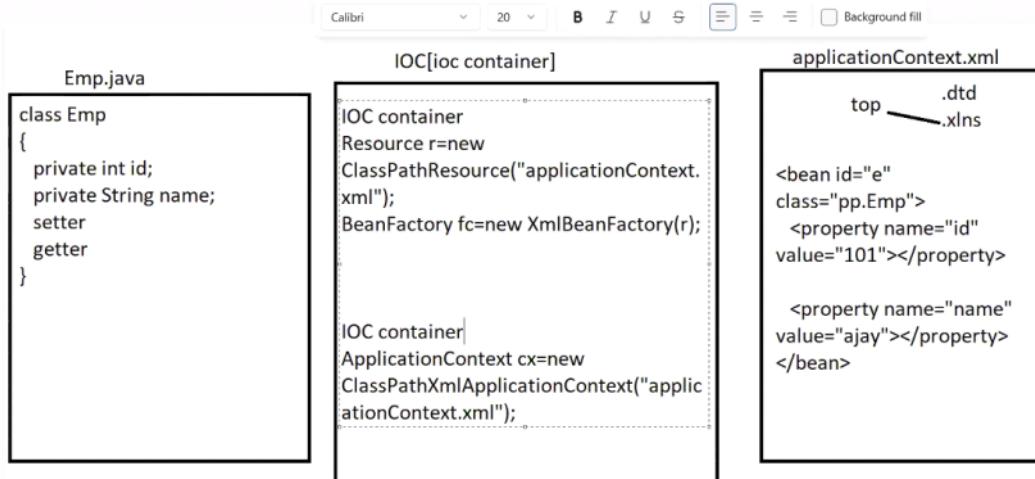
1st **BeanFactory** :- BeanFactory IOC container is used to construct a Bean object using Xml , we will object of Resource Interface first which is stored in implementation class that is ClassPathResource that contains Xml File name as a String in arguments ,After this we will construct XmlBeanFactory class object that contains Resource object in constructor argument that is an implementation class of BeanFactory

```
IOC container
Resource r=new
ClassPathResource("applicationContext.
xml");
BeanFactory fc=new XmlBeanFactory(r);
```

2nd ApplicationContext:-It is another way to construct IOC container we will store implementation class object of

ClassPathXmlApplicationContext which contains xml file name as arguments as String

```
IOC container  
ApplicationContext cx=new  
ClassPathXmlApplicationContext("applic  
ationContext.xml");
```



Dependency Injection

Dependency injection is used to inject a bean object with xml using IOC container, we can use 2 types of dependency injection in spring

1st:-

Constructor injection:- constructor injection is mainly used to set bean class properties using constructor, in xml we will use `<constructor-arg>` as a tag of xml to inject a bean using IOC container

a)constructor injection with primitives

b)constructor injection with the dependent bean class object

Collection Object with

Auto Wiring :-

Auto wiring is a concept of spring core through which we will automatically inject ,two different dependent bean objects we can't use auto wiring with primitives and string we can use auto wiring with dependent class object only .In spring we will use following techniques

1st By name auto wiring mode: by name auto wiring mode is used to inject two dependent bean objects by their names , with the help of this auto wiring we can inject one or more than one objects

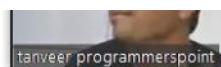
```
1 <?xml version="1.0" encoding="UTF-8"?>  
2<beans  
3   xmlns="http://www.springframework.org/schema/beans"  
4   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
5   xmlns:p="http://www.springframework.org/schema/p"  
6   xsi:schemaLocation="http://www.springframework.org/schema/beans  
7           http://www.springframework.org/schema/beans/spring-beans-3.0.xsd">  
8 <bean id="address" class="pp.Address"></bean>  
9 <bean id="e" class="pp.Emp" autowire="byName"></bean>  
10 </beans>
```

in case of `byName` autowiring mode property name and bean id should be same, but in case of `by type` autowiring mode property name and bean name could be different

2nd By type auto wiring mode: by type auto wiring mode we can inject only one bean object according to type

3rd Default auto wiring mode :in default no auto wiring by default

4th Constructor auto wiring mode: constructor auto wiring mode is used to inject beans using constructor



Singleton is a design pattern through which we can construct only single object of a class its constructor always should be private and we will construct object inside this class and store inside a static final reference variable, and a static method will return this object

```
class Dummy
{
    private static final Dummy d=new Dummy();
    private Dummy()
    {

    }
    public static Dummy getObj()
    {
        return d;
    }
    public void show()
    {
        System.out.println("show is running");
    }
}
class Demo
{
    public static void main(String gg[])
    {
        //Dummy d=new Dummy();
        Dummy d=Dummy.getObj();
        d.show();
    }
}
```

Spring jdbc

JDBC template class is used to perform operations using jdbc it provide some methods to perform operations with database like dml ,ddl ,dcl ,dtl etc. Some commonly used methods of jdbc template classes are

- 1) `public int update(String query)` is used to insert, update and delete records.
- 2) `public int update(String query, Object... args)` is used to insert, update and delete records using PreparedStatement using given arguments.
- 3) `public void execute(String query)` is used to execute DDL query.
- 4) `public T execute(String sql, PreparedStatementCallback action)` executes the query by using PreparedStatement callback.
- 5) `public T query(String sql, ResultSetExtractor rse)` is used to fetch records using ResultSetExtractor.
- 6) `public List query(String sql, RowMapper rse)` is used to fetch records using RowMapper.

DAO is **an interface** we provide over the base persistence mechanism. In computer software, a data access object (DAO) is a pattern that provides an abstract interface to some type of database or other persistence mechanism.

11. DAO support - Spring

Search for: [What is DAO class in Spring?](#)

Why do we write DAO class in Java?

WHY WE USE DAO: **To abstract the retrieval of data from a data resource such as a database.** The concept is to "separate a data resource's client interface from its data access mechanism." The problem with accessing data directly is that the source of the data can change. 03-Oct-2013

ResultSetExtractor is an interface which is used to execute select query using JdbcTemplate , when we will call query method of JdbcTemplate class then it will return implementation class object of ResultSetExtractor interface, we will override its extractData method and we will return list of objects from extractData methods

Anonymous class

```
interface Dummy
{
    void show();
}
class Demo
{
    public static void main(String gg[])
    {
        Dummy d=new Dummy(){
            public void show()
            {
                System.out.println("show is running");
            }
        };
        d.show();
    }
}
```

RowMapper interface

When we will execute query method of jdbc template class it contains 2 arguments 1st one is a query as String and 2nd one is implementation class object of RowMapper we will override mapRow method of RowMapper interface through which we will return objects in form of List, it means there is no need to manually construct a list here(ArrayList as in ResultSetExtractor) here

```
public List<Emp> show(){
    return template.query("select * from emp", new ResultSetExtractor <List>(){

        @Override
        public List<Emp> extractData(ResultSet rs) throws SQLException, DataAccessException {
            ArrayList<Emp> al=new ArrayList<Emp>();
            while(rs.next()){
                Emp e=new Emp();
                e.setId(rs.getInt("id"));
                e.setName(rs.getString("name"));
                al.add(e);
            }
            return al;
        }
    });
}

public List<Emp> showData(Emp e){
    return template.query("select * from emp", new RowMapper <Emp>() {

        @Override
        public Emp mapRow(ResultSet rs, int arg1) throws SQLException {
            Emp e=new Emp();
            e.setId(rs.getInt("id"));
            e.setName(rs.getString("name"));
            return e;
        }
    });
}
```

Screen clipping taken: 09-11-2022 09:09 PM

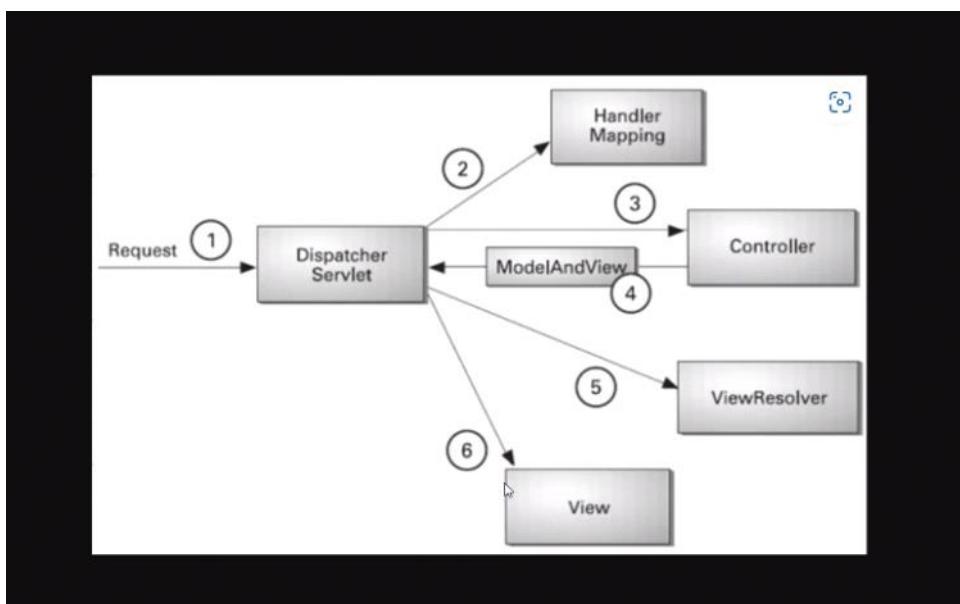
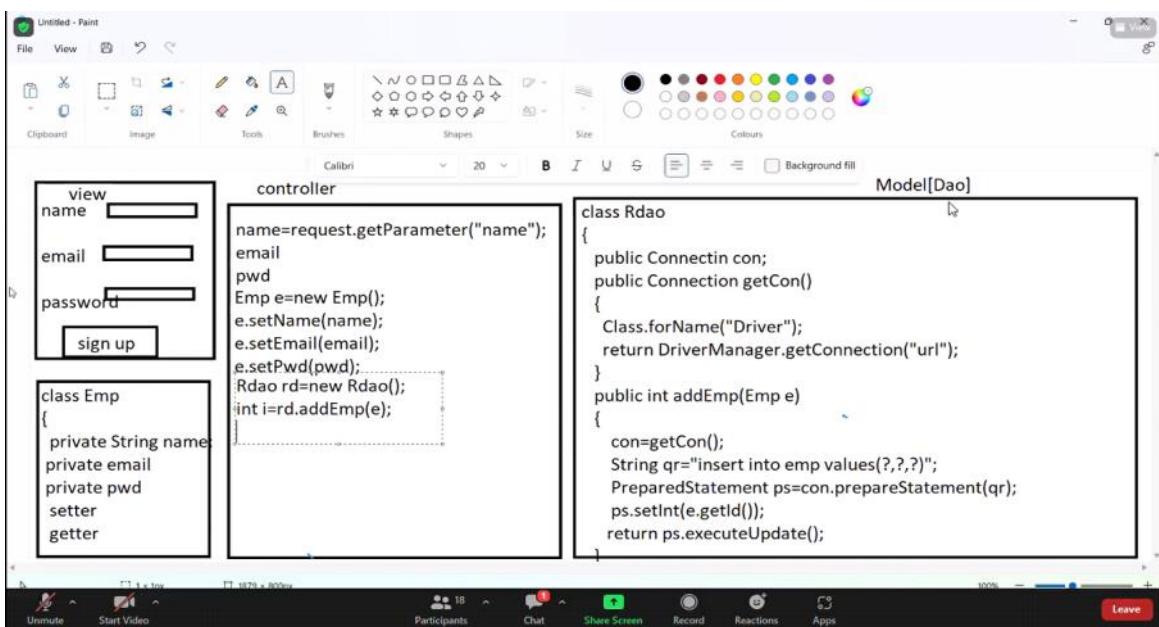
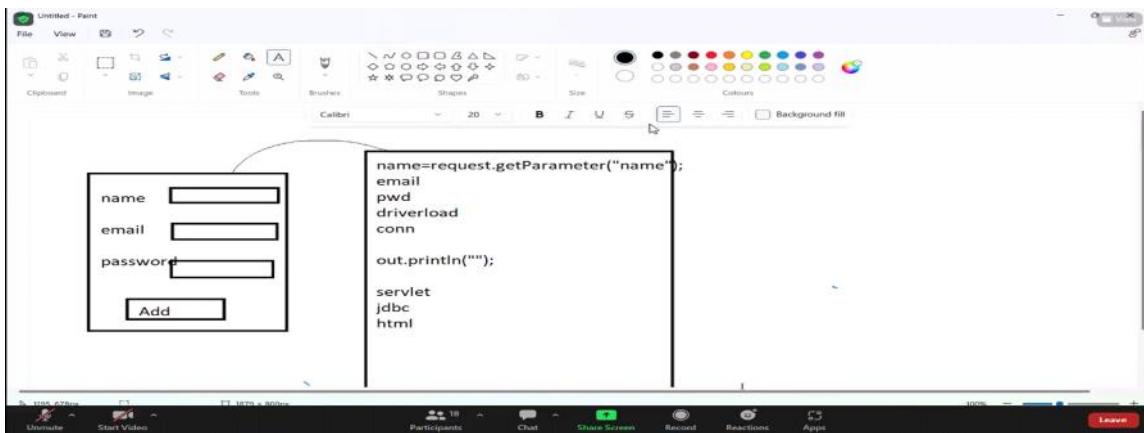
Hibernate

PreparesStatementCallback interface is mainly used to execute parametrized query on data base, we can use PreparesStatementCallback with any jdbcTemplate class method like update, query so when we will construct implementation class object of PreparesStatementCallback then we will override doInPreparedStatement method

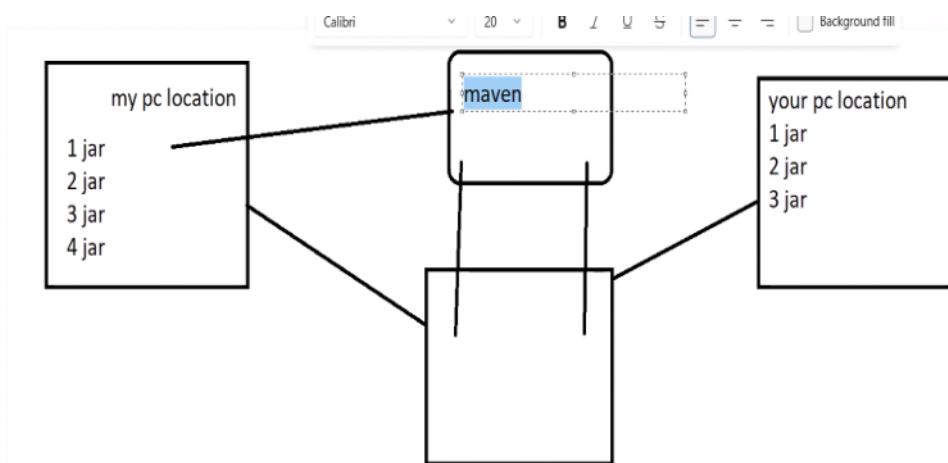
NamedParameterJdbcTemplate is mainly used to execute parameterized query on data base, in which we will use : (colon)& parameter name when we want to put values using NamedParameterJdbcTemplate class

SimpleJdbcTemplate

Normal Model View Controller



Maven dependency



We will use ModelAndView class to transfer data from controller to view. Constructor of ModelAndView class contains 3 arguments

1sr-view name, 2nd variable name, 3rd value of a variable

Model Interface is used to transfer values from controller to view, we will use its addAttribute() method that contains 2 arguments
1st name of variable and second one is its value

JSTL

If

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <c:if test="${nm == 'ppoints'}">
3 <h1>welcome to page</h1>
4 </c:if>

```

Switch case

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <c:choose>
3 <c:when test="${nm=='ppoint'}">
4 <h1>ppoint</h1>
5 </c:when>
6 <c:when test="${nm=='ppoints'}">
7 <h1>ppoints</h1>
8 </c:when>
9 <c:otherwise>
10 <h1>not matched case</h1>
11 </c:otherwise>
12 </c:choose>

```

For each of ArrayList

```

1 package pp;
2
3 import java.util.ArrayList;
8
9 @Controller
10 public class MyController {
11     @RequestMapping("/second")
12     public String second(Model m)
13     {
14         ArrayList<String> al=new ArrayList<String>();
15         al.add("first");
16         al.add("second");
17         al.add("third");
18         m.addAttribute("nm", al);
19         return "second";
20     }
21 }
22

```

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <c:forEach var="i" items="${nm}">
3 <c:out value="${i}"></c:out><br>
4 </c:forEach>

```

To make variable

```

1 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
2 <c:set var="i" scope="session" value="${1000}"></c:set>
3 <c:out value="${i}"></c:out>

```

The two major errors that can come in springMVC And JSTL are in my project that are

Hibernate template class is mainly used to implement hibernate with spring, it provides an advantage there is no need of manual configuration and will always construct table automatically. There is no need to put/write SQL queries
Commonly used methods of hibernate template class are

No.	Method	Description
1)	void persist(Object entity)	persists the given object.
2)	Serializable save(Object entity)	persists the given object and returns id.
3)	void saveOrUpdate(Object entity)	persists or updates the given object. If id is found, it updates the record otherwise saves the record.
4)	void update(Object entity)	updates the given object.
5)	void delete(Object entity)	deletes the given object on the basis of id.
6)	object get(Class entityClass, Serializable id)	returns the persistent object on the basis of given id.
7)	object load(Class entityClass, Serializable id)	returns the persistent object on the basis of given id.
8)	List loadAll(Class entityClass)	returns the all the persistent objects.

Spring boot is used to construct robust secure and easy to test application, it provides some annotations through which we will perform some important operations with our program

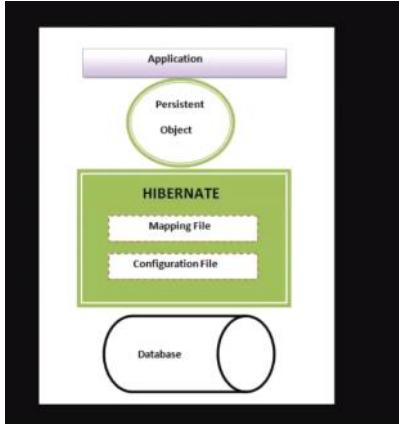
The major advantage of Spring Boot is it provide auto configuration , it means it will automatically configure all the configurations and there is no need for manual configuration . It means we want to focus only on coding we don't have to focus on configuration

When we will construct any spring boot application than we will use spring boot annotation and if we have to construct object of any bean class of our application than

always object will be provided by spring boot container but spring container will return object when we will mark our class by @component annotation.
But spring container always construct object in singleton scope but if we want to construct object with prototype scope then we will use @scope annotation and different object will be constructed
When we will use Autowire annotation then it will use by type autowiring mode, if you want to use

Hibernate

Hibernate is an ORM tool object relational mapping tool, or it's a database framework which is used mainly used to construct database independent application
Hibernate uses hql(hibernate query language) instead of sql , sql contains all those common queries which is same for all databases



Different between save and persist:-persist method will return void value while save method will return a primary key.
get and load method of Session interface:-when we will call get method it will directly hit database and search object corresponding to given id , if object is found then it will return object and if object is not found it will return null(NullPointerException), it contains 2 arguments 1st one is class and second one is primary key value.

```
1 import org.hibernate.Session;
2 import org.hibernate.cfg.Configuration;
3
4 import pp.Emp;
5
6 public class SearchByLoad {
7
8     public static void main(String[] args) {
9         Session session=new Configuration().configure().buildSessionFactory().openSession();
10        Emp e=session.load(Emp.class, new Integer(105));
11        System.out.println(e.getId()+" "+e.getName());
12        session.close();
13    }
14 }
15 }
```

When we will call load method at session object than it will not hit database directly, it will internally create fake object and return this fake object when we will call getter methods then load will hit database and search object corresponding to given id, if object is found than it will replace fake object with original object, but if object is not found it will throw ObjectNotFoundException.

Hql hibernate query language

It is used to execute all the common quires on database we will use Query interface to implement hql in hibernate, when we will call createQuery method at session reference variable it will return implementation class object of query interface. Commonly used methods of Query interface are

public int executeUpdate() is used to execute the update or delete query.
public List list() returns the result of the relation as a list.
public Query setFirstResult(int rowno) specifies the row number from where record will be retrieved.
public Query setMaxResult(int rowno) specifies the no. of records to be retrieved from the relation (table).
public Query setParameter(int position, Object value) it sets the value to the JDBC style query parameter.
public Query setParameter(String name, Object value) it sets the value to a named query parameter.

```

1 import java.util.List;
8
9 public class Show {
10
11     public static void main(String[] args) {
12         Session session=new Configuration().configure().buildSessionFactory().openSession()
13         //Query qr=session.createQuery("from Emp");
14         //qr.setFirstResult(0);
15         //qr.setMaxResults(3);
16         Query qr=session.createQuery("from Emp where id=:id");
17         qr.setParameter("id",new Integer(101));
18         List<Emp> al=qr.list();
19         for(Emp i:al)
20         {
21             System.out.println(i.getId()+" "+i.getName()+" "+i.getSal());
22         }
23         session.close();
24     }
25 }
26

```

State of hibernate object

There are 3 stage of hibernate object,

1st Bean object(Transient state):-a bean class object should be considered as bean object at the time of its creation

```

Emp e=new Emp();
Scanner sc=new Scanner(System.in);
System.out.println("enter id");
e.setId(sc.nextInt());
System.out.println("enter name");
e.setName(sc.next());
System.out.println("enter sal");
e.setSal(sc.nextInt());

```

2nd Persistent object/state:-A object should be considered as persistent object when it will take entry in session

```
int id=(int) session.save(e);
```

3rd:Detached state(removed state):-After connection close a persistent object will convert into detached object

```
session.close();
```

TABLE PER HIERARCHY:-

In this type of hierarchy is the major drawback of this hierarchy is that it will store null values in rows, we will use some extra columns (discriminator column) to identify discriminating values

TABLE PER CONCRETE CLASS:- IN THIS TYPE OF inheritance strategy table according to classes, but the major drawback of this strategy is there is no relationship between the table

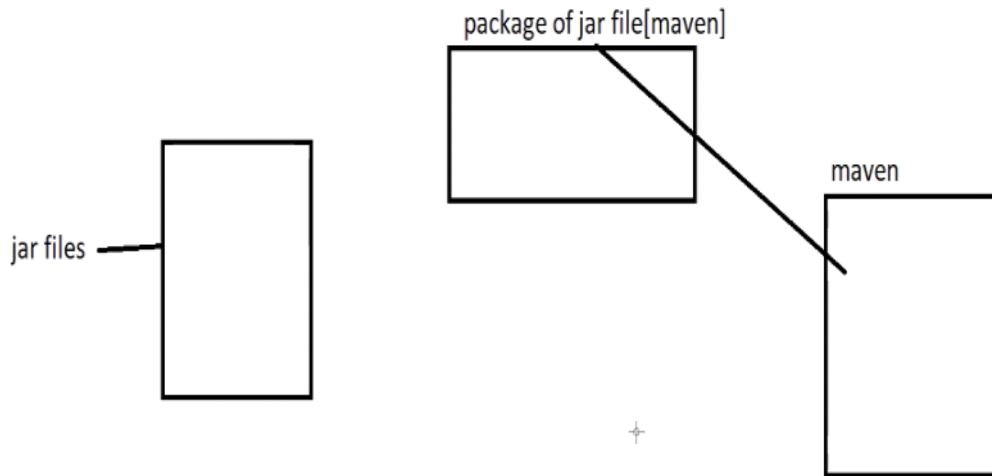
TABLE PER SUBCLASS:-In this type of inheritance strategy table created according to classes and the major advantage of this strategy is that it will establish relationship between the tables.

Criteria API:-Criteria Api is mainly used to call aggregate function in hibernate and it is also used to apply projection in hibernate, but in hibernate Criteria is an interface and when we will call createCriteria() on session reference variable it will return implementation class object of criteria interface

Maven

14 October 2022 06:14 PM

Maven is a collection of jar files or a package through which we can construct configuration ready application it means maven mainly used to configure everything in our project automatically and by using JARS file we will configure all configuration manually.



When we will construct a maven project than we will get pom.xml file and with the help of pom.xml file we will set all the dependencies of project with the help of xml tags

MongoDB

Mongo Db local host port number is 271017

Steps to set path in MONGO DB

As we know mongo dB stored location is

C:\Program Files\MongoDB\Server\5.0\bin

Right click on this pc-->Properties-->View advanced system setting-->Environment variables->path->edit->new->paste->ok

BASIC STEPS TO EXECUTE MONGO DB

go to Command prompt

Mongo-> press enter

List of database can be seen by command

show dbs

Show name of current working database

db->Enter

By default test is the database for mongodb

Create or use database command is

use database_name

We can drop the database with the help of

Db.dropDatabase()

List of documents with

Show collections

Create a new collection with the help of following command

db.createCollection('student')

Drop a collection with the help of following commands

db.collection_name.drop()

If we want to add some fields in collection then we have two options

Either we can insert one or we can insert many

```
db.student.insertOne({  
  'roll':102,  
  'name':'abhi',  
  'branch':'it',  
  'address':'indore',  
  'marks':[88,78,79,96,89]  
})|
```

```

db.student.insertMany([
{
  'roll':103,
  'name':'jay',
  'branch':'ec',
  'address':'ujjain',
  'marks':[88,78,79,96,89]
},
{
  'roll':104,
  'name':'abhishek',
  'branch':'ec',
  'address':'indore',
  'marks':[98,97,99,96,99]
},
{
  'roll':105,
  'name':'dev',
  'branch':'it',
  'address':'bhopal',
  'marks':[68,68,59,46,59]
}
])

```

We can show or view all fields or record with the help of

db.student.find()

db.student.find().pretty()

We can count number of records in a collection with the help of following commands

db.student.find().count()

We can display or show limited records with the help of following limit function

db.student.find().limit(3)

Show specific records

As we know we can show all records with the help of find function but if we want to apply some filtrations then with find function then we will use following syntax/command

db.student.find({'roll':101,'branch':'cs'})

```

> db.student.find({'roll':101})
{ "_id" : ObjectId("638f2731cd47682926e17bee"), "roll" : 101, "name" : "ajay", "branch" : "cs", "address" : "indore",
  "marks" : [ 78, 98, 89, 86, 87 ] }
> db.student.find({'address':'indore'})
{ "_id" : ObjectId("638f2731cd47682926e17bee"), "roll" : 101, "name" : "ajay", "branch" : "cs", "address" : "indore",
  "marks" : [ 78, 98, 89, 86, 87 ] }
{ "_id" : ObjectId("638f2769cd47682926e17bef"), "roll" : 102, "name" : "abhi", "branch" : "it", "address" : "indore",
  "marks" : [ 88, 78, 79, 96, 89 ] }
{ "_id" : ObjectId("638f28d3cd47682926e17bf1"), "roll" : 104, "name" : "abhishek", "branch" : "ec", "address" : "indore",
  "marks" : [ 98, 97, 99, 96, 99 ] }
> db.student.find({'address':'indore','branch':'cs'})
{ "_id" : ObjectId("638f2731cd47682926e17bee"), "roll" : 101, "name" : "ajay", "branch" : "cs", "address" : "indore",
  "marks" : [ 78, 98, 89, 86, 87 ] }

```

To search only specific top data

```
> db.student.findOne({'address':'indore'})  
{  
    "_id" : ObjectId("638f2731cd47682926e17bee"),  
    "roll" : 101,  
    "name" : "ajay",  
    "branch" : "cs",  
    "address" : "indore",  
    "marks" : [  
        78,  
        98,  
        89,  
        86,  
        87  
    ]  
}  
> |
```

Projections

To show only the specific data of collection ie specific column

The first curly bracket of find() function should be used to apply filtrations, and the second curly bracket is used to apply projections

```
db.student.find({},{'_id':0,'roll':1,'name':1})
```

```
> db.student.find({},{'_id':0,'roll':1,'name':1,'branch':1})
```

```
> db.student.find({},{'_id':0,'roll':1,'name':1,'branch':1,'address':1})
```

Find() function with conditions

ie and /or conditions

```
> db.student.find({'$and':[{'address':'indore'},{'branch':'it'}]},{'_id':0,'roll':1,'name':1,'branch':1,'address':1})
```

```
> db.student.find({'$or':[{'address':'indore'},{'branch':'it'}]},{'_id':0,'roll':1,'name':1,'branch':1,'address':1})
```

Greater than less than commands with equals to commands

```
> db.student.find({'roll':{'$gt':103}})
```

```
> db.student.find({'roll':{'$gte':103}})
```

```
> db.student.find({'roll':{'$lt':103}})
```

```
> db.student.find({'branch':'ec','$or':[{'address':'indore'},{'roll':{'$gt':103}}]})
```

Not equal

```
> db.student.find({'branch':{'$ne':'it'}})
```

IN

```
> db.student.find({'branch':{'$in':['it','cs']}})
```

NOTIN

```
> db.student.find({'branch':{'$nin':['it','cs']}})
```

```
db.student.find({'roll':{'$lte':106}, 'roll':{'$gte':103}, 'address':{'$in':['indore','ujjain']}}, {'_id':0})
```

Update

We can use update with multiple operators in mongo db to update a existing record

```
> db.student.update({'roll':106}, {'$set':{'address':'dewas'}})
```

Upsert means if it is present than it will update it and if not it will create a new entry

```
db.student.update({'roll':107}, {'$set':{'name':'divyansh', 'branch':'cs', 'address':'ujjain'}}, {'upsert':true})
```

```
db.student.update({}, {'$set':{'fees':2500}})
```

```
db.student.updateMany({}, {'$set':{'fees':2500}})
```

To increment add all in entries

```
db.student.updateMany({}, {'$inc':{'fees':200}})
```

To decrement add all in entries

```
db.student.updateMany({}, {'$inc':{'fees':-200}})
```

```
db.student.update({'roll':108}, {'$unset':{'address':'bhopal'}})
```

+1 for ascending order and -1 for descending

```
db.student.find().sort({'fees':-1})
```

Limit only 5 records

```
db.student.find({},{_id:0}).sort({'roll':1}).limit(5)
```

Limit and skip 3 records

```
db.student.find({},{_id:0}).sort({'roll':1}).limit(5).skip(3)
```

```
db.pp.deleteOne({'roll':103})
```

```
db.pp.deleteMany({})
```

```
db.pp.deleteMany({'roll':{'$gte':101}})
```

```
db.pp.insertMany([{'roll':101,'name':'aaa'},{'roll':102,'name':'abhi'},{'roll':103,'name':'yash'}])
```

It can also delete the fields

```
db.pp.remove({'roll':103})
```

Aggregate is mainly used to perform aggregation through which we can find out minimum, maximum, average ,sum, first, last, aggregate data in query

```
> db.student.aggregate([{$group:{_id:'$branch'}}])
```

```
> db.student.aggregate([{$group:{_id:'$branch'}}])
{ "_id" : "ec" }
{ "_id" : "it" }
{ "_id" : null }
{ "_id" : "cs" }
> db.student.aggregate([{$group:{_id:null,sum_fee:{$sum:'$fees'}}}])
{ "_id" : null, "sum_fee" : 20300 }
> db.student.aggregate([{$group:{_id:null,sum_fee:{$max:'$fees'}}}])
{ "_id" : null, "sum_fee" : 3500 }
> db.student.aggregate([{$group:{_id:null,sum_fee:{$min:'$fees'}}}])
{ "_id" : null, "sum_fee" : 1500 }
> db.student.aggregate([{$group:{_id:null,sum_fee:{$avg:'$fees'}}}])
{ "_id" : null, "sum_fee" : 2537.5 }
```

```
> db.student.aggregate([{$group:{_id:'$branch',sum_fee:{$avg:'$fees'}}}])
{ "_id" : "it", "sum_fee" : 2750 }
{ "_id" : null, "sum_fee" : 2150 }
{ "_id" : "ec", "sum_fee" : 2500 }
{ "_id" : "cs", "sum_fee" : 2750 }
> db.student.aggregate([{$group:{_id:'$address',sum_fee:{$max:'$fees'}}}])
{ "_id" : "indore", "sum_fee" : 3500 }
{ "_id" : "ujjain", "sum_fee" : 2500 }
{ "_id" : "bhopal", "sum_fee" : 2500 }
{ "_id" : null, "sum_fee" : 1500 }
{ "_id" : "dewas", "sum_fee" : 2800 }
```

To pop last item of marks array

```
> db.student.updateOne({'roll':101},{'$pop':{'marks':1}})
```

To pop first item of marks array

```
> db.student.updateOne({'roll':101},{'$pop':{'marks':-1}})
```

To push item of marks array(it will push only in last)

```
> db.student.updateOne({'roll':101},{'$push':{'marks':95}})
```

To remove two items from array at a time

```
> db.student.updateOne({'roll':101},{'$pull':{'marks':{'$in':[95,98]}}})
```

Mongo Db indexes :- Mongo Db indexes are mainly used to increase the accessibility

WHEN WE WILL USE INDEXES WITH COLLECTION THEAN IT INTERNALLY USE BINARY SEARCH TREE TO STORE THE DATA

By default it is applied on _id

```
db.student.createIndex()
db.student.createIndex({'roll':1})

    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "createdCollectionAutomatically" : false,
    "ok" : 1
```

```
> db.student.createIndex({'roll':1})
{
    "numIndexesBefore" : 1,
    "numIndexesAfter" : 2,
    "createdCollectionAutomatically" : false,
    "ok" : 1
}
> db.student.getIndexes()
[
    {
        "v" : 2,
        "key" : {
            "_id" : 1
        },
        "name" : "_id_"
    },
    {
        "v" : 2,
        "key" : {
            "roll" : 1
        },
        "name" : "roll_1"
    }
]
> db.student.dropIndex({'roll':1})
{ "nIndexesWas" : 2, "ok" : 1 }
>
```

```

db.createCollection('emp',{
  validator:{
    $jsonSchema:{
      'bsonType':'object',
      'required':['eno','ename'],
      'properties':{
        'eno':{
          'bsonType':'number',
          'description':'eno required integer type'
        },
        'ename':{
          'bsonType':'string',
          'description':'ename required string type'
        },
        'job':{
          'bsonType':'string',
          'description':'job optional string type'
        },
        'address':{
          'bsonType':['string'],
          'description':'address optional string type'
        },
        'hiredate':{
          'bsonType':'date',
          'description':'hiredate optional date type'
        }
      }
    }
  }
})|

```

Description of collection

```

> Object.keys(db.student.findOne())
[ "_id", "roll", "name", "branch", "address", "marks", "fees" ]

```

Atlas

Atlas is tool through which we can construct Mongo Db database on cloud, it provide some tool through which we can directly construct some clusters and cluster contains

Our collection and , and we can use this database into multiple applications

Angular Notes

12 December 2023 01:51 PM

To install angular cli in VsCode :- npm -g install @angular/cli

To create a new app:- ng new Project_Name

To run angular app:- ng serve

To open directly on server:-ng serve --open

Angular is Component based Architecture

To generate a new component:-ng generate component/g c component_name

To use Angular Material in your project

ng add @angular/material

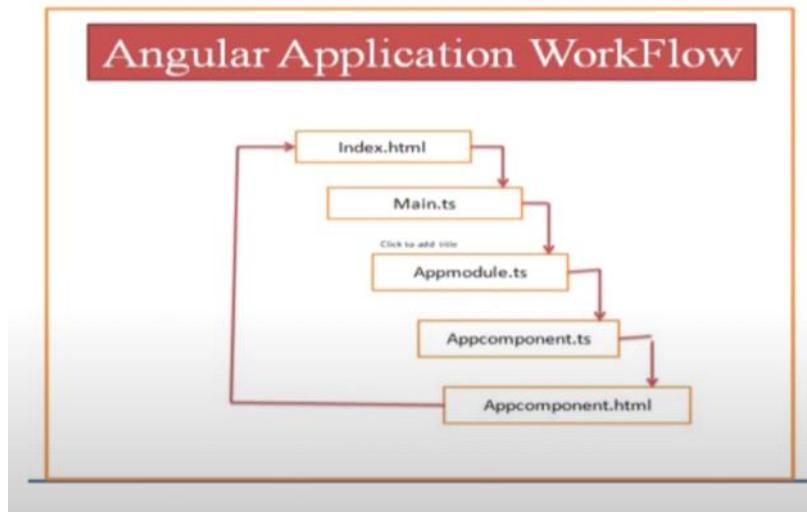
Package.json and package-lock.json files contains references of the npm packages that has been installed in the project during creation or the other packages that will be installed

"^10.6.5":-10 major,6 minor,5 is patch

"^10.6.5":-here ^ carrot symbol represents to check only minor and patch part not the higher versions.

"~0.6.0":-here ~ tilde symbol represents to only check patches available online.

Boot flow of angular project



Modules in angular:-

Module in Angular refers to a place where you can group the components, directives, pipes, and services, which are related to the application.

In case you are developing a website, the header, footer, left, center and the right section become part of a module.

**To get the app.module.ts file we have to use the command `ng new my_project --no-standalone` to

generate the app module.

If we want to add any feature module in our app module then we will use the command
Ng generate module/g m student --module=app

To create a component inside a module we will use ng g c module_name/component_name and the component will automatically add to the description of the module

Decorators:-

- **Decorators** are a feature of TypeScript and are implemented as functions. The name of the decorator starts with @ symbol following by brackets and arguments, since decorators are just functions in TypeScript.
 - Decorators are simply functions that return functions. These functions supply metadata to Angular about a particular class, property, value, or method...
 - **Decorators are invoked at runtime.**
 - Decorators allow you to execute functions. For example @Component executes the Component function imported from Angular 16.
-
- @NgModule() to define modules...
 - @Component() to define components...
 - @Injectable() to define services...
 - @Input() and @Output() to define properties...that send and receive data from the dom.

There are many built-in decorators available in Angular...and many properties on each decorator

Types of decorator

- **Class decorators**, e.g. @Component and @NgModule
- **Property decorators** for properties inside classes, e.g. @Input and @Output
- **Method decorators** for methods inside classes, e.g. @HostListener
- **Parameter decorators** for parameters inside class constructors, e.g. @Inject

Each decorator has a unique role

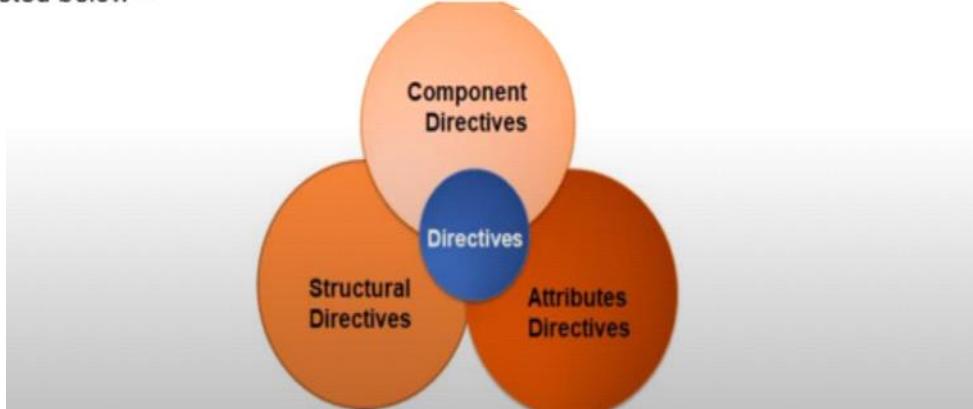
Use preserveWhiteSpaces meta data in component to add spaces between buttons or html elements without css if the spaces are provided in html file. It takes boolean

Understanding templates

In Angular, a template is a blueprint for a fragment of a user interface (UI). Templates are written in HTML, and special syntax can be used within a template to build on many of Angular's features.

Basically, directives are used to extend the power of the HTML attributes and to change the appearance or behavior of a DOM element.

Directive in Angular is a javascript class, which is declared as @directive. Angular has 3 types of directives, which are listed below –



What is ANGULAR - COMPONENTS

The component is a combination of data html template and logics

The component represent an area of a view that shows inside the browser

The component are loads inside the browser.

Data Binding in angular
It is method to use our typescript/javascript variables inside the html components

In html component we can use any typescript and javascript code by following syntax

{{ your code}} this is also called string interpolation in angular

SHARE DATA BETWEEN COMPONENTS

1. Parent to child Component via **@input** Decorator

2. Child to Parent Component Via **@Output** Decorator

3. Child to Parent when there is Event, using the **@Output Decorator** and Event **Emitter**

```
<input type="text" (keyup.enter) = "onKeyup()" >
```

Event filtration

Template Variable in angular

```
<input type="text" (keyup.enter) = "onKeyup(username.value)" #username>
```

```
onKeyup(username) {  
  console.log(username);  
}  
[el] userna...
```

*Angular 2 way data Binding

To use ngModel one should import the Ffrom module in ts file

```
import { FormsModule } from '@angular/forms'
```

Then add formsModule in imports

```
imports: [  
  BrowserModule,  
  FormsModule,
```

```

imports: [
  BrowserModule,
  FormsModule,
  AppRoutingModule,
],
providers: [],
bootstrap: [AppComponent]
})
export class AppModule { }

```

```
<input type="text" [(ngModel)] = "userName">
```

ONE-WAY DATA BINDING



Can only bind data component to view

Example of one way data Binding is

```

<!-- one-way and Two-way Data Binding -->
<input type="text" value = 'one way data binding' >

```

Here the value property is hard coded/hand coded ,to use binding property we will use

The below format

```

<!-- one-way and Two-way Data Binding -->
<input type="text" [value] = 'textValue' (keyup.enter) = "onKeyup()" >

```

```

7  })
8
9 export class AppComponent {
10
11   bool:boolean = true;
12
13   userName: string;
14   textValue: string = "Value is coming from component"
15
16   buttonClick() {
17     console.log('Button Click Event worked');
18   }

```

TWO-WAY DATA BINDING



```

<!-- one-way and Two-way Data Binding -->
<input type="text" [value] = 'textValue' (keyup.enter) = "onKeyup()">
<br>
<input type="text" [(ngModel)] = 'textValue' (keyup.enter) = "onKeyup()">
  
```

ONE-WAY DATA BINDING

```

# String Interpolation - {{title}}
# Property Binding - [src]
# Class Binding - [class.text-red]
# Style Binding - [style.backgroundColor]
  
```

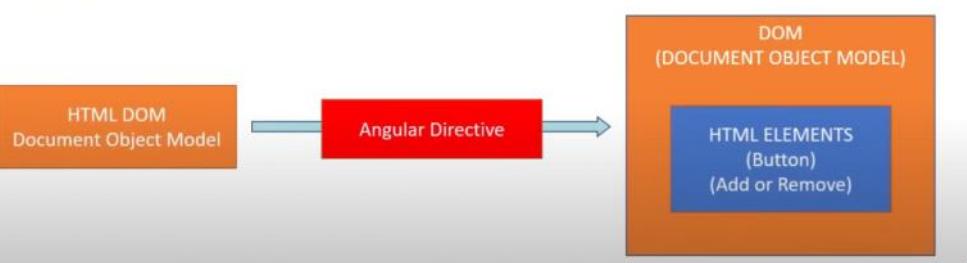
TWO-WAY DATA BINDING

```
# Two-way Binding - [(ngModel)]
```

ANGULAR – DIRECTIVE

An angular directive is special type of technology that can manipulate the DOM object,

Directives can do adding html elements removing html elements from DOM dynamically.



ANGULAR – DIRECTIVE (Components)

```
# Directives are Components Without a View  
# Angular Components are Directives, With a View
```

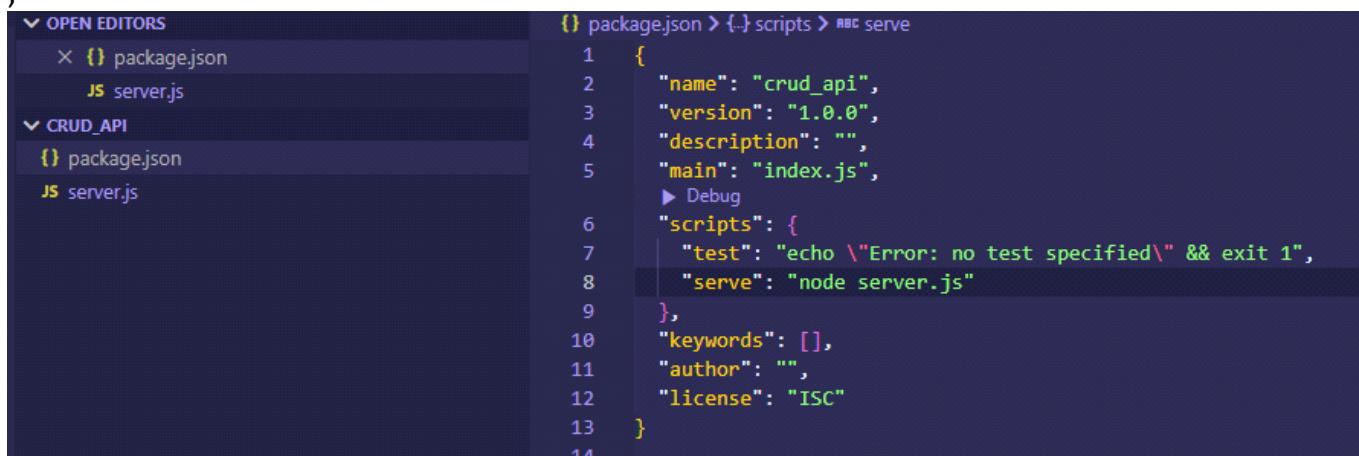
NgFor Directive

1. We use NgFor Directive to Render an Array inside the View
2. NgFor Directive is a Structural Directive.
3. With NgFor Directive we can Manipulate The DOM.

Node.js

15 December 2023 02:12 PM

To create a package.json file we will use npm init -y,
Then create a server.js file to run this file you have to add this file in package.json in script object



```
1 {  
2   "name": "crud_api",  
3   "version": "1.0.0",  
4   "description": "",  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1",  
8     "serve": "node server.js"  
9   },  
10  "keywords": [],  
11  "author": "",  
12  "license": "ISC"  
13 }  
14
```

And to run server.js file command in terminal is npm run serve.

To install express just give terminal command npm install/i express.

Routes will be created in server.js file .

Run basic git command

Git init

Git add .

Git commit -m "desired message"

To prevent refreshing again and again after an update in our project we will use nodemon

To install nodemon npm command is:- npm install/I nodemon -D

Add in scripts of server.js in scripts object



```
1 {  
2   "name": "crud_api",  
3   "version": "1.0.0",  
4   "descrip The main field is a module ID that is the primary entry point to your program.  
5   "main": "index.js",  
6   "scripts": {  
7     "test": "echo \"Error: no test specified\" && exit 1",  
8     "serve": "node server.js",  
9     "dev": "node server.js"  
10    },  
11    "keywords": [],  
12    "author": "",  
13    "license": "ISC",  
14    "dependencies": {  
15      "express": "^4.18.2"  
16    },  
17    "devDependencies": {  
18      "nodemon": "^3.0.2"  
19    }  
20  }  
21
```

Now we will run our app by npm run dev.

Connection to MongoDb compass

Npm install/I mongoose

And then use mongoose in your server.js file and with mongoose.connection("URI") you can create a connection

```
js server.js > fx then() callback > fx app.listen() callback
1  const express=require('express');
2  const mongoose=require('mongoose')
3  const app=express()
4
5  //routes
6  app. (parameter) res: Response<any, Record<string, any>, number>
7    res.send("Hello neers NodeApi.....")
8  }
9
10 //MongoDb connection
11 mongoose.connect('mongodb://localhost:27017/CrudApi').then(
12   ()=>{
13     console.log("Connected to MongoDB ")
14     app.listen(3001, ()=>{
15       console.log(`Node Crud_Api is running on port 3001`)
16     })
17   }
18 ).catch((error)=>{
19   console.log(error)
20 })
```

We won't be hard coding any values as if our port number and URL then we have to add dotenv dependency and create a .env file and put our variables there and to use .env file we will configure it by require("dotenv").config()

//

Basics

Zetwerk is a one stop complete manufacturing service provider which has a wide range of services that includes, Like zetwerk provide manufacturing services like die casting ,Extrusions (it a process of creating a shaped object such as rod or tube by forcing a material into mold), die casting, CNC machining , Forging, Injection Molding ,Sheet metal stamping, Providing a solution to dedicated assemblies, etc. And can provide any desired industrial need for special requirements.

Some common manufacturing sector where zetwerk has provided its services are Transportation, INDUSTRIAL MACHINERY & EQUIPMENT, CONSUMER PRODUCTS, ELECTRONICS & APPLIANCES, CONSTRUCTION & INFRASTRUCTURE, and even AEROSPACE & DEFENSE

And Some of Zetwerk manufacturing services include
1st:-Project management ,

Every customer is assigned a dedicated project manager to ensure a smooth and transparent delivery

2nd:-Quality Control

3rd:-Managing Logistics

4th:-Manage Supply Chain

Zetwerk offers manufacturing solutions across industries such as Oil & Gas, Renewables, Aerospace, Apparel and Retail. For customers, Zetwerk's manufacturing network ensures products are made faster, at competitive prices and with world-class quality.

In addition, Zetwerk in-house Manufacturing Operating System software, ZISO, enables digital selection of supply, real-time tracking across stages, visual updates of products getting made, seamless communication across stakeholders and quality assurance of the final product, thereby offering customers peace of mind.

For manufacturing partners, Zetwerk drives higher utilization of

manufacturing capacities and offers various portfolio services (including logistics, raw material procurement, working capital access) to drive revenue growth and optimize manufacturing costs.

Zetwerk has nearly unlimited manufacturing capabilities and capacity, ranging from CNC machining and die casting to 3D printing and apparel production. We serve customers in virtually every major industry, from consumer electronics and home appliances to aircraft engine components and infrastructure.

Simple Shiksha Task



Spreadsheet

Sr .No	Issues Noticed
1	Extra white space in the Main page carousel
2	Student Carousel not working properly, the slider has to be in working mode.
3	In Courses=>Industrial Training in most of the courses offered , the Enquire now button is going out of the box and it has to be aligned accordingly. https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-full-stack-web-development
4	The Enquire now button is only adjusted in Industrial Training in Data Analytics course so changes in all has to be made. URL is https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-full-stack-web-development
5	In Industrial Training in Data Analytics course slide in URL https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-in-data-analytics the description in the poster has to be change ,the present description is of full stack course. And overview section has to be updated accordingly. And also the topics that will be covered section has to be changed accordingly.
6	To change the favicon on website, suggested by Mrinal Gupta.
7	In Industrial Training in AI & Machine Learning the Poster description and Topics that has to be covered And Overview section has to be updated accordingly. https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-in-ai-machine-learning
8	In Industrial training In Internet of things changes of poster description, overviews and Topics that will be covered has to be change. https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-in-internet-of-things And same with Industrial Training in MERN Stack Development https://simpleshiksha.com/upskill-your-career/industrial-training/enroll?course_package=industrial-training-in-mern-stack-development
9	The upcoming events that are showing in website has to be updated, the events that are in the website are older ones. https://simpleshiksha.com/events
10	Enquire Now button requires adjustment in Technology / Software Development Python With Django Web Development. url:- https://simpleshiksha.com/technology/software-development/enroll?course_package=python-with-django-web-development

Git clone <https>Link of the code from repository

Git status :-to check the modified file

Status are of 3types 1st:-untracked(new file created),2 modified(old file modified),3rd staged(file added ready to commit)

Git commit -m "message"

Git push origin main

init - used to create a new git repo

git init

git remote add origin <- link ->

git remote -v (to verify remote)

git branch (to check branch)

git branch -M main (to rename branch)

git push origin main