**Day 1**

**15-03-2021**

**Full Stack :**

Program

Input a=10, b=20

Process sum = a+a

Output write sum

class Abc {

public int add(int x, int y) {

int sum = x+y;

return sum;

}

}

**Testing: We are going to check our code working fine or not without main method or functions.**

**Testing : It use to find the error or defect or bugs in application.**

**Testing divided into 2 types**

1. **Black box testing**
2. **White box testing**

**Black box testing**

**Input -----🡪 Output**

**White box testing**

**Input -🡪 Process -🡪 Output**

**Unit Testing : Unit testing is a type of white box testing.**

**Unit is where we write the set of line of coding to perform a specific task. Like function/ method or class or modules etc.**

**To achieve unit testing. Third party vendor provide API(Application Programming interface)**

**jUnit :jUnit is small testing framework open source framework provide by vendor which help to do the java programming unit testing.**

**To do jUnit testing it required jar file.**

**jUnit Test Case : jUnit Test case is class which contains more than one test methods with Annotation @Test.**

**Every @Test method can contains more than assertXXX() methods to check actual and expectation values.**

**jUnit Test Suite: Test Suite combines more than one test case classes.**

**JDBC :**

**Database :**

**Account Table : In Database**

**AccNo(PK), Name, Amount**

**1120 Ravi 1200**

**1121 Ramesh 1500**

**1122 Ajay 1600**

**commit;**

**AccountDao (Data Access Object)**

**class AccountDao {**

**public float getBalance(int accnot) {**

**JDBC logic**

**Get balance amount**

**}**

**}**

**Class AccountDao {**

**@Test void testGetBalance(){**

**}**

**}**

**Maven :**

**Gradle**

**Git :**

**Agile :**

**Docker :**

**Kubernetes :**

**Maven : Maven is known a build tools.**

**Build tool is responsible to compile the program, run the program, creating jar, war or ear file, downloading dependencies or jar files depending upon the application requirements. Creating the documentation for the projects.**

**Before Maven We were using ANT tool.**

**Netbean**

**Eclipse :**

**MyEclipse**

**RAD**

**Etc**

**Maven use POM.xml file.**

**POM is known as Project Object Model. It is a type of xml file which hold all configuration details for the maven projects.**

**Creating sample project using maven command prompt**

**mvn archetype:generate**

**Maven Project : For all we have to give groupName using Id.**

**groupId : name for more than project with id identifiers.**

**groupId : groupNameforMoreThanOneproject**

**artifactId : actualProjectName:**

**Maven Life cycle :**

**3 life cycle :**

1. **clean**
2. **build (default )**
3. **site**

**Every life cycle contains more than one phase :**

**Phase : phase is a group of orders goals.**

**Goal : Goal is a single unit of task which does some real work.**

**Goal 🡪**

**Phase 🡪**

**Life cycle 🡪**

**Maven phase :**

**mvn validate : mvn validate : it check the pom.xml file code.**

**mvn clean**

**mvn compile**

**mvn test**

**mvn install**

**mvn deploy**

**mvn exec:java –Dexec.mainClass=com.App**

**Day 2**

**16-03-2021**

**Creating the maven project using eclipse IDE**

**Maven tool use repository(Folder or directory) concept to hold to hold/ store dependencies(vendor Jar files) required for the application.**

**Maven uses local and remote repository.**

**Local repository:**

**Default path :**

**C:\Users\91990\.m2\repository**

**We have to add few tags in pom.xml file to download the external dependencies jar file. First maven search that version jar file from local repository part of m2 folder. If that version jar file available then that jar add the eclipse or maven projects. If that jar is not available in local repository it download from remote repository and keep in local repository for features reference.**

**If we want to download the external Dependencies**

<dependencies>

<dependency>

<groupId></groupId>

<artifactId></artifactId>

<version></version>

</dependency>

<dependency>

<groupId></groupId>

<artifactId></artifactId>

<version></version>

</dependency>

</dependencies>

**Here we have to remember groupId, artifactid and version of external jar files.**

**Maven documentation provide all jar file groupId,artifactid and version.**

**Creating jar or war file for user-defined project using maven.**

**mvn package ( This command we have to execute in that directory where pom.xml file present)**

**This command is use to create jar or war file depending upon type of project.**

**If you want user-defined name for the jar file**

**Then we have to write few tags in pom.xml file**

<build>

<finalName>MyJar</finalName>

</build>

Running jar file through command prompt

**mvn install : This command is use to add our local jar file into local repository part of .m2 folder.**

**Web Technologies**

**https://**[**www.google.com**](http://www.google.com) **URL**

**http/https(req)**

**---------🡪 Req---🡪**

**Client Server**

**🡨-----Res(http/https)-----**

**http: hyper text transfer protocol :**

**set of rules which help to communicate more than one machine or device.**

**s : secure**

**www : world wide web**

**google : domain or server**

**com : commercial**

**URL : Uniform resource locator :**

**HTML/HTML5 : display the contents on browser**

**CSS/CSS3 : apply good look and feel for web contents or presentation on contents.**

**JS(JavaScript): It is use to do action on contents.**

* 1. **: HTML,CSS and JavaScript**

**HTML : Hyper text mark up language. It is use to create the web page. Web page means display the contents in browser in different formats.**

**HTML contains lot of pre-defined tags or elements.**

**Syntax**

**<tagName> opening tag**

**</tagName> closing tag**

**Few contains doesn’t contains closing tag**

**<tagName/> self closing tags.**

**HTML is case insensitive.**

**Means we can write tags in upper case or lower case or any case.**

1. **Html**
2. **Head**
3. **Body**
4. **Title tag : This tag is use to display the title message on title bars. It must in between head tags.**

**HTML 4 version**

**<!doctype HTML public URL=”………….dtd”**

**Document type definition : This file contains the rule for the HTML file.**

**What the root tag name.**

**That tag contains how many child tag**

**Head and body**

**Body tags contains how many paragraph tags**

**0 to infinity**

**.xhtml**

**HTML 5**

**<!DOCTYPE html > : giving the instruction to browser we are going to write HTML 5 features.**

**. without this we can use all HTML5 Features.**

**In HTML5 they introduce more tags.**

**We can’t achieve all HTML5 features alone we have to depends upon JS and servers.**

**Java**

**Asp.net**

**Php**

**VS Code ( Visual Studio Code).**

**Hyper link <a href=”demo.html”>Click here</a>**

**External Hyper link**

**Internal Hyper link (book mark)**

**Image tag**

**List tag UL, OL etc**

**Table tags**

**Forms tags**

**HTML form internally use Query param concept**

**URL?user=Ravi&pass=1234**

**By default form use get method. If method is get the information send through URL using query PARAM concept.**

**Then we can use post() if we want to send the data with security.**

**If the method is post then information or data send through body part of request.**

**Performance wise get is faster than post method.**

**And in get we can send maximum 255 character data only.**

**CSS : Cascading Style sheet**

**CSS provide lot of property in the form of key-value pairs which help apply good look and feel for the web page which not possible or difficult with alone html.**

**If we are going control the contents as well as formatting style in html page. The maintain that application become very difficult.**

**Separation concern(Actual content and formatting style must be separate).**

**CSS mainly divided into 3 types.**

1. **Inline CSS**
2. **Internal css or embedded css**
3. **External css**

**Inline CSS**

**Syntax**

**<tagName style=”property:value;property:value;”></tagName>**

**tagName : p, h1 to h6, div, b, I, form, table etc**

**Internal or Embedded CSS**

**<style type=”text/css”>**

**Selector {property:value;property:value}**

**</style>**

**Selector**

1. **Universal selector : \***
2. **Specific selector : tagName**
3. **Multi specific selector : tagName,tagName**
4. **Class selector (local) : tagName.className**
5. **Class selector (global) .className**
6. **Id selector : #idName**

**Class selector Vs Id selector**

**Two tags may have tagName, same name attribute can be belong to same class.**

**Class : collection of more than one tags.**

**Id must unique for tags.**

**Using Id only we can read, write and update the DOM(Document Object Model).**

**Day 3**

**17-03-2021**

**External CSS**

**HTML5 Semantic tags**

**Header, footer, main, section, nav etc.**

**JavaScript : JavaScript was object based interpreter scripting language.**

**Till ES5**

**ECMA (European Computer Manufacture Association)**

**ECMA is a concept.**

**JavaScript is a one of the implementation of ECMA.**

**Object based Vs Object Oriented**

**The main use of the JavaScript to do validation on client side.**

**Using JavaScript we can do programming on web page.**

**JavaScript tags**

**<script type=”text/JavaScript”> opening tag**

**</script> closing tag**

**This tag we have to write in between head tag or body tag of html web page.**

**And in one web page we can write more than one script tags.**

**Variables and data types**

**In JavaScript to declare the variable we use var keywords.**

**Syntax to declare the variable**

**var variableName;**

**java script is known as loosely data types.**

**Depending upon what type of value we assign it behave that type of data types.**

**var a; // default value of a is undefined**

**var n=10; // number data types consider**

**var m=10.10// number data types consider**

**var name=”Ravi Kumar” //string consider**

**var res = true; //Boolean consider**

**var obj = new Date(); //object reference.**

**Operators**

**Arithmetic Operator : +,-, \*, /, %**

**Conditional operator : >, >=, <, <=, ==, !=**

**Logical operator : &&, ||, !**

**Assignment operator : =**

**Increment and decrement : ++, --**

**Triple : ===**

**Typeof :**

**If statements**

1. **Simple if**
2. **If else**
3. **Nested if**
4. **If else if**

**Switch statements**

**Looping**

**While loop**

**Do while loop**

**For loop**

**function : function is use to write set of instruction to perform a specific task.**

**2 types**

1. **Pre-defined (global functions).**
2. **User-defined functions.**

**Pre-defined functions**

1. **Alert(“msg”): this function is use to display pop message.**
2. **Prompt(): This function is use to receive the value through keyboards.**
3. **parseInt() : it covert string to integer.**
4. **parseFloat() : it convert string to float.**
5. **eval() : it is use to convert string to number(with or without decimal).**
6. **Confirm() : This function takes the confirmation from user to do the task.it contains two button ok and cancel. If user click ok it return true else return false.**

**do {**

**alert(“1 2”)**

**prompt()**

**eval()**

**switch statement**

**case 1 :Add**

**case 2: Sub**

**default : wrong choice**

**confirm : do you want continue**

**}while()**

**User – defined functions**

1. **Normal syntax to create user-defined functions**

**function functionName(parameterList) {**

**}**

**Events :events is a interaction between user and component (html tags).**

**Event provide bridge between html code and java script functions.**

**Type of events**

**In JavaScript all events start with on followed by event name.**

**onClick**

**onDblclick**

**onMouseOver**

**onMouseOut**

**onKeyUp**

**onKeyDown**

**onSubmit**

**onChange**

**onBlur**

**onFocus**

**etc**

**DOM : Document Object Model**

**All html tag is know is a dom like html, head, body, p, h1 etc.**

**In browser memory DOM hierarchy created.**

**Html**

**Head body**

**Meta p textNode**

**Title div**

**Style**

**DOM API (Document Object Model Application Programming interfaces).**

**JavaScript, Java,C#, python etc.**

**Using DOM API we can read, write or update html contents dynamically.**

**To read the text field data using DOM**

**document.getElementById(“idName”).value;**

**Form Validation**

**We can do form validation useName is required, password required, minlenght and max length etc.**

1. **Using JavaScript**
2. **Using HTML5 Features**

**Day 4 : 18-03-2021**

**JavaScript Objects**

**object : any real world entity**

**state or properties—have—fields/variable**

**person**

**behaviour—do/does –function/methods**

**bank**

**animal**

**car**

**customer**

**employee**

**object**

**property**

**behaviour**

**object**

**property**

**behaviour**

**object**

**property**

**behaviour**

**object**

**Divided into 2 types**

**Pre-defined objects**

**BOM : Browser Object Model: BOM hierarchy provide browser related details.**

**DOM : Document Object Model : DOM hierarchy provide details about web page contents.**

**JavaScript provide Basic pre-defined objects**

**String**

**Date**

**Array : in JavaScript array is use to store the heterogeneous elements.**

**Array is dynamic memory size.**

**Boolean**

**Number**

**Math**

**Etc**

**String**

**window : window is a top most object from BOM hierarchy. Window object contains lot of property, method as well as other object as a property.**

**alert(): alert() or window.alert()**

**prompt();**

**confirm();**

**setTimeout()**

**setInterval()**

**clearInterval()**

**JavaScript or web application communication API:**

1. **Synchronous communication**
2. **Asynchronous communication**

**Synchronous communication :**

**L1**

**L2**

**L3**

**L4**

**L1 to L4 code execute sequence.**

**L2 code depends upon the l1 then L3 code depends upon the l2 l4 code depends upon the l3.**

**dis1();**

**dis2();**

**dis3();**

**dis4();**

**Client Server**

**1st Request**

**2nd Request**

**3rd Request**

**4th Request**

**Above points communication is known as Synchronous communication.**

**Asynchronous communication:**

**L1 to L4 execute in dependently**

**Dis1() to dis4() execute in dependently**

**Req1 to Req4() execute in dependently.**

**AJAX :**

**Asynchronous JavaScript and XML.**

**setTimeout()**

**setInterval()**

**clearInterval()**

**The 3 function do the task Asynchronous communication on view side.**

**Callback function : passing the function body or function itself to another function as a parameter is known as callback functions.**

**User-defined objects**

**In JavaScript till ES5 no class. So if we want to create the object we have to take the help function to create the user-defined objects.**

**Syntax**

**function ObjectName() {**

**this.propertyName=value;**

**this.propertyName=value;**

**}**

**this : it is keyword which help to refer the current object.**

function Person() {

    alert("Object created using constructor")

    this.name = "Ajay";

    this.age = 21;

    this.display = function() {

        document.write("Name is  "+this.name+"<br/>")

        document.write("Age is "+this.age+"<br/>");

    }

}

//Person();   //now it is like a normal function

var p1 = new Person();

alert("name is "+p1.name);

alert("age is "+p1.age);

p1.display();

var p1 = new Person(100,"Ramesh");

alert("name is "+p1.name);

alert("age is "+p1.age);

p1.display();

var p2 = new Person(101,"Raju");

alert("name is "+p2.name);

alert("age is "+p2.age);

p2.display();

**Inheritance : Inheritance is use to inherits the properties and behaviour of old class(functions) to new class(functions)**

**Day 5 : 19-03-2021**

**ES6 Features**

**Till ES5 we can declare the variable using var keywords.**

**From E6 onwards we can declare the variable using var, let and const.**

**JavaScript is one of the implementation of ES5 as well as ES6 or ES7.**

**TypeScript is also one of the implementation of ES6 or ES7. TypeScript is a super set of JavaScript which support all features of ES6 and ES7.**

**Angular 2 to 11--🡪 TypeScript**

**React JS 🡪 JavaScript using ES6.**

1. **Using var keyword we can declare global scope.**

**Using var keyword we can re-declare same variable once again with same value or different values.**

1. **Using let keywords we can declare local or block scope.**

**Using let keyword we can’t re-declare same variable with same value or different values.**

**JavaScript hosting :**

**In C**

**void info();**

**void main() {**

**info();**

**}**

**void info() {**

**}**

**C++**

**class A;**

**void main() {**

**A obj;**

**obj.dis();**

**}**

**class A {**

**public : void dis() {**

**}**

**};**

**JavaScript Hoisting : It is use to move all declaration on top the current scope or page.**

**Hosting apply for declaration not for initialization.**

**HTML/HTML5, CSS/CSS3, JavaScript Using ES5**

**jQuery : jQuery is a external JavaScript library which provide set of function which internally connected to each to other to do DOM Operation very easily.**

**Angular : Angular is a framework. Framework is always very big for to do enterprise application. Framework follow standard rules and regulation. The implementation of all design pattern is taken care by framework.**

**70% 80% task taken care by framework. Framework is also known as template or protocol.**

**React JS : React JS is light weighted library not a framework.**

**Vue JS**

**Node JS : Node JS is not a JavaScript library nor Framework. It is a run time environment for JavaScript library or framework. Like is JRE in Java Node JS in JavaScript.**

**Before Node JS JavaScript was using only for Client side scripting language.**

**FrontEnd developer depends upon the backend technologies or developers.**

**Java Web Technologies, Asp.net,Php, Python etc.**

**After Node JS we can write JavaScript on server side. Means using Node JS Server Side scripting possible.**

**“Node’s JS goal is to provide an easy way to build scalable network programming”.**

**Using Node JS we can achieve non block IO Operation.**

**Node JS Programming we can achieve using JavaScript.**

**Node JS is open source, platform independent, server side and networking application.**

**JEE (Servlet, JSP(Java Server Pages), EJB (Enterprise Java Bean), Spring MVC etc**

**Server -🡪 Tomcat, Web Logic, JBoss,**

**Asp.net -🡪 IIS Server**

**Normally all server are thread base.**

**Tomcat Server : 1000 people can access application at same time.**

**Normally are server are thread base but Node JS allow us to create user-defined server. Those server are event based. Node JS server serves each request in a event loop that can be handle simultaneously request with the help callback and asynchronous communication.**

**Node JS it is not a framework or library**

**Not for beginner.**

**Multi threaded**

**In simple**

**Node JS = Run time environment + JavaScript modules.**

**Using Node JS we can run JavaScript using terminal or through command prompt.**

**Node JS providing own terminal.**

**We can run node js program using two**

1. **Using normal command prompt**

**Open command prompt**

**Then write node**

**Node terminal is know as**

**REPL (Read Eval Print Loop).**

1. **Using node terminal.**

**We can’t use document and window object in node js program.**

**Node JS provide global objects.**

**console: it is a type of global object which help to display the object in console.**

**Process : It is also type of global object which give details about your machine processor.**

**demo.js**

**ES6 Features**

1. **let, const and var.**
2. **for in loop and of loop**
3. **OOPs**
4. **Class, constructor, static, etc.**

**Node JS modules**

**Modules : modules is a collection of function, property/variable or ES5 or ES6 style classes.**

**Module is like package in java.**

**These modules divided into 2 types.**

1. **Local modules.**
2. **External module provided by third party vendors.**

**As well as we can create user-defined modules.**

**File handling programs**

**IO package in java**

**Local modules**

**fs modules.**

**This module is use to do file operation synchronous as well as asynchronous operations.**

**http module : This module is use to create basic server.**

**Day 6 : 22-03-2021**

**http module is a basic core module. Which provide very basic features of http.**

**To receive the value from form using get or post with the help of http module more complex.**

**There are various third party open – source module available to achieve client server application using node js.**

**To download those third party module we have to use**

**npm (node package manager). It is like a mvn in java.**

**First to check the npm version.**

**Express framework : It is opens source framework base upon node js which help to do client - server application using node js.**

**Express module provide features get post put delete (http methods). It provide routing features.**

**Express module base upon the http module.**

**To install any external module we have to use the command as**

**Express support http protocol all features.**

**npm install –g moduleName**

**globally**

**or**

**npm install moduleName**

**locally.**