Day 1 : 21/10/2023

Backend technology -🡪 Java and connecting database using JDBC.

Java

Core Java

J2SE J2EE J2ME

JavaSE JavaEE JavaME

JSE JEE JME

Java Standard Edition Java Enterprise Edition Java Micro Edition

Desktop application Servlet and JSP : web application

Console base application We will teach how to create

Basic Java Programming Rest API

OOPs Concept those rest api we will

Object, call in angular

Class, maven build tools

Exception handling

Collection Framework with Data structure

JDBC topic to connect MySQL

Java8(Lambda Expression and Stream API)

and 11 and 15

Regex

Java 22

Intro to Mongo DB

No SQL Database

Using Java we can connect to database using

1. JDBC
2. ORM -🡪 Hibernate and JPA.
3. Using Core Java, Servlet /JSP we can use JDBC or Hibernate or JPA.
4. Spring Framework we can use JDBC or Hibernate or JPA.
5. In Spring boot JDBC or JPA or Spring Data.

Phase 3 or Course 3

Spring framework and Spring boot

Junit 5 testing tool

Course 4 : Testing and deployment

Testing

TestNG, Selenium,

Docker, CI and CD using Jenkin, Overview of AWS

In AWS we will teach you how to deploy application in EC2 instance using Docker with CI and CD tool ie Jenkin

Frontend -🡪 angular

Backend 🡪 spring boot

Database -🡪 MySQL

Day 2 : 22/10/2023

Java : Java is pure object oriented and platform independent programming language.

Java develop in nov 1995. The initial name of java is Oak.

Version of Java

Java 1.0,1.1,1.2 7, 8,9,11,15 22

Java is an open source.

From java 11 onward java is not open source.

Java was belong to sun micro system

Now it is part of oracle.

Oops

object : object is any real world entity. Example

property or state or variable -🡪 have 🡪 variable, fields etc

name, age, weight, height, colour etc.

int, float, char, double, string etc.

Person

Behaviour or function or methods -🡪 do/does 🡪 function or methods.

Teaching(), sleeping(),talking(), typing(), etc

Bank

Animal

Car

Customer

class : blue print of object or template of object or user defined data type which help to describe the object.

Syntax of class

class ClassName {

fields or variable declaration

method or function declaration

}

class Demo {

main method : pre defined methods.

}

Class name must follow pascal naming

1. If class contains one world first letter upper case.
2. If class contains more than one world each world first letter upper case.

Syntax to write method in java

returnType methodName(parameterList) {

Method body;

}

void display() {

}

Method name is display no passing parameter and no return type.

void add(int x, int y) {

}

Method name is add. We need to pass two parameter of type int but no return type.

String sayHello(String name) {

return “Welcome user “+name;

}

Method name is sayHello. Need to pass one parameter of type string and return string value

In java all method must part of class.

class Test {

public static void main(String args[]) {

System.out.println(“Welcome to Java”);

}

}

Method name and variable name must be follow camel naming rules.

1. If variable name or method name one world. Then it must be in lower case. Like main(), sleeping(), display(), read() etc and id, name,salary etc.
2. If variable name or method name more than one world then from second world first letter upper case like displayInfo(), calSalary(), empId, etc.

Data types : Data type is a type of data which tells what data type data it can hold.

a=10; python

var b=20; js

let c:number =30; ts

datatype variableName;

datatype variableName=value;

In java data types are divided into two types.

1. Primitive data types :it is use to store only value

8 types

1. byte 1 byte -128 to 127
2. short 2 byte
3. int 4 byte
4. long 8 byte

without decimal point

1. float 4 byte
2. double 8 byte

with decimal point

1. char 2 byte in single character
2. boolean 1 bit true or false value

Operator :

1. arithmetic operator : +, -, \*, /, %
2. logical operator : &&, ||, !
3. conditional operator : >, <, >=, <=, ==, !=
4. assignment operator =
5. increment and decrement operator : ++, --

if statement

1. simple if
2. if else
3. if else if
4. switch statement

syntax of switch statement

int label =1;

switch(variableName) { variable type must be int, char or string in java.

case 1:block1

break;

case 2:block12

break;

case 3:block3

break;

case 4:block4

break;

default : default block ;

break;

}

Looping : looping is use to execute the task or statement again and again repeatedly

Till the condition become false.

Initialization

Condition

Task

Increment and decrement.

1. While loop : entry loop
2. Do while loop : exit loop
3. For loop

Type casting :

Converting from one data type to another data type is known as type casting.

Two type of type casting.

1. Implicit type casting
2. Explicit type casting

int family

----------------🡪 implicit type casting -----🡪

byte short int long

🡨------------ explicit type casting -------------

int to float implicit

float to int explicit

in java by default every decimal number double consider.

1. Non primitive or reference data types. : it is use to store value as well as reference of another data types.

4 types of non primitive

1. array :array is user defined or reference data type which help to store

same type of values.

In java array is known as fixed in memory size.

datatype arrayname[]; array declaration

int num[];

int a;

int num1[]={10,20,30,40,50}; array declaration with initialization

array memory creation syntax

datatype arrayname[]=new datatype[size];

int num2[]=new int[10];

num2 can hold 10 value. Start from 0 to 9 position.

Taking the value through keyboard in java

Using Scanner class.

Scanner is a pre-defined class part of util package.

Package is a collection of classes as well as interfaces.

We need to create Scanner class object.

Syntax to create Scanner class object

Scanner obj = new Scanner(System.in);

While creating scanner class object we will get error. Because Scanner class part of util

Package. We need to import it.

1. class

Scanner : pre defined class which help to scan the value through keyboard.

String : in Java String is a pre defined class. which help to store more than one

Character in double quote.

String is pre defined class part of lang package.

By default java imported lang package.

1. interface
2. enum

in Ts or JS

class Customer {

}

var obj1 = new Customer();

let obj2=new Customer();

Scanner obj1 = new Scanner(System.in);

obj.nextInt(); int value

obj.next() string value

obj.nextFloat() float value

Day 3 : 28/10/2023

Java OOPs concept

object : any real world entity.

Property or state -🡪 have -🡪 variable or fields.

Person

Behaviour -🡪 do/does -🡪 functions or methods

Bank

Car

Animal

Customer

Employee

class : blue print of object or template of object or user defined data type which help to describe the object.

Method or function syntax

return type methodName(parameterList) {

}

void info() {

coding….

}

Method name is info. For this method we are not passing any parameter while calling

And this method doesn’t return any value to caller method.

void add(int x, int y) {

}

Method name is add while calling this method we need to pass 2 parameter of type int and not return type.

String sayHello(String name) {

Coding

return “Welcome to my method user “+name;

}

Method name is sayHello. We need to pass one parameter of type string and it return string value to caller method.

Fields or variable.

In Java variable or fields are divided into 3 types.

1. Instance variable
2. The variable which declared inside a class but outside method including main method also is known as instance variable.
3. Instance variable hold default value base upon their data types.

Int family 🡪0

Float family 🡪0.0

Char -🡪 space or white space

Booean –> false

String 🡪 null

1. Instance variable we can use all method directly but method must be part of same class and it must be non static.
2. Local variable
   1. The variable which declared inside a method is known as local variable.
   2. Local variable doesn’t hold default value we have to initialize.
   3. The scope of the variable within that method where it declared.
3. Static variable

Constructor : constructor is a type of special method which help to create the object.

Pts.

1. Constructor have same name as class itself.
2. Constructor no return type not even void also.
3. Constructor get call automatically whenever we create the object that class.

class Employee {

Employee() {} in Java Constructor

constructor(){} in typescript or angular

}



When local or parameter variable and instance variable have same name then local variable or parameter variable hide the visibility of instance variable.

If we want to refer to instance variable this we need to use this keyword. this is a keyword which refer to current to object.

Parameter constructor

Encapsulation : binding or wrapping data or variable and code or method/function is single unit is known as encapsulation.

Example : class.

JavaBean class

Inheritance :

Inheritance is use to inherits or acquire the property and behaviour of old class to new class.

class OldClass { super class or base class or parent class.

property

behaviour

}

class NewClass extends OldClass{ sub class or derived class or child class.

property

behaviour

}

Types of inheritance

1. Single inheritance : one super class and one sub class

class A { }

class B extends A { }

1. Multilevel inheritance : one super class and n number of sub class connected one by one.

class A { }

class B extends A { }

class C extends B { }

class D extends C { }

1. Hierarchical inheritance : one super class and more than one sub class connected directly to super class.

class A { }

class B extends A { }

class C extends A { }

class D extends A { }

1. Multiple inheritance : more than one super class and one sub class

class A { }

class B { }

class C extends A,B { } Java doesn’t support multiple inheritance. Error

this type of inheritance in java we can achieve using interface.

class Employee {

id,name,salary

readEmployee()

displayEmployee();

}

class Manager extends Employee {

numberOfEmp;

readManager()

displayManager();

}

class ProjectManager extends Manager{

clientInfo

readProjectManager()

displayProjectManager();

}

class Developer extends Employee{

techName;

readDeveloper()

disDeveloper()

}