Day 1 :

06-01-2023

Full Stack :

Java

Python

.net

MEAN Stack

MERN Stack

Java Full Stack

Phase1

SDLC

Software Development Life cycle.

SDLC Model

Water fall model

Increment model

V model

Agile

Etc

Git : DevOps

Login (SignIn and SignUp) : HTML

Dev1 SignIn

Dev2 SingUp

Git

Manager -🡪

Java Programming

Basic Program

OOPs

Procedure

Object oriented programming

Functional programming

Aspect oriented programming

Phase 1 :

SDLC and Agile

Git

CI/CD

Core Java and data structure

Phase2

Intro to cloud

AWS : Overview : install Jenkin etc.

Into to database mysql or oracle or mongo db.

Phase3

Spring framework or spring boot

Phase4

UI html, css, js angular / react

Phase5

Testing selenium, docker, Kubernetes,

CI and CD

AWS Overview

Program : set of instruction to perform a specific task.

Input read a=10,b=20,sum

Process sum = a+b;

Output write sum

UML

ER-Diagram

SDLC

STLC : Software Testing Life cycle

JSON : Java Script Object Notation

XML :

RDBMS : MySQL table format

No sql : Mongo DB json format

Neo4J graph format

Day 2 :

06-02-2023

Git

Maven

Gradle

Docker

Cloud tech

Etc





Day 3 :

06-05-2023

DevOps

Day 3

Git

Day 4

CI and CD Tool Using Jenkin

Git : Git is distributed sub version control tool which help to record the flow the application.

Git is an open source software.

Dev1 login page.

Dev2 Remote Repository

Dev3

Please login to virtual lab.

Then create the Folder

Then create the file with some contents

Inside a open the terminal to verify git version

git --version

git init This command is use to local folder as local git repository.

ls -a this command display all files and folder including .git folder. (Unix command).

git status This command is use to check current status of local repository

git add filename This command is use to add the file from local machine or folder to

stagging area.

Or

git add . we can add all file and folder present current location.

git commit -m “commit message” this command is use to push the file

from git stagging area to local repository.

git config --global user.email “[akash300383@gmail.com](mailto:akash300383@gmail.com)”

git config --global user.name “akash”

In Same folder create another file with simple message.

In terminal

git status

git add .

git status

git commit -m “task2”

git branch : Branch is like a pointer which hold more than one commit details.

Using git branch command we need to check default branch details

By default branch name can be master or main.

Command to create user defined branch

git branch branchname;

git branch : this command to check all branches present in local repository

git branch A This command is use to create user defined branch

git checkout branchname: This command is use to switch from one branch to another branch

git merge branchname : this command is use to merge user defined branch code to current branch

git merge A

git branch -D branchName

git branch -D A This command is use to delete the branch

if we want to share the code to other team then we need to configure our local repository to remote repository

Remote repository can be

Git hub

Git lab

AWS

Azure

git branch -M main : rename local branch name

git remote add origin URL : this command is use to link local repository to remote repository.

Day 4 :

git remote add origin <https://token@github.com/Kaleakash/test_repository.git>

ghp\_MRDeF2TlL5gakBPq7c1WLIKkVucHxh1Q5iuq

git remote remove origin this command remove remote repository from

local repository

git remote add origin <https://token@github.com/Kaleakash/test_repository.git>

please create separate folder

open the terminal and write the command as

git clone <https://github.com/Kaleakash/test_demo.git>

once you clone test\_demo folder created in your local machine

that folder contains test file.

Next file if you need any new changes then open the terminal inside a folder and write

The command as git pull

git branch -M main

CI and CD : Continuous Integration and Continuous Deployment / Delivery

Remote Repository : GitHub /GitLab/ AWS / Azure etc

Manager

Login page

Dev1 Shared Repository (structure of project present in main or master branch).

application page

Dev2

Feedback page

Dev3

Dev1 create user defined branch with some rules and start the working inside that branch.

git add .

git commit -m “login page created”

git push -u origin branchName;

After dev1 or dev2 or dev3 when we merge the code to actual working code

We need to compile, run , test (build the project).

May be we can build success fully or we can get the error.

We are combining more than one developer code ie integration phase.

Jenkin is an Open source CI and CD tool created using Java technology or language.

Java is an open source. Jenkin is a plugin base CI and CD.

Java : Java is an open source, pure object and platform independent programming language.

Java introduce in nov 1995. Initial name of the Java is Oak.

OOPs : Object Oriented programming

Object : object is any real world entity.

Property or state -🡪 have (data type int, float, char, string,boolean)

Person

Behaviour 🡪do/does 🡪 action 🡪 teaching(), listening(), sleeping(), eating() etc

Bank

Wheel(int), colour(string), price(float) etc -🡪 have

Car

Start(), appliedGear(), moving(), stop() etc 🡪 behaviour

name,

Employee

Customer

Etc

Online application :

Pid,pname,price,qty etc.

Product

addProject, updatePrice, deleteProduct, viewProduct etc

viewProduct , orderTheProduct etc.

Order

Account

Login

Manager

Customer

etc

object is concept.

class : class is known as blue print of object or template object or it is user defined data type which help to create the object.

Class syntax :

class Car {

property

behaviour

}

class Car {

int wheel;

String colour;

float price;

void start() {

}

void stop() {

}

}

Car innova = new Car(); car class object created.

Car bmw = new Car(); another class object created. (memory created);

06/12/2023

Eclipse is a type of Open source IDE (Integrated development Environment ) . it is open source

Tool which help to write java program.

Package is a collection of classes and interfaces.

Package is just like a directory or folder which contains more than one class as well as interface.

Data types

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

In java data types are divided into 2 types.

1. Primitive type : it is use to store only value.

8 primitive types

byte 1 byte -127 to 128 (range number)

short 2 byte

int 4 byte

long 8 byte without decimal

float 4 byte

double 8 byte with decimal

char 2 byte any single character

boolean 1 bit true or false value

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

Variable : variable is a name which hold some value.

Syntax

datatype variableName;

int a; declaration

int b=20; declaration with initialization.

double b;

boolean result;

a=10;

a = true;

a=”akash”;

Operator : operator are use to do some operation on variable.

Arithmetic Operator : +, -, \*, /, % modules (remainder)

Assignment : int a=10;

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

int a=10;

int b =20;

boolean res;

res = a>b;

increment and decrement

++ --

int a=10;

a=a+1;

System.out.println(a);

a++;

System.out.println(a);

a--

System.out.println(a);

Conditional statement : it is use to execute set of statement base upon condition.

1. If statement

If(condition) {

True block

}

1. If else

If(condition) {

True block

} else {

False block

}

1. If else if

if(condition1) {

true block

} else if(condition2) {

True block

}else if(condition3) {

True block

}else {

False block

}

Switch statement

In switch statement user can take the decision which block we need to execute.

Syntax

switch(variableName) { variable can be int family or char no float/double

case value1: block1;

break;

case value2: block2;

break;

case value3: block3;

break;

case value3: block3;

break;

default : defaultblock;

break;

}

Taking the value through keyboard in Java

In java we can take help of Scanner class.

Scanner is a pre defined class which provide set of method

Which help to take or scan the value through keyboard.

Syntax to create Scanner class object

Scanner sc = new Scanner(System.in);

This Scanner pre defined class part of util package.

We need to import that package while using scanner class.

Outside a class

We need to import

import java.util.Scanner;

Looping

Looping is use to execute the task again and again till the condition become false.

Initialization : start and end position i=1,n=10

Condition it must be true i<=n or i>=n

Do the task System.out.println(“Hello”);

Increment or decrement i++ or n--

While loop

int i=1,n=1000;

while(i<=n) { entry loop

System.out.println(“Hello”);

Syste.out.println(i);

i++;

}

Do while loop : this loop execute at least once doesn’t matter condition true or false.

int i=1,n=1000;

do{

System.out.println(“Hello”);

Syste.out.println(i);

i++;

}while(i<=n); exit

For loop

Array : Array is reference or non primitive data type which help to store more than one

Value of same type.

int a=1,b=2,c=3,d=4;

array declaration

datatype arrayName[];

int num[]; it allow to store more than one value of type int.

int n;

int num[]={10,20,30,40,50}; this num variable hold five values.

int a=10;

array follow index concept. The index value start from zero position.

System.out.println(num); // That variable address.

System.out.println(num[0]);--🡪10

System.out.println(num[1]);--🡪20

a=20;

sid,sname,sage

int sid[];

int sid;

to retrieve the value from array we use for loop

syntax

1 2 4

for(initialization ; condition; increment/decrement) {

body of the loop; 3

}

Array memory creation

datatype arrayname[]=new datatype[size];

int abc[]=new int[10]; default value for int array is zero.

System.out.println(abc[0]); 0

System.out.println(abc[9]); 0

System.out.println(abc.length); 10

Method or functions

Method or function is use to write set of instruction to perform a specific task.

Using function or method we can do re-usability of code.

Syntax

accessSpecifiers returnType methdoname(parameterList) {

}

Access specifiers optional in method.

No passing parameter as well as no return type.

void display() {

set of code.

}

Void is a keyword no return any value to caller method.

char c=’b’;

String means combination of more than one character enclosed in double quote.

In Java store more than one character java provided pre defined class ie String.

String :

Syntax to store the value in String class

String name1 = “Akash”; //literal style

String name2 = new String(“Akash”); // memory creation style using new keyword.

object : object is any real world entity which have properties and behaviour

Car

Employee

Person

Bank

Animal

class : class is blue print of object or template of object.

``

Day

Date : 06/15/2023



Constructor : Constructor is a type of special method. Which help to create the object in all OOPS language.

Pts

1. Constructor have same name as class itself.
2. Constructor doesn’t contain any return type not even void also.
3. Constructor get call automatically when we create the object of that class. But for method we need to call explicitly to execute that code.

If we not write any constructor in program inside a class by default JVM (Java Virtual Machine)

Provide default constructor which doesn’t contains any logic it only help to create the memory. If we want to do any custom logic when memory created like initialization then we need to write empty constructor. If we write explicitly any constructor then JVM doesn’t provide any default constructor.

Parameterized constructor : constructor can take 1 or more than one parameter.

In the life of the object if we want to do any task only one time that type of task we need to write inside an empty or parameter constructor.

If the life of the object if we want to perform any task again and again that type of task we need to write inside method.

JDBC : Java Database Connectivity : using this concept with core java as well web application we can store as well as retrieve data from database ie mysql or oracle or mongo db etc.

ORM : Object Relation Mapping : Hibernate or JPA.

Encapsulation : Binding or wrapping data (variable) and code (function or methods) in single unit is known as Encapsulation.

In Java or all oops language we bind variable and method inside a class. no one access our variable and method directly without creating object of my class.

class is good example of encapsulation.

To make our data secure we need to add private key for all instance variable.

If we declare the instance variable as private we can’t change as well as can’t access instance variable

Value outside class through object.

Inheritance : Inheritance is use to inherits properties and behaviour of old class to new class.

class OldClassName { super class or base class or parent class

properties

behaviour

}

class NewClassName extends OldClass{ // sub class or child class or derived class.

property

behaviour

}

class A {

void dis1() {}

}

class B extends A {

void dis2() {}

}

A obj1 = new A();

obj1.dis1();

B obj2 = new B();

obj2.dis2();

obj2.dis1();

Types of Inheritance

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

class Employee { } oner super class

class Manager extends Employee { } one sub class

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

class Employee { } super class

class Manager extends Employee { } sub class

class ProjectManager extends Manager { }

1. Hierarchical Inheritance : one super class and n number of sub classes directly connected to super class.

Class Employee { }

Class Manager extends Employee {}

Class Developer extends Employee {}

Class Tester extends Employee {}

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

Class Father {}

Class Mother {}

Class Child extends Father, Mother {} Error in this line multiple inheritance java doesn’t support using class it will support using interface.

class Employee {

id,name,salary with methods.

}

class Manager extends Employee {

numberOfEmp with methods.

}

class ProjectManager extends Manager {

clientDetails and methods

}

class Developer extends Employee{

project name and with some methods .

}

Day

Date : 06/22/2023

Polymorphism : One name many forms or many implementation.

2 types

Compile time polymorphism : javac Demo.java

Static binding or early binding

Example : Method Overloading

Method Overloading : Method have same name but different parameter list (ie type of parameter list or number of parameter list must be different)

Run time polymorphism : java Demo

Dynamic binding or late binding

Example : Method overriding

Compile and interpreter

Compiler convert whole code in another format like byte code in java.

Interpreter check covert the code line by line

Method overriding : Method have same name and same method signature number of parameter list, type of parameter list and return type must be same.

To achieve method overriding we need to use inheritance. Without inheritance we can’t achieve method overriding.

void info() {

logic different

}

void info() {

logic different

}

Super class must be generic

Sub class must be specific

Class Bike

Class Honda extends Bike

Class Activa extends Honda

Car

Bank

abstract , final and static keyword

abstract keyword :

1. abstract is a keyword we can use with method and class but not with variable.
2. abstract method : method without body or incomplete method or without curly braces is known as abstract method.

abstract returnType methodName(parameterList);

abstract void withdraw(); // method declaration

1. if class contains abstract method ie one or more that class we need to declare as abstract class.

abstract class className{

}

1. whichever class extends abstract class that class must be provide the body for all abstract method belong to super class.
2. abstract class can contains normal as well asl abstract method.
3. If class is abstract we can’t create the object of that class.

final keyword :

1. Final keyword we can use with variable, method and class.
2. Final variable : to declare a constant value we use final keyword.

final double PI=3.242;

PI=12345; error

final int A=10;

A=20; error

int b=30;

b=50;

1. final method : if method is final sub class can’t override that method. Final method in sub class we can use it but can’t override it.
2. final class : if class is final we can’t inherits that class means we can’t make sub class for final class.

static : static keyword we can use with variable and method but not with class.

1. if variable is static we can call or access that variable with help of class name. object is not required.
2. If method is static we can call that class method with help of class name object is not required.
3. Static variable we can access through object also.
4. Static metho we can access through object also.



06/23/2023

Abstraction : hiding the internal implementation without knowing background details.

String name = “Welcome to Java Training”;

name.toUpperCase();

interface : interface is a reference data type. Interface also known as 100% pure abstract class.

syntax

interface interfaceName {

variables;

methods;

}

In interface by default all variables are

public static and final

by default all method interface are public and abstract.

interface Abc {

public static final int A=10;

int B=20;

public abstract void dis1();

void dis2();

}

interface Abc {

int A=10;

void dis1();

}

interface Xyz {

int B=20;

void dis1();

void dis2();

}

interface Mno extends Abc,Xyz{

int C=30;

void dis3();

}

One interface can extends another interface but interface can extends more than one interface. But class can extends only one class.

class Demo implements Abc,Xyz {

dis1() and dis2() provide the body.

}

We can’t create the interface object. Because it contains abstract method.

Interface is use to provide to specification and class provide implementation for that interface.

Package :

Package is a collection of classes and interfaces which have same name but different purpose use.

Package is just like directory or folder which we are creating more than one classes or interfaces which have same name.

User defined package

Education

School College

Attendance Attendance

education.school education.college

Attendance.java Attendance.java

Access Specifiers

Java provided 4 types of access specifiers which help to expose the visibility or accessibility of

Variable, method and class from same package or other package.

private : we can use with instance variable, static variable, non static method, static method, constructor but not with local variable and class.

scope : within a same class.

default : we can use with all (no access specifiers means default access specifiers)

scope : within a same package

protected : we can use with instance variable, static variable, non static method, static method, constructor but not with local variable and class.

scope : within a same package other package if sub class.

public : we can use with instance variable, static variable, non static method, static method, constructor, class but not with local variable.

Scope : we can access in same package as well as other package