**Day 1 : 01-09-2022**

**Java Full stack**

Phase 1

Agile

Git

Core Java

Basic Programming

OOPs

Exception Handling

Multi threading : Overview

File handling

Collection Framework

Data Structure

Java 8 features

Phase 1 end project

Phase 2 : web application

Database – MySQL

Maven tool build

Jdbc

ORM : Hibernate tool

JEE : Java Enterprise Edition

Servlet

JSP

EJB

Phase 2 end project

Phase 3 : Spring Framework (backend technologies)

Spring core, context, mvc, dao, spring orm (jpa), spring rest, spring boot, spring micro service, spring data, into spring security etc.

Unit testing – Junit 5

Phase 3 end project.

Phase 4 : (frontend technologies)

Html

Css

Bootstrap

JavaScript

Adv Java

React JS

Phase 4 end project

Phase 5 : testing and deployment

TestNG

Selenium tool

Docker

CI and CD Jenkin

Kubernetes

AWS Overview

S3, EC2 instance etc

Phase 5 end project

Capstone project

Rest API 🡪JSON

Front end (React JS) backend (Spring boot) MySQL

Git : git is distributed sub version control which help to keep the track or record the execution of the application or program or software.

First create the folder : Repository

Then create the file inside a folder abc.txt

Then write some message inside abc.txt file

To make repository folder as git local repository

Please open terminal ie cmd, unix terminal or git bash terminal

git --version

git init : This command is use to make local repository as git local repository

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

git add filename this command is use to add the file to staging area.

git add . This command is use to add all files or folder present in current location in staging area.

git commit –m “commit message”; you get error message if you are doing first time.

Remote repository

Git hub

Git lab

Code commit

Etc

First time we need to set email id and name for git terminal

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

git config --global user.name “akash”

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

Whenever we created local repository by default git provided default branch it may be main or master.

To check default branch name we can run the command as

git branch

creating user defined branch

git branch branchname

git checkout branchname

git merge branchname : this command to add user defined branch changes in current branch

git branch –D branchName : this command is use to delete local branch

git branch -M main : this command is use to rename branch name

git remote add origin <https://github.com/Kaleakash/testapp.git>

git remote add origin <https://token@github.com/username/repository.git>

This command is use to link local repository with remote repository

git push -u origin main

git push –u origin branchname

git hub provided token base authentication to push the data in remote repository.

Generating token

Right icon click on setting

Then left panel select developer setting option

Personal access token

Token classic

Click on generate new token option



After click generate the token



git remote remove origin This command is use to remove remote repository

link from local repository.

**Day 2 : 01-10-2022**

**Git clone https://github.com/Kaleakash/vadafone\_java\_fullstack\_2023\_batch.git**

This command is use to download new remote repository in local machine.

git pull

this command is use to pull updated contents from remote repository in existing local repository.

You need to run this command inside repository folder.

git clone URL

git pull

git branch (main or master)

git branch branchName

git checkout branchname

do the changes in my notes if you need

then if you want to get new updated from remote repository

move to master or main branch using command

git checkout main/master

git pull (in main or master branch)

do the changes in user defined branch.

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

Nov 1995. Initial name of the java is Oak. Developed by James gosling and team.

Java was belong to sun micro system and part of oracle.

Version of java 1.0 java 18, etc

Open jdk

object : object any real world entity

property or state 🡪 have -🡪 variables or fields.

Person

Behavior -🡪do/does -🡪 function or methods

Bank

Animal

Car

Customer

Employee

Product

Object is a concept.

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

syntax of the class

class ClassName {

fields or variable;

methods or function;

}

class Car {

int wheel;

float price;

String color;

void start() { }

void appliedGear() { }

void moving() { }

void stop() { }

}

Class name must be follow pascal naming rules.

1. If class contains one word then first letter must be upper case.
2. If class contains more than one word each word first letter upper case.

Car, Employee, Customer, Manager, EmployeeDetails, CustomerDetails etc.

Variable and methods must be follow camel naming rules.

1. If variable or method name contains one word. Then we need to use in lower case.
2. If it contains more than one word then from 2nd word onward first letter upper case.

First open any editor and write the java program

class Demo {

public static void main(String args[]) {

System.out.println(“Welcome”);

}

}

Save the file with className.java

Open the terminal in that location where java program saved.

javac Demo.java to compile the program

java Demo to run the program

Eclipse is type of Open source IDE.

Eclipse IDE for JEE

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

2 types

1. Primitive data types : it is use to hold the value

8 types

1. byte 1
2. short 2
3. int 4
4. long 8 without decimal
5. float 4
6. double 8 with decimal

1. char 2 any single character
2. boolean 1 bit true or false value
3. Non primitive or reference data type : it is use to hold the value as well as reference of another data types.

array

class pre defined or user defined String, Date

interface pre defined or user defined :

enum pre defined or user defined

type casting : converting one data type to another data types is known as type casting.

2 types

1. Implicit type casting
2. Explicit type casting

int family

--------------------🡪implicit ---------------------🡪

byte short int long

🡨--------------explicit -------------------------------

int to float family : 4 types

int to float implicit

float to int explicit

Operator

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

Logical operator : &&, ||, !

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

Assignment operator : =

Increment and decrement operator : ++, --

instanceOf

If statement

if(condition){

}

If else

if(condition) {

}else {

}

If else if

If(condition) {

}else if(condition) {

}else if(condition) {

}else {

}

Switch statement

switch(variableName) { int family , char or string

case 1:block1;

break;

case 2:block2;

break;

default : defaultblock;

break;

}

Loop

While loop

Do while loop

For loop

Enhanced loop

Reference data types

array : array is a type of reference data type which is use to store move than one value of same type.

Syntax to declare array

datatype arrayName[];

int a;

int abc[]; declaration

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

syso(xyz);

get the value from array we need to use index position. Index position start from zero.

syso(xyz[0]);

for each loop

for(datatype variableName : arrayName) {

}

Creating memory size for array

datatype arrayName[]=new datatype[size];

int abc[]=new int[10];

syso(abc[0]); 0

syso(abc[9]); 0

default value

int family -0

float family –0.0

char space

boolean false

String null

**Day 3 : 01-11-2022**

**Java OOPs Concept**

object : object is any real world entity or object is instance of class.

Properties or state -🡪 have

Person

Behavior 🡪 do/does

Bank

Customer

Product

Mobile

Object is concept. To implement the object we need to take the one of the user defined data type ie class.

class : class is blue print of object or template of object or collection object which have same properties and behavior or user defined data type which help to describe the object.

class Car {

int wheel;

String color;

float price;

void start() {}

void appliedGear() {}

void moving() {}

void stop() {}

}

Object creation for user defined class

ClassName objectReferenceName = new ClassName();

To call method . if method belong to anther class

referenceName.methodName();

to access the value for instance variable through object

referenceName.variableName = value;

types of variable or fields

1. Instance variable
2. The variable which declared outside a method including main method 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, boolean false, string null.
4. Instance variable we can access in all method directly. But method must be part of same class and that method must non static.
5. Local variable
   1. The variable which declared inside a method including main method is known as local variable.
   2. The local variable doesn’t hold default. We have to initialize.
   3. The scope of the variable within that method where it declared.
6. Static variable

When local variable and instance variable have same name then local variable hide the visibility of instance variable. if we need to use instance variable inside method when local variable also have same then we can use this.instancevariable.

this is keyword which refer to current object or instance

constructor : Constructor is a type of special method which help to create the memory. While writing constructor we need to follow few points.

1. Constructor have same name as class itself.
2. Constructor doesn’t contains return type not even void also.
3. Constructor no need to call it will call automatically when we create the object.
4. If we not write explicitly any constructor java compiler provided default constructor. Default constructor always empty constructor.
5. If we write explicitly empty or parameterized constructor java compiler doesn’t provide default constructor

Method syntax

returnType methodName(parameterList) {

body for the method

}

void : void is a keyword. Which we can use with method and will doesn’t return any value.

We can write return type String, int, float, char, double etc.

Generally in java method must be part of class.

// no passing parameter and no return type.

void display() {

}

// passing parameter and no return type.

void add(int x, int y) {

int sum = x+y;

System.out.println(“sum ”+sum);

}

//passing parameter and return value

String sayHello(String name) {

// coding

return “Welcome user “+name;

}

boolean validate(String name, String password) {

// coding

return true; or return false

}

// no passing parameter but return value

int getNumber() {

return 100; // we can write code to generate Random number

}

Difference between constructor and method

In the life of the object if we want to perform any task only one time that type of task we can write empty or parameterized constructor.

In the life of the object if we want to perform the task more than one time that type of task we have to write inside methods.

Encapsulation : it use to bind or wrap data (variables) and code (function or methods) in a single unit is known as Encapsulation.

Example : class

If variable are private we can’t assign the value for those variable object.

Inheritance :

Inheritance is use to inherits or acquire properties and behavior of old class to new class.

extends is keyword which help to achieve inheritance.

class OldClass { // super class or base class or parent class

properties

behavior

}

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

properties

behavior

}

With help of new class object we can access own properties and behavior as well as old class properties and behavior.

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 and n number of sub class directly connected 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 this type of inheritance using class. We can achieve this type of inheritance using interface.

OOPs relationship

is a relationship

has a relationship

Manager / Developer is a Employee

Employee has a Address

class Employee { // super class must be generic

id,name,salary

Address add = new Address();

}

class Manager extends Employee{// sub class must be specific

numberOfEmp;

}

class Developer extends Employee{

projectName

}

class ProjectManager extends Manager{

clientId;

}

class Address {

city and state

}

**Day 4 : 01-12-2022**

Has a relationship

3 types

1. Association
2. Aggregation
3. Composition

If we want to achieve has a relationship inside one class we have to create another class object.

class A {

B obj1 = new B(); zero or 1 or many

B obj2 = new B();

}

class B {

A obj1 = new A(); zero or 1 or many

A obj2 = new A();

}

class Manager {

Address padd =new Address(); 1 or many

Address ladd = new Address();

}

class Address {

}

It is a type of has relationship ie association. But this association is known as weak association. Weak association is known as aggregation.

class Student {

StudentHistory sh = new StudentHistory();

}

class StudentHistory {

}

It is also type of association. This association is known as strong association. String association is known as composition.

Taking the value through keyboards in Java

1. Using Scanner class.

Scanner is pre defined class part of util package. Package is a collection of classes and interfaces.

Syntax to create Scanner class object

Scanner obj = new Scanner(System.in);

import java.util.Scanner;

Polymorphism

One name many forms or many implementation.

2 types

1. Compile time

Static binding or early binding

Ex : Method overloading : The method have same name but different parameter list (type of parameter list or number of parameter list must be different).

class Operation {

void add(int x, int y) {

System.out.println(x+y);

}

void add(int x, int y, int z) {

System.out.println(x+y+z);

}

void add(float x, float y) {

System.out.println(x+y);

}

void add(String x, String y) {

System.out.println(x+y);

}

}

1. Run time

Dynamic binding or late binding

Ex : Method overriding : method have same name and same method signature ie number of parameter list, type of parameter list and return type must be same. To achieve method override we need inheritance.

Annotation : meta – data . data about data. Java provided lot of pre defined annotation. All annotation start with pre fix @ followed by name of the annotation.

Few annotation we can use on class level or method level or property level.

Core java provided pre defined annotation @Override. This annotation we can use in sub class method if that method override super class method.

Non access specifiers keywords

abstract, static and final keywords

abstract : abstract is keyword which we can use with method and class but not with variable.

1. abstract method : method without body or without curly braces or incomplete method is known as abstract method.

abstract returnType methodName();

abstract void speed();

1. abstract class : if class contains one or more abstract method then we have to declare the class as abstract class.

abstract class Bike {

}

1. whichever class extends abstract class that class must be provide the body for all abstract method belong to that class mandatory. Sub class can ignore if that class itself is abstract class.
2. Abstract class we can’t create the object.
3. Abstract class can contains normal as well as abstract method. Means it can contains zero or 1 or many abstract method.
4. Abstract class without abstract method is use to restrict the user not allow to create the object.

abstract class Bike {

abstract void speed();

abstract void mailage();

public void color() {

}

}

class Honda extends Bike {

}

static keyword

1. static keyword we can use with variable and method but not with class. (if class is inner class then we can static keyword for inner class but not for outer class).
2. static variable : if variable is static we can assign the value for that variable with help of class name. we can assign the value for static variable through object.
3. static method : if method is static we can call that method with help of class name as well as through object.
4. inside static method we can access only static variable directly but inside non static method we can access both the type of variable static as well as non static directly.

Every class we will get only one static memory.



**Day 5: 01-13-2022**

final keyword

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

Syntax final datatype variableName = value;

final int A=10;

final variable can be static as well as non static.

1. final method : if method is final we can’t override that method but sub class inherits that method.
2. final class : if class is final we can’t inherits that class. We can create object of that class.

interface : interface is a type of reference data types also known as 100% pure abstract class till java 7.

Syntax to declare the interface

interface interfaceName {

fields;

methods;

}

By default all variables in interface are public static and final

By default all methods in interface are public and abstract.

interface Abc {

public static final int A=10;

static final int B=20;

final int C=30;

int D=40; by default consider

public abstract void dis1();

abstract void dis2();

void dis3(); by default consider

}

interface Abc { super interface

int A=10;

void dis1();

}

interface Xyz { super interface

int B=20;

void dis2();

}

interface Mno extends Abc,Xyz{ sub interface.

int C=30;

void dis3();

}

Like a class one interface can extends another interface. But interface can extends more than one interface but class can’t.

class Test implements Abc,Xyz { // sub class

must be provide the body for dis1 and dis2 methods.

}

Class always implements interface and class can implements more than one interface. Whichever class implements one or more interface that class must be provide the body for all abstract method belong to that interface.

While override method with access specifiers rules

Super class / interface sub class

public public

protected public

protected

default(nothing) public

protected

nothing

private we can’t override

Difference between interface and abstract class.

1. Abstract class can contains normal as well as abstract method. But interface contains only abstract method.
2. Using abstract class we can achieve partial abstraction but using interface we can achieve full abstraction.
3. Abstract class can contains final as well as normal variable but interface contains only final variable.
4. Normal class can extends only one abstract class but normal class can implements more than one interface.
5. Abstract class can contains default as well as parameterized constructor. But interface doesn’t contains default as well as we can’t write parametrized constructor.

Abstraction : hiding the internal implements without knowing background details.

Common between interface and abstract class.

1. We can’t create the object of interface as well as abstract class.
2. Whichever class extends or implements abstract class or interface that class must be provide the body for all abstract method which belong to that interface or abstract class.

Runtime polymorphism using object creation.

Super(), super keyword and this keyword

By default every sub class constructor contains super() which help to call super class empty constructor.

Package : package is a collection of classes and interface. When we need two or more than one class or interface which have same name but functionality wise different purpose. Package is like a folder or directory when two classes and interface which have same name but different purpose use.

Package mainly divided into 2 types

1 user defined package

2. pre defined or built in package

User defined package

education

school college

Attendance.java Attendance.java

Date

java.util.Date;

java.sql.Date;

**Week 2**

**Day 6 : 01-09-2022**

Access Specifiers : Access Specifiers are use to provide the access visibility of variables, methods and classes within a same package or other package.

private : we can use private access specifiers with instance variable, static variable non static method, static method, constructor but we can’t use with local variable and class.

scope : private scope with a same class.

default (nothing) : we can use with all.

scope : with in a same package.

protected : we can use protected access specifiers with instance variable, static variable non static method, static method, constructor but we can’t use with local variable and class.

scope : with in a same package other package if it is sub class.

public : we can use public access specifiers with instance variable, static variable non static method, static method, constructor, class but we can’t use with local variable.

scope : We can access same package as well other package.

Pre defined or built in package

java javax 🡪root package

lang language swing

io input out sql

util utility servlet

sql structured query language ejb enterprise java bean

awt abstract window toolkit jms java messaging service

etc etc

by default in every java program lang package imported. So while using any classes and interfaces part of lang package we no need to import.

By default every java program extends Object class. In Java Object is pre defined class pat of lang package and by default every class it may be pre defined or user defined extends Object class.

Exception Handling :

Exception is an object which created or generated when unexpected thing or abnormal condition occurs during the execution of programs is known as a exception. To handle using some technique is known as exception handling.

Java

Compile the program run the program

javac java

compile time error run time error

syntax error

or typo error

Run time error

Error Exception

Error :The error which generated at run time which we can’t handle it. JVM Crash, Software or hardware issue, out of memory.

Exception : it is a type of error which generated at run time which we can handle it.

Divided by zero.

In Java Error and Exception both are pre defined class part of lang package.

Object

Throwable

Error Exception

Checked Exception UnChecked Exception

RuntimeException

IOException

SQLException ArithmeticException

NumberFormatException

ArrayIndexOutOfBoundsException

To handle both type of exception java provided 5 keywords.

1. try
2. catch
3. finally
4. throw
5. throws

try catch block

try {

}catch(Exception e) {

}

We have to use try with single catch block.

1. If any exception generate we want to display error message.
2. If we don’t known name of exception.

Try with multiple catch block

try {

}catch(ArithmeticException e){

}catch(ArrayIndexOutOfBoundsException e) {

}

Try with multiple catch block.

Base up on specific exception we want to do specific task.

Integer is pre defined class contains parseInt() static method which help to convert string to integer.

String str = “10a”;

System.out.println(str+10); 1010

System.out.prinitln(Integer.parseInt(str)+10); 20

finally block : finally is a type of block which will execute compulsory 100% sure if any exception generate or not.

try block : the code one line or more than one line which generate execute we have to write inside a try block.

catch block : this block execute only if any exception generate. No exception no catch block.

But finally block execute compulsory if any exception generate or not.

try

catch catch catch catch finally

catch finally catch

catch finally

throw : this keyword is use to throw the exception explicitly with condition. It can be pre defined exception or user defined. It is use to raise the exception explicitly.

throw new Exception();

or

throw new ExceptionSubClass();

throws throws keyword is use to throw the exception to caller method. Throws keyword we use with method signature.

void display() throws Exception,ExceptionSubClass {

}

Checked exception check at compile time as well as run time.

Unchecked exception we can ignore some extend. But checked exception we need to handle it mandatory using try – catch or throws. Otherwise we can’t compile the program.

try{

open the file

do read and write operation

}catch(Exception e) {

}finally {

close the file

}

Finally block is use to close the resources properly.

Week 2 : Day 7 : 01-17-2022

Multi threading :

Program : set of instruction to perform a specific task is known as program.

Processor : after written the program we need to give the program to processor to process or execute it. Processor is responsible to execute the code.

Process : time taken to execute the code or program in execution.

Thread : thread is small execution of a code within a process. Thread is a part of process.

Process is heavy weighted. It takes more memory or more resource to do the task. Thread is light weighted it takes less memory or less resources to do the same task.

In Java inside a main by default. Default thread execute. To check that default thread details.

Java provided pre defined class with name as Thread. It is part of lang package.

Thread.currentThread(); currentThread is static method and this method return type is thread class reference.

Thread t = Thread.currentThread();

System.out.println(t); Thread[main,5,main];

main 🡪 name of the thread t.getName(), t.setName(“MyThread”);

5 🡪 priority of the thread t.setPriority(1) min 1 max 10 norm 5

Main 🡪 group of the thread. t.getThreadGroup();

Java by default thread base programming language.

Multi tasking :

Process based

Pr1

Pr2

Pr3

Thread based

In Java we can achieve multi tasking using thread base.

In Java we can create more than one thread using

1. Extends Thread class
   1. Create user defined class and that class must be extends Thread class.
   2. Then create the Thread class reference.
   3. With thread referenced we need to call start() method.
   4. Start method internally call run method part of Thread. That run method with empty implementation.
   5. If we want to do custom implementation then we need to override run method.
2. Implements Runnable interface.
   1. Create user defined class and implements Runnable interface.
   2. Runnable is an interface which contains one abstract method ie run().
   3. So when class implements Runnable interface we need to override run method mandatory.