Phase 2 or course 2

Day 1 04-08-2023

Java : Java is an open source and pure object oriented programming language.

Version of Java

Java 1.0 Java 18/19

Java

J2SE J2EE J2ME

JavaSE JavaEE JavaME

Standard Edition Enterprise Edition Micro edition

Java basic programming

OOPs

Exception handling

Collection Framework

JDBC

Open source IDE : Eclipse,

OOPs : Object Oriented Programming system

object : object is any real world entity.

Property or state -🡪 have -🡪variable/ fields

Person

Behaviour -🡪do/does -🡪 functions / methods

Place

Bank

Animal

Car

Customer

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

Syntax of class.

class className {

fields;

methods;

}

package is a collection of classes and interfaces.

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

2 types

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

8 types of primitive

1. byte 1byte
2. short 2
3. int 4
4. long 8 : without decimal number
5. float 4
6. double 8 : with decimal
7. char 2 : single character
8. boolean 1 bit : true or false value.
9. Non primitive or reference data types : it is uses to store value as well as reference of another data types

4 types

1. array
2. class
3. interface
4. enum

array : array is known a reference data types which is use to store more than one value of same types.

datatype arrayName[]; array declaration

int num[];

int num1[]={10,20,30,40,50,60}; array declaration with initialization

array start with index position zero.

System.out.println(num1[0]);

System.out.println(num1[1]);

System.out.println(num1[2]);

Using loop we can access those values one by one

While loop

Do while loop

For loop

For each loop or enhanced loop

Creating the memory size for the array

datatype arrayname[]=new datatype[size];

In Java we can take the value through keyboard lot of way

1. using Scanner class : Scanner is a pre defined class which contains lot of pre defined methods which help to take the value through keyboards.

Syntax to create the Scanner class object.

Scanner obj = new Scanner(System.in);

Scanner class is a part of util package.

java and javax are known as root package.

By default every java program import lang package.

Means we can use all classes and interfaces part of lang package without importing.

In java String is a pre defined class part of lang package.

Car

Property , wheel, color, amount etc

8 primitive + string reference data type we use to

Declare the variable.

Behaviour

We can write behaviour using function or methods in java

returnType methodName(parameterList){

}

void info() { info is method name and this method

no return type

}

String sayHello() { sayHello method name and it return string

Value

return “Welcome”;

}

void add(int x, int y) { add method passing two int value

}

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

Pts.

1. Constructor have same name as class itself.
2. Class doesn’t contains return type not even void also
3. Constructor no need to call it will call automatically when we create the object of that class.

this is a keyword which refer to current object.

When local variable or parameter variable and instance variable have same name

To refer to instance variable we use this keyword.

In the life of the object if we want to perform any task only one time

That type of task we have to write inside a empty or parameter constructor.

In the life of the object if we want to perform any task more than one time

That type of task we have to write inside a method.

Encapsulation : binding or wrapping data(variable) and code(methods) in a single unit is known as

Encapsulation.

By default every java class internally follow encapsulation.

class Employee {

variable and methods.

}

JavaBean class Vs Normal class

1. JavaBean class must be public not normal class not mandatory.
2. All variable present inside a JavaBean class must be private but in normal class may or may not.
3. In JavaBean class for each variable we need to provide setter and getter method.
4. Setter method to set the value with condition if we need.
5. Getter method is use to get the value.
6. But in normal class if we write private variable we can provide helper method with name doesn’t matter.

public class Employee {

private int id;

private String name;

private float salary;

public void setId(int id) {

this.id = id;

}

public int getId() {

return this.id;

}

}

Angular 🡪 Model class

Java 🡪 JavaBean class

Database 🡪 Table

Day 2 04-08-2023

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

class OldClass { super class or base class or parent class

properties

behaviour

}

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

properties

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 n number of sub class connected directly to super class

Class A { }

Class B extends A{ }

Class C extends A{ }

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

Class A { }

Class B { }

Class C extends A,B { } But java doesn’t support this type of inheritance.

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

Oops relationship

1. Is a relationship
2. Has a relationship

class Employee {

id,name,salary

}

class Manager extends Employee {

numberOfEmp

Address add = new Address();

}

class Developer extends Employee {

projectName;

}

class ProjectManager extends Manager{

clientInfo;

}

class Address {

city and state

}

Manager is a Employee

Developer is a Employee

ProjectManager is a Manager

Has a relationship

3 types

Association

1. Aggregation
2. Composition

class A { zero or 1 or many

B obj1 = new B();

}

class B { zero or 1 or many

A obj2 = new A();

}

1-0, 0-1, many-0,many-1, 0-0

Core GUI AWT or Swing or JavaFX

Standalone :

Web Application html, css , servlet and jsp

Android : mobile application

MEAN / MERN

Spring framework and spring boot.

class Manager {

Address padd = new Address(;

Address oadd = new Address();

}

class Address {

}

It is a type of association but it weak association. Weak association is known as aggregation.

Class Student {

StudentHistory sh = new StudentHistory();

}

Class StudentHistory {

}

It is a also type of association but is known as strong association. Strong association is known as composition.

class Trainer {

array of students;

}

class Student {

}

Polymorphism : One name many forms or many implementation.

2 types

1 compile time or static binding or early binding

Method Overloading :

The method have same name but different parameter list or type of parameter list

2 run time or dynamic binding or late binding

Method Overriding :

The 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 inheritance.

Non access specifiers

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

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

abstract void speed();

1. abstract class :

abstract class className {

}

1. if class contains abstract method then that class must be declare as an abstract.
2. Whichever class extends abstract class that class must be provide the body for all abstract method belong to that class. That class can ignore if that class itself is abstract class.
3. Abstract class we can’t create object.
4. Abstract class can contains normal as well as abstract method ie zero or 1 or all abstract method.

static

1. Static keyword we can use with variable and method but not with class.
2. Static variable : if variable is static we can assig the value for that variable with help of class as well as object.
3. Static method :if method is static we can call that method with help of class name as well as object.
4. Inside non static method we can access both static as well as non static variable directly.
5. Inside static method we can access only static variable directly we can’t access non static variable directly.

Every class we will get only one static memory.

For that class how many object we will create that many heap memory get created.

final : final keyword we can use with variable, method and class.

1. Final variable : to declare constant value in java we use final keyword.

final float PI=3.12f;

1. Final method : if method is final we can’t override that method.
2. Final class : if class is final we can’t inherits or extends that class.

Annotation : annotation is like a decorator in angular. Annotation or decorator is meta-data. Meta-data means data about data. Like angular java also provided lot of pre defined annotation. All annotation start with @ followed by annotation name.

Abstraction : hiding the internal implementation without knowing background details.

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

Using abstract class we can achieve partial abstraction but using interface we can achieve 100 abstraction.

Syntax to create the interface in java

Interface interfaceName{

Fields;

Methods;

}

In interface all variable by default are public static and final

In interface all methods are by default public and abstract.

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

Class always implements interface. Class can implements more than one interface at time.

Whichever class implements any interface that class must be provide the body for all methods

Belongs that interface.

Access specifiers :in java we can use 4 types of access specifiers which hep to expose visibility of variable, method and class.

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

Scope : private within a same class.

1. default (nothing): we can use with all

scope : within a same package.

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

Scope : within in a same package as well as other package if it is sub class

1. public : we can use with instance variable, static variable, non static method, static method, constructor and class but local variable.

Scope : same package as well as other package.