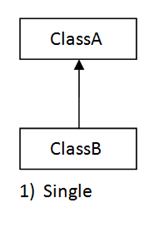
**Types of Inheritance**

**There the five types of inheritance as below**

1. Single inheritance
2. Multiple inheritance
3. Multilevel inheritance
4. Hybrid inheritance
5. Hierarchical inheritance
6. Simple or Single inheritance



It is the basic type of inheritance where a child class extends to a parent class.

It is simply, One child & One parent relationship.

Example:

**package** com.inheritance;

**public** **class** Insurance {

**void** getInsuranceDetails() {

System.***out***.println("this is insurance details..");

}

}

**package** com.inheritance;

**public** **class** HealthInsurance **extends** Insurance {

**void** getHealthInsuranceDetails() {

System.***out***.println("this is health insurance details.");

}

}

**package** com.inheritance;

**public** **class** TestMain {

**public** **static** **void** main(String[] args) {

HealthInsurance healthInsurance = **new** HealthInsurance();

healthInsurance.getInsuranceDetails();

healthInsurance.getHealthInsuranceDetails();

}

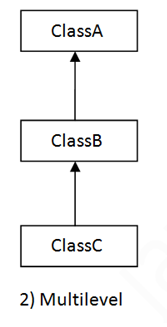
}

Output

this is insurance details..

this is health insurance details.

1. Multilevel inheritance



It has only one base class and multiple derived class. It refers to the concept of one class extending (Or inherits) more than one base class.

Multilevel Inheritance is when a superclass is inherited by an intermediate class, which is then inherited by a derived class, forming 3 or more levels of inheritance.

As we can see in the above picture, ClassA is a superclass and ClassB is an intermediate class which inherits ClassA. And ClassC is a derived class which inherits ClassB. It forms 3 levels. That’s why it is multi-level inheritance.

**package** com.inheritance;

**public** **class** Account {

**void** getAccountDetails() {

System.***out***.println("this is account details..");

}

}

**package** com.inheritance;

**public** **class** CurrentAccount **extends** Account {

**void** getCurrentAccountDetails() {

System.***out***.println("this is current account details");

}

}

**package** com.inheritance;

**public** **class** SavingAccount **extends** CurrentAccount {

**void** getSavingAccountDetails() {

System.***out***.println("this is saving account details");

}

}

**package** com.inheritance;

**public** **class** TestMain {

**public** **static** **void** main(String[] args) {

SavingAccount savingAccount = **new** SavingAccount();

savingAccount.getAccountDetails();

savingAccount.getCurrentAccountDetails();

savingAccount.getSavingAccountDetails();

}

}

Output

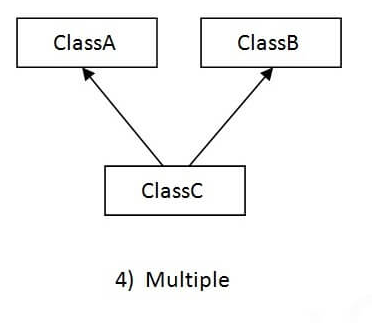
this is account details..

this is current account details

this is saving account details

1. Multiple inheritance

* When a single subclass inherits from multiple parent classes is known as Multiple inheritance.
* Multiple inheritance is not possible in Java.
* When a subclass tries to extends multiple superclasses, we will get compile error.



Why multiple inheritance not supported in java in case of classes?

Class base has test () method and class derived has also test () method. Class test extends Base, Derived, which test method It will called, so it create the ambiguity so that’s why multiple inheritance does not supports in java.

**package** com.multiple.inheritance;

**public** **class** A {

**void** m1() {

}

}

**package** com.multiple.inheritance;

**public** **class** B {

**void** m1() {

}

}

**package** com.multiple.inheritance;

**class** C **extends** A,B {

**public** **static** **void** main(String[] args) {

C c= **new** C();

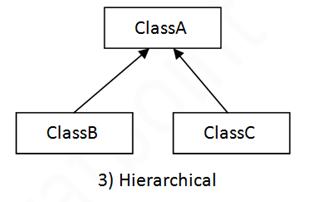
c.m1();

}

}

Note- it will get the compile time error.

1. Hierarchical inheritance



* One class is inherited by many sub classes called as.
* As we can see in the above picture, ClassA is the super class and ClassB and

ClassC are two sub classes which are inheriting ClassA.

**package** com.inheritance;

**public** **class** Loan {

**void** getLoanDetails() {

System.***out***.println("this is loan details");

}

}

**package** com.inheritance;

**public** **class** HomeLoan **extends** Loan {

**void** getHomeLoanDetails() {

System.***out***.println("this is home loan details..");

}

}

**package** com.inheritance;

**public** **class** PersonalLoan **extends** Loan {

**void** getPersonalLoanDetails() {

System.***out***.println("this is personal loan details");

}

}

**package** com.inheritance;

**public** **class** CarLoan **extends** Loan {

**void** getCarLoanDetails() {

System.***out***.println("this is car loan details.");

}

}

**package** com.inheritance;

**public** **class** TestMain {

**public** **static** **void** main(String[] args) {

HomeLoan homeLoan = **new** HomeLoan();

CarLoan carLoan = **new** CarLoan();

PersonalLoan personalLoan = **new** PersonalLoan();

homeLoan.getHomeLoanDetails();

carLoan.getCarLoanDetails();

personalLoan.getPersonalLoanDetails();

}

}

Output

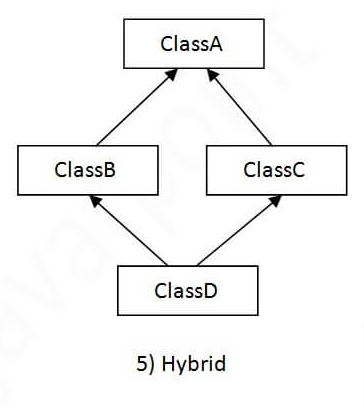
this is home loan details..

this is car loan details.

this is personal loan details

1. Hybrid inheritance

* Hybrid inheritance, a mix of two or more of the above kinds of inheritance.
* This type of inheritance is also not possible as it contains multiple inheritance which is not possible in java.



**Aggregation (Has-A relationship)**

If a class has an entity reference, it is known as Aggregation.

It represents Has-A relationship.

Example: Employee Has-A Address.

package com.test;

public class Employee {

private int id;

private String firstName;

private String lastName;

private String mobileNumber;

private Address address;

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getMobileNumber() {

return mobileNumber;

}

public void setMobileNumber(String mobileNumber) {

this.mobileNumber = mobileNumber;

}

public Address getAddress() {

return address;

}

public void setAddress(Address address) {

this.address = address;

}

// getter and setter

}

package com.test;

public class Address {

private String streetNo;

private String city;

private String state;

private String country;

public String getStreetNo() {

return streetNo;

}

public void setStreetNo(String streetNo) {

this.streetNo = streetNo;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

}

package com.test;

import java.util.Scanner;

public class Test {

public void getUserDetails() {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter employee id>>");

int id = scanner.nextInt();

System.out.println("Enter employee first name>>");

String firstName = scanner.next();

System.out.println("Enter employee last name>>");

String lastName = scanner.next();

System.out.println("Enter employee mobile number>>");

String mobileNumber = scanner.next();

System.out.println("Enter street no>>");

String streetNo = scanner.next();

System.out.println("Enter city>>");

String city = scanner.next();

System.out.println("Enter state>>");

String state = scanner.next();

System.out.println("Enter country>>");

String country = scanner.next();

// set the value into employee object here

Employee employee = new Employee();

employee.setId(id);

employee.setFirstName(firstName);

employee.setLastName(lastName);

employee.setMobileNumber(mobileNumber);

// set value into address object here

Address address = new Address();

address.setStreetNo(streetNo);

address.setCity(city);

address.setState(state);

address.setCountry(country);

//set address object into employee object

employee.setAddress(address);

// get the value from employee object here

System.out.println("Employee ID>>" + employee.getId());

System.out.println("Employee First Name>>" +

employee.getFirstName());

System.out.println("Employee Last Name>>" +

employee.getLastName());

System.out.println("Employee Mobile Number>>" +

employee.getMobileNumber());

System.out.println("Employee Street No>>" +

employee.getAddress().getStreetNo());

System.out.println("Employee City>>" +

employee.getAddress().getCity());

System.out.println("Employee State>>" +

employee.getAddress().getState());

System.out.println("Employee Country>>" +

employee.getAddress().getCountry());

}

public static void main(String[] args) {

Test test = new Test();

test.getUserDetails();

}

}

Output

Enter employee id>>

10

Enter employee first name>>

Ajay

Enter employee last name>>

pawar

Enter employee mobile number>>

8595958575

Enter street no>>