1. create multilevel inheritance for

//Vehicle

//Four\_wheeler

//Petrol\_Four\_Wheeler

//FiveSeater\_Petrol\_Four\_Wheeler

//Baleno\_FiveSeater\_Petrol\_Four\_Wheeler

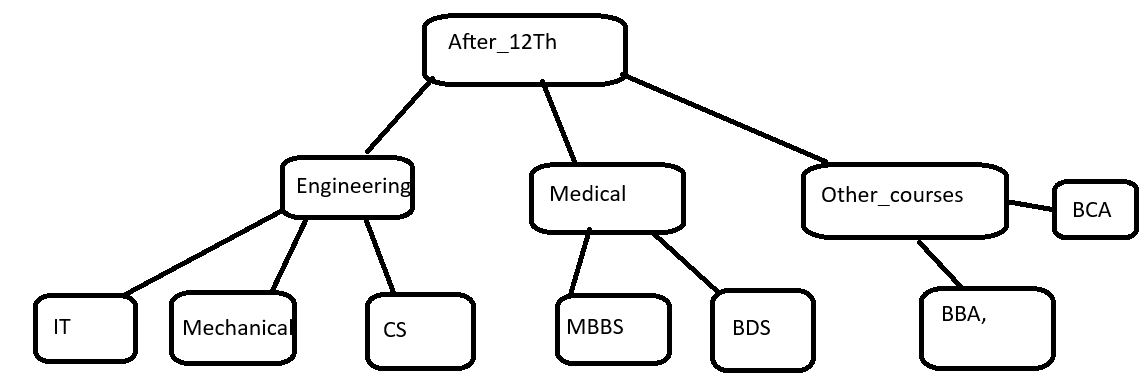
1. class Vehicle {
2. void showVehicleType() {
3. System.***out***.println("This is a vehicle.");
4. }
5. }
6. //Child class
7. class FourWheeler extends Vehicle {
8. void showWheelerType() {
9. System.***out***.println("This is a four wheeler.");
10. }
11. }
12. //Subchild class
13. class PetrolFourWheeler extends FourWheeler {
14. void showFuelType() {
15. System.***out***.println("This four wheeler uses petrol.");
16. }
17. }
18. //Another subchild class
19. class FiveSeaterPetrolFourWheeler extends PetrolFourWheeler {
20. void showSeatingCapacity() {
21. System.***out***.println("This petrol four wheeler has 5 seats.");
22. }
23. }
24. //Final subchild class
25. class Baleno extends FiveSeaterPetrolFourWheeler {
26. void showCarModel() {
27. System.***out***.println("This is a Baleno - 5 seater petrol four wheeler.");
28. }
29. }
30. public class Vehicle\_multilevel {
31. public static void main(String[] args) {
32. Baleno myCar = new Baleno();
33. myCar.showVehicleType();
34. myCar.showWheelerType();
35. myCar.showFuelType();
36. myCar.showSeatingCapacity();
37. myCar.showCarModel();
38. }
39. }

2.Demonstrate the use of the super keyword

The super keyword in Java is used to access a parent class methods, variables, or constructor. It helps when the child class wants to reuse or extend the functionality of the parent class.

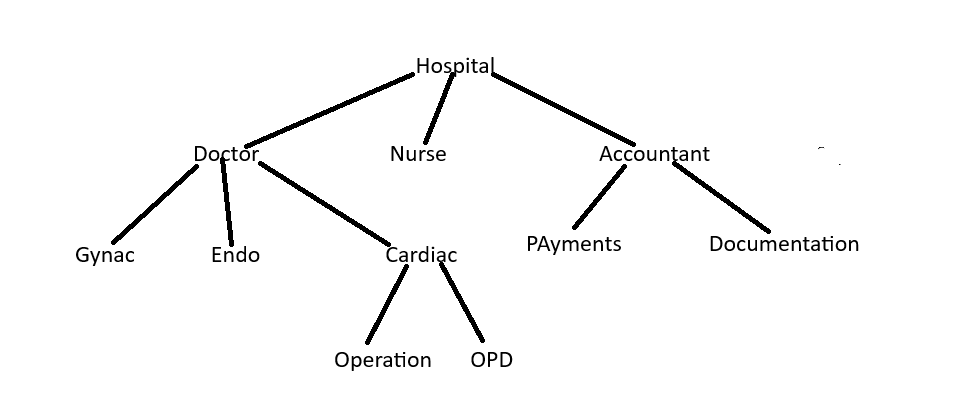
1. Create Hospital super class and access this class inside the patient child class and access properties from Hospital class.
2. //Superclass
3. class Hospital {
4. String hospitalName = "City Hospital";
5. String location = "Chennai";
6. void showHospitalDetails() {
7. System.***out***.println("Hospital: " + hospitalName);
8. System.***out***.println("Location: " + location);
9. }
10. }
11. //Subclass
12. class Patient extends Hospital {
13. String patientName = "Yash";
14. void showPatientDetails() {
15. System.***out***.println("Patient Name: " + patientName);
16. }
17. }
18. public class Hospital\_Patient {
19. public static void main(String[] args) {
20. Patient p = new Patient();
21. p.showPatientDetails();
22. p.showHospitalDetails();
23. }
24. }

3.Create Hierarchical inheritance



1. package Assignment4;
2. //Superclass
3. class After12th {
4. void show() {
5. System.***out***.println("Available career options after 12th:");
6. }
7. }
8. //Child class 1
9. class Engineering extends After12th {
10. void engCourses() {
11. System.***out***.println("Engineering courses: IT, Mechanical, CS");
12. }
13. }
14. //Child class 2
15. class Medical extends After12th {
16. void medCourses() {
17. System.***out***.println("Medical courses: MBBS, BDS");
18. }
19. }
20. //Child class 3
21. class OtherCourses extends After12th {
22. void otherCourses() {
23. System.***out***.println("Other courses: BBA, BCA");
24. }
25. }
26. public class Carrer\_option {
27. public static void main(String[] args) {
28. Engineering eng = new Engineering();
29. Medical med = new Medical();
30. OtherCourses other = new OtherCourses();
31. eng.show();
32. eng.engCourses();
33. med.medCourses();
34. other.otherCourses();
35. }
36. }

4. Create practice on this



package oops;

class Hospital {

void displayHospital() {

System.***out***.println("Welcome to the Hospital.");

}

}

class Doctor extends Hospital {

void displayDoctorDept() {

System.***out***.println("Doctor Department:");

}

}

class Gynac extends Doctor {

void gynacDoctor() {

System.***out***.println("- Gynac Specialist");

}

}

class Endo extends Doctor {

void endoDoctor() {

System.***out***.println("- Endo Specialist");

}

}

class Cardiac extends Doctor {

void cardiacDoctor() {

System.***out***.println("- Cardiac Specialist");

}

void cardiacOperation() {

System.***out***.println(" - Operation Section");

}

void cardiacOPD() {

System.***out***.println(" - OPD Section");

}

}

class Nurse extends Hospital {

void displayNurseDept() {

System.***out***.println("Nursing Department");

}

}

class Accountant extends Hospital {

void displayAccountDept() {

System.***out***.println("Accountant Department:");

}

void paymentsSection() {

System.***out***.println("- Payments Section");

}

void documentationSection() {

System.***out***.println("- Documentation Section");

}

}

public class HospitalStructure {

public static void main(String[] args) {

System.***out***.println("=== Hospital Hierarchy ===");

Gynac gynac = new Gynac();

gynac.displayHospital();

gynac.displayDoctorDept();

gynac.gynacDoctor();

Endo endo = new Endo();

endo.endoDoctor();

Cardiac cardiac = new Cardiac();

cardiac.cardiacDoctor();

cardiac.cardiacOperation();

cardiac.cardiacOPD();

Nurse nurse = new Nurse();

nurse.displayNurseDept();

Accountant accountant = new Accountant();

accountant.displayAccountDept();

accountant.paymentsSection();

accountant.documentationSection();

}

}

5.Polymorphism

1. Create a class Calculator with the following overloaded add()

1.add(int a, int b)

2.add(int a, int b, int c)

3.add(double a, double b)

package Assignment4;

class Calculator {

void add(int a, int b) {

System.***out***.println("Sum of 2 integers: " + (a + b));

}

void add(int a, int b, int c) {

System.***out***.println("Sum of 3 integers: " + (a + b + c));

}

void add(double a, double b) {

System.***out***.println("Sum of 2 doubles: " + (a + b));

}

}

public class calculator\_main {

public static void main(String[] args) {

Calculator calc = new Calculator();

calc.add(5, 10);

calc.add(3, 6, 9);

calc.add(2.5, 4.5);

}

}

1. Create a base class Shape with a method area() that prints a message. Then create two subclasses Circleàoverride area() to calculator and print area of circle Rectangleà override area() to calculate and print area of a rectangle
2. package Assignment4;
3. class Shape {
4. void area() {
5. System.***out***.println("Calculating area in base class...");
6. }
7. }
8. class Circle extends Shape {
9. void area() {
10. double radius = 5.0;
11. double area = 3.14 \* radius \* radius;
12. System.***out***.println("Area of Circle: " + area);
13. }
14. }
15. class Rectangle extends Shape {
16. void area() {
17. int length = 10;
18. int breadth = 5;
19. int area = length \* breadth;
20. System.***out***.println("Area of Rectangle: " + area);
21. }
22. }
23. public class circle\_area {
24. public static void main(String[] args) {
25. Shape s;
26. s = new Circle();
27. s.area();
28. s = new Rectangle();
29. s.area();
30. }
31. }

3.Create a Bank class with a method getInterestRate() create subclasses: SBIàreturn 6.7% ICICIàreturn 7.0% HDFCàreturn 7.5%

package Assignment4;

class Bank {

double getInterestRate() {

return 0.0;

}

}

class SBI extends Bank {

double getInterestRate() {

return 6.7;

}

}

class ICICI extends Bank {

double getInterestRate() {

return 7.0;

}

}

class HDFC extends Bank {

double getInterestRate() {

return 7.5;

}

}

public class Bank\_interest {

public static void main(String[] args) {

Bank b;

b = new SBI();

System.***out***.println("SBI Interest Rate: " + b.getInterestRate() + "%");

b = new ICICI();

System.***out***.println("ICICI Interest Rate: " + b.getInterestRate() + "%");

b = new HDFC();

System.***out***.println("HDFC Interest Rate: " + b.getInterestRate() + "%");

}

}

4.Runtime Polymorphism with constructor Chaining create a class vehicle with a constructor that prints “Vehicle Created”

Create a subclass Bike that override a method and uses super() in constructor

Combined question

Create an abstract class SmartDevice with methods like turnOn(), turnOff(), and performFunction().  
Create child classes:

* SmartPhone: performs calling and browsing.
* SmartWatch: tracks fitness and time.
* SmartSpeaker: plays music and responds to voice commands.
* Write code to store all objects in an array and use polymorphism to invoke their performFunction().
* package Assignment4;
* class BasicVehicle {
* BasicVehicle() {
* System.***out***.println("Vehicle Created");
* }
* void start() {
* System.***out***.println("Vehicle is starting...");

}

* }
* class Bike extends BasicVehicle {
* Bike() {
* super();
* System.***out***.println("Bike Created");
* }
* void start() {
* System.***out***.println("Bike is starting...");
* }
* }
* abstract class SmartDevice {
* abstract void turnOn();
* abstract void turnOff();
* abstract void performFunction();
* }
* class SmartPhone extends SmartDevice {
* void turnOn() {
* System.***out***.println("SmartPhone is ON");
* }
* void turnOff() {
* System.***out***.println("SmartPhone is OFF");
* }
* void performFunction() {
* System.***out***.println("SmartPhone is making calls and browsing");
* }
* }
* class SmartWatch extends SmartDevice {
* void turnOn() {
* System.***out***.println("SmartWatch is ON");
* }
* void turnOff() {
* System.***out***.println("SmartWatch is OFF");
* }
* void performFunction() {
* System.***out***.println("SmartWatch is tracking fitness and showing time");
* }
* }
* class SmartSpeaker extends SmartDevice {
* void turnOn() {
* System.***out***.println("SmartSpeaker is ON");
* }
* void turnOff() {
* System.***out***.println("SmartSpeaker is OFF");
* }
* void performFunction() {
* System.***out***.println("SmartSpeaker is playing music and responding to voice commands");
* }
* }
* public class Bike\_main {
* public static void main(String[] args) {
* BasicVehicle v = new Bike();
* v.start();
* System.***out***.println("\n--- Smart Devices in Action ---");
* SmartDevice[] devices = {
* new SmartPhone(),
* new SmartWatch(),
* new SmartSpeaker()
* };
* for (SmartDevice device : devices) {
* device.turnOn();
* device.performFunction();
* device.turnOff();
* System.***out***.println();
* }
* }
* }

**2.**Design an interface Bank with methods deposit(), withdraw(), and getBalance().  
Implement this in SavingsAccount and CurrentAccount classes.

* Use inheritance to create a base Account class.
* Demonstrate method overriding with customized logic for withdrawal (e.g., minimum balance in SavingsAccount).
* package Assignment4;
* interface BankInterface {
* void deposit(double amount);
* void withdraw(double amount);
* double getBalance();
* }
* class Account implements BankInterface {
* protected double balance;
* public void deposit(double amount) {
* balance += amount;
* System.***out***.println("Deposited: " + amount);
* }
* public double getBalance() {
* return balance;
* }
* public void withdraw(double amount) {
* if (amount <= balance) {
* balance -= amount;
* System.***out***.println("Withdrawn: " + amount);
* } else {
* System.***out***.println("Insufficient balance.");
* }
* }
* }
* class SavingsAccount extends Account {
* private final double minimumBalance = 500;
* public void withdraw(double amount) {
* if ((balance - amount) >= minimumBalance) {
* balance -= amount;
* System.***out***.println("Withdrawn from SavingsAccount: " + amount);
* } else {
* System.***out***.println("Cannot withdraw. Minimum balance of " + minimumBalance + " must be maintained.");
* }
* }
* }
* class CurrentAccount extends Account {
* public void withdraw(double amount) {
* if (amount <= balance) {
* balance -= amount;
* System.***out***.println("Withdrawn from CurrentAccount: " + amount);
* } else {
* System.***out***.println("Insufficient funds in CurrentAccount.");
* }
* }
* }
* public class Bank\_main {
* public static void main(String[] args) {
* BankInterface savings = new SavingsAccount();
* savings.deposit(1000);
* savings.withdraw(400);
* savings.withdraw(200);
* System.***out***.println("Savings Balance: " + savings.getBalance());
* System.***out***.println();
* BankInterface current = new CurrentAccount();
* current.deposit(1000);
* current.withdraw(1100);
* current.withdraw(700);
* System.***out***.println("Current Balance: " + current.getBalance());
* }
* }

**3**

Create a base class Vehicle with method start().  
Derive Car, Bike, and Truck from it and override the start() method.

* Create a static method that accepts Vehicle type and calls start().
* Pass different vehicle objects to test polymorphism.
* package Assignment4;
* class MyVehicle {
* void start() {
* System.***out***.println("Vehicle is starting...");
* }
* }
* class Car extends MyVehicle {
* void start() {
* System.***out***.println("Car is starting...");
* }
* }
* class Bike\_h extends MyVehicle {
* void start() {
* System.***out***.println("Bike is starting...");
* }
* }
* class Truck extends MyVehicle {
* void start() {
* System.***out***.println("Truck is starting...");
* }
* }
* public class car\_main {
* static void startVehicle(MyVehicle v) {
* v.start();
* }
* public static void main(String[] args) {
* MyVehicle car = new Car();
* MyVehicle bike = new Bike\_h();
* MyVehicle truck = new Truck();
* *startVehicle*(car);
* *startVehicle*(bike);
* *startVehicle*(truck);
* }
* }

**4.**

Design an abstract class Person with fields like name, age, and abstract method getRoleInfo().  
Create subclasses:

* Student: has course and roll number.
* Professor: has subject and salary.
* TeachingAssistant: extends Student and implements getRoleInfo() in a hybrid way.
* Create and print info for all roles using overridden getRoleInfo().
* package Assignment4;
* abstract class Person {
* String name;
* int age;
* Person(String name, int age) {
* this.name = name;
* this.age = age;
* }
* abstract void getRoleInfo();
* }
* class Student extends Person {
* String course;
* int rollNo;
* Student(String name, int age, String course, int rollNo) {
* super(name, age);
* this.course = course;
* this.rollNo = rollNo;
* }
* void getRoleInfo() {
* System.***out***.println("Student Name: " + name);
* System.***out***.println("Age: " + age);
* System.***out***.println("Course: " + course);
* System.***out***.println("Roll No: " + rollNo);
* }
* }
* class Professor extends Person {
* String subject;
* double salary;
* Professor(String name, int age, String subject, double salary) {
* super(name, age);
* this.subject = subject;
* this.salary = salary;
* }
* void getRoleInfo() {
* System.***out***.println("Professor Name: " + name);
* System.***out***.println("Age: " + age);
* System.***out***.println("Subject: " + subject);
* System.***out***.println("Salary: " + salary);
* }
* }
* class TeachingAssistant extends Student {
* String assignedSubject;
* TeachingAssistant(String name, int age, String course, int rollNo, String assignedSubject) {
* super(name, age, course, rollNo);
* this.assignedSubject = assignedSubject;
* }
* void getRoleInfo() {
* System.***out***.println("Teaching Assistant Name: " + name);
* System.***out***.println("Age: " + age);
* System.***out***.println("Course: " + course);
* System.***out***.println("Roll No: " + rollNo);
* System.***out***.println("Assigned Subject: " + assignedSubject);
* }
* }
* public class Teacher {
* public static void main(String[] args) {
* Person s = new Student("Alice", 20, "B.Tech", 101);
* Person p = new Professor("Dr. Bob", 45, "Mathematics", 75000);
* Person t = new TeachingAssistant("Charlie", 22, "M.Sc", 201, "Physics");
* System.***out***.println("=== Student Info ===");
* s.getRoleInfo();
* System.***out***.println("\n=== Professor Info ===");
* p.getRoleInfo();
* System.***out***.println("\n=== Teaching Assistant Info ===");
* t.getRoleInfo();
* }
* }

5.Create:

* Interface Drawable with method draw()

Abstract class Shape with abstract method area()  
Subclasses: Circle, Rectangle, and Triangle.

* Calculate area using appropriate formulas.
* Demonstrate how interface and abstract class work together.
* package Assignment4;
* interface MyDrawable {
* void draw();
* }
* abstract class MyShape implements MyDrawable {
* abstract void area(); // Abstract method for area
* }
* class MyCircle extends MyShape {
* double radius = 5;
* public void draw() {
* System.***out***.println("Drawing Circle");
* }
* public void area() {
* double area = 3.14 \* radius \* radius;
* System.***out***.println("Area of Circle: " + area);
* }
* }
* class MyRectangle extends MyShape {
* double length = 10;
* double breadth = 5;
* public void draw() {
* System.***out***.println("Drawing Rectangle");
* }
* public void area() {
* double area = length \* breadth;
* System.***out***.println("Area of Rectangle: " + area);
* }
* }
* class MyTriangle extends MyShape {
* double base = 6;
* double height = 4;
* public void draw() {
* System.***out***.println("Drawing Triangle");
* }
* public void area() {
* double area = 0.5 \* base \* height;
* System.***out***.println("Area of Triangle: " + area);
* }
* }
* public class Shape\_main {
* public static void main(String[] args) {
* MyShape circle = new MyCircle();
* MyShape rectangle = new MyRectangle();
* MyShape triangle = new MyTriangle();
* System.***out***.println("=== Circle ===");
* circle.draw();
* circle.area();
* System.***out***.println("\n=== Rectangle ===");
* rectangle.draw();
* rectangle.area();
* System.***out***.println("\n=== Triangle ===");
* triangle.draw();
* triangle.area();
* }
* }