```
import java.util.Scanner;

// QB 791
class subject{
    void display(){
        System.out.println("Parent class method");
    }

}
class java extends subject{
    void displayMarks(){
        System.out.println("Child class method");
    }

    public static void main(String[] args) {
        java obj = new java();
        obj.displayMarks();
        obj.display();
    }
}

// QB 792

class Base{
    void baseDisplay(){
        System.out.println("Base class method");
    }
}
class Derived1 extends Base{
    void derived1Display(){
        System.out.println("Derived1 class method");
        }
}
class Derived2 extends Derived1{
    void derived2Display(){
        System.out.println("Derived2 class method");
    }

    public static void main(String[] args) {
        Derived2 obj = new Derived2();
        obj.derived2Display();
        obj.derived1Display();
        obj.baseDisplay();
    }
}

// QB 793

class HOD1 {
    void displayHOD1(){
```

```
        System.out.println("HOD1 method");
    }
}
class Faculty1 extends HOD1{
    void displayFaculty1(){
        System.out.println("Faculty1 method");
    }

}
class Faculty2 extends HOD1{
    void displayFaculty2(){
        System.out.println("Faculty2 method");
    }
}
class Student1 extends Faculty1{
    void displayStudent1(){
        System.out.println("Student1 method");
    }
}
class Student2 extends Faculty2{
    void displayStudent2(){
        System.out.println("Student2 method");
    }
}
class Run{
    public static void main(String[] args) {
        Student1 s1 = new Student1();
        Student2 s2 = new Student2();
        s1.displayStudent1();
        s1.displayFaculty1();
        s1.displayHOD1();
        s2.displayStudent2();
        s2.displayFaculty2();
        s2.displayHOD1();
    }
}

// QB 794

class Member{
    String name;
    int age;
    long mobile_number;
    String address;
    double salary;

    void printSalary(){
        System.out.println("Salary is: "+salary);
    }
}
```

```
class Employee extends Member{
    String specialization;

}
class Manager extends Member{
    String department;
}
class Run1{
    public static void main(String[] args) {
        Employee e1 = new Employee();
        e1.name = "Yash";
        e1.age = 23;
        e1.address = "LJIET";
        e1.mobile_number = 1234567890;
        e1.salary = 30000;
        e1.specialization = "Computer engineering";
        Manager m1 = new Manager();
        m1.name = "Abhi";
        m1.age = 25;
        m1.address = "LJIET";
        m1.mobile_number = 1234567890;
        m1.salary = 30000;
        m1.department = "Computer Engineering";
        System.out.println(e1.name);
        System.out.println(e1.age);
        System.out.println(e1.address);
        System.out.println(e1.mobile_number);
        System.out.println(e1.salary);
        System.out.println(e1.specialization);
        System.out.println(m1.name);
        System.out.println(m1.age);
        System.out.println(m1.address);
        System.out.println(m1.mobile_number);
        System.out.println(m1.salary);
        System.out.println(m1.department);
    }
}

// QB 795

class Shape{
    void print(){
        System.out.println("This is shape");
    }
}
class Rectangle extends Shape{
    void print(){
        super.print();
        System.out.println("This is Rectangular shape");
    }
```

```
}
class Circle extends Shape {
    void print(){
        System.out.println("This is circular shape");
    }
}
class Square extends Rectangle{
    void print() {
        super.print();
        System.out.println("Square is rectangle");
    }
}
class Run2{
    public static void main(String[] args) {
        Square obj = new Square();
        obj.print();
    }
}

// QB 796

class rectangle{
    int length, breadth;
    void printarea(){
        System.out.println("Area: "+(length * breadth));
    }
    void printperimeter(){
        System.out.println("Perimeter: "+(2*(length+breadth)));
    }
    rectangle(int length, int breadth){
        this.length = length;
        this.breadth = breadth;
    }
}

class square extends rectangle{
    int side;
    square(int side){
        super(side,side);
        this.side =side;
    }

    public static void main(String[] args) {
        square obj = new square(5);
        obj.printarea();
        obj.printperimeter();
        rectangle obj1 = new rectangle(5,8);
        obj1.printarea();
        obj1.printperimeter();
    }
```

```
}

// QB 797

class Parent{
    void method(){
        System.out.println("This is parent class");
    }
}
class Child extends Parent{
    void method(){
        System.out.println("This is child class");
    }

    public static void main(String[] args) {
        Child obj = new Child();
        obj.method();
        Parent obj1 = new Parent();
        obj1.method();
    }
}

// QB 841

class Vehicle{
    int max_speed = 120;
}
class car extends Vehicle{
    int max_speed = 100;
    void display(){
        System.out.println("Max speed of car class is: "+max_speed);
        System.out.println("Max speed of vehicle class is: "+super.max_speed);
    }

    public static void main(String[] args) {
        car obj = new car();
        obj.display();
    }
}

// QB 842

class Parent1{
    void method1(){
        System.out.println("Parent class method");
    }
}
class child1 extends Parent1{
    void method1(){
        super.method1();
```

```
    }

    public static void main(String[] args) {
        child1 obj = new child1();
        obj.method1();
    }
}

// QB 843

class Test{
    Test (){
        System.out.println("No-Arg Constructor");
    }
    Test (int i){
        System.out.println("int-Arg Constructor");
    }
}

class Construct extends Test{
    Construct(){
        super(5);
        System.out.println("No-Arg child class Constructor");
    }

    public static void main(String[] args) {
        Construct obj = new Construct();
    }
}

// QB

class demo{
    void display(){
        System.out.println("Parent class method");
    }
}
class A extends demo{
    void display(){
        System.out.println("Class A method");
    }
}
class B extends demo{
    void display(){
        System.out.println("Class B method");
    }

    public static void main(String[] args) {
        B obj = new B();
        obj.display();
```

```
    }
}

// QB 814

class Figure {
    double dim1;
    double dim2;
    Figure(double dim1, double dim2) {
        this. dim1 = dim1;
        this. dim2 = dim2;
    }
    double area(){
     return 0;
};
}
class Rectangle2 extends Figure {
    Rectangle2(double a, double b) {
        super(a, b);
    }
    double area() {
        System.out.println("Inside Area for Rectangle.");
        return dim1 * dim2;
    }
}
class Triangle extends Figure {
    Triangle(double a, double b) {
        super(a, b);
    }
    double area() {
        System.out.println("Inside Area for Triangle.");
        return dim1 * dim2 / 2;
    }
}
class Areas {
    public static void main(String args[]) {

        System.out.println("Enter length and breadth of Rectangle");
        double len=12;
        double bdt=14;
        System.out.println("Enter height and side of Triangle");
        double ht=12;
        double sd=18;
        Rectangle2 r = new Rectangle2(len, bdt);
        Triangle t = new Triangle(ht, sd);
        System.out.println("Area is " + r.area());
        System.out.println("Area is " + t.area());
    }
}
```

```
// QB 815

class Vegetable
{
    public String color;
}

class Potato extends Vegetable
{
    public String toString()
    {
        color = "Brown";
        return "potato -->"+color;
    }
}

class Brinjal extends Vegetable
{
    public String toString()
    {
        color = "purple";
        return "Brinjal -->"+this.color;
    }
}

class Tomato extends Vegetable
{
    public String toString()
    {
        color="red";
        return "Tomato -->"+color;
    }
}

class veg_dis
{
    public static void main(String [] args)
    {
        Potato p = new Potato();
        Brinjal b = new Brinjal();
        Tomato t = new Tomato();
        System.out.println(p);
        System.out.println(b);
        System.out.println(t);
    }
}

// QB 816

class A1{
```

```
    void display(){
        System.out.println("I am in A Class");
    }
}
class B1 extends A1{
    void display(){
        super.display();
        System.out.println("I am in B Class");
    }

    public static void main(String[] args) {
        B1 b1 = new B1();
        b1.display();
    }
}

// QB 817

class Book {
    String author_name;

    Book(String author_name) {
        this.author_name = author_name;
    }

    void display() {
        System.out.println("Author: " + author_name);
    }
}
class book_publication extends Book{
        String title;
        book_publication(String title){
            super("");
            this.title = title;
        }
        void display(){
            System.out.println("Book: "+title);
        }
    }
    class paper_publication extends Book{
        String title;
        paper_publication(String title){
            super("");
            this.title = title;
        }
        void display(){
            System.out.println("Paper: "+title);
        }
    }
    class one{
```

```
        public static void main(String args[]){
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter Author name: ");
            Book b1 = new Book(sc.nextLine());
            System.out.println("Enter Book name: ");
            book_publication o2 = new book_publication(sc.nextLine());
            System.out.println("Enter Paper name: ");
            paper_publication o3 = new paper_publication(sc.nextLine());
            Book r;
            r = b1;
            r.display();
            r = o2;
            r.display();
            r = o3;
            r.display();
        }
    }

        // EX
// single inheritance
class Employee
{
                float salary=34534*12;
}
class Executive extends Employee
{
        float bonus=3000*6;
        public static void main(String args[])
        {
                Executive obj=new Executive();
                System.out.println("Total salary credited: "+obj.salary);
                System.out.println("Bonus of six months: "+obj.bonus);
        }
}
//*****************************
//Multi-level Inheritance
class Student
{
        int reg_no;
        void getNo(int no)
        {
                reg_no=no;
        }
        void putNo()
        {
                System.out.println("registration number= "+reg_no);
        }
}

class Marks extends Student   //intermediate sub class
```

```
{
        float marks;
        void getMarks(float m)
        {
                marks=m;
        }
        void putMarks()
        {
                System.out.println("marks= "+marks);
        }
}

class Sports extends Marks  //derived class
{
        float score;
        void getScore(float scr)
        {
                score=scr;
        }
        void putScore()
        {
                System.out.println("score= "+score);
        }
}
class MultilevelInheritanceExample
{
        public static void main(String args[])
        {
                Sports ob=new Sports();
                ob.getNo(987);
                ob.putNo();
                ob.getMarks(78);
                ob.putMarks();
                ob.getScore(68);
                ob.putScore();
        }
}
//*****************************
//Hierarchical Inheritance
class Student1
{
        public void methodStudent()
        {
                System.out.println("The method of the class Student1 invoked.");
        }
}
class Science extends Student1
{
        public void methodScience()
        {
```

```
                System.out.println("The method of the class Science invoked.");
        }
}
class Commerce extends Student1
{
        public void methodCommerce()
        {
                System.out.println("The method of the class Commerce invoked.");
        }
}
class Arts extends Student1
{
        public void methodArts()
        {
                System.out.println("The method of the class Arts invoked.");
        }
}
class HierarchicalInheritanceExample
{
        public static void main(String args[])
        {
                Science sci = new Science();
                Commerce comm = new Commerce();
                Arts art = new Arts();
                //all the sub classes can access the method of super class
                sci.methodStudent();
                comm.methodStudent();
                art.methodStudent();
        }
}
//*******************************
//Hybrid Inheritance (without multiple)

class GrandFather  //parent class
{
        public void show()
        {
                System.out.println("I am grandfather.");
        }
}

class Father extends GrandFather  //inherits GrandFather properties
{
        public void show()
        {
                System.out.println("I am father.");
        }
}

class Son extends Father  //inherits Father properties
```

```
{
        public void show()
        {
                System.out.println("I am son.");
        }
}

class Daughter extends Father   //inherits Father properties
{
        public void show()
        {
                System.out.println("I am a daughter.");
        }
        public static void main(String args[])
        {
                Daughter obj = new Daughter();
                obj.show();
        }
}

//Multiple Inheritance(Java does not support multiple inheritances due to
ambiguity.)
//*****************************

//QB 845
class Circle {
    final double CIRCLE_PI = 3.14159265359;
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    public double area() {
        return CIRCLE_PI * radius * radius;
    }
}

class Main{
        public static void main(String args[]){
                Circle myCircle = new Circle(5.0);
                double area = myCircle.area();
                System.out.println("The area of the circle is: " + area);
        }
}

//QB 846
class Animal
{
        static final int NUM_LEGS = 4;
```

```
    static final int NUM_EARS = 2;
    static final int NUM_EYES = 2;
    static final int HAS_TAIL = 1;

        final void displayCharacteristics()
        {
                System.out.println(NUM_LEGS);
                System.out.println(NUM_EARS);
                System.out.println(NUM_EYES);
                System.out.println(HAS_TAIL);
        }
}
class Cow extends Animal{
        /* //final method in parent class with same name, so this method can't be
used here to override.
        void displayCharacteristics()
        {
                System.out.println(NUM_LEGS);
                System.out.println(NUM_EARS);
                System.out.println(NUM_EYES);
                System.out.println(HAS_TAIL);
        }
        */
        void displayAdditionalCharacteristics(){
                System.out.println("Hi cow - Additional Characteristics");
        }
}
class Run{
        public static void main(String args[]){
                Cow c = new Cow();
                c.displayCharacteristics() ; //parent method called as final
                c.displayAdditionalCharacteristics();
        }
}

//QB 847
final class Animals {

    // static method
    public static void display() {
        System.out.println("This is an animal.");
    }
}

class Tiger extends Animals {  //invalid as cannot inherit from final Animals
//class Tiger {               //valid as can inherit from non-final Animals

    // method specific to Tiger class
    public void displayTiger() {
        System.out.println("This is a tiger.");
```

```
    }

        public static void main(String args[])
        {
                Tiger t = new Tiger();
                // t.display();  //cant access final class property
                t.displayTiger();
                Animals a = new Animals();
                a.display();
        }
}

/*since the Animal class is marked as final, it cannot be extended or subclassed
further,
but its static methods can still be accessed by other classes without extending,
such as the Tiger class in this example.*/
```