Write the code in java for the **“Vehicle”** class. The tester class and the output is given below:

| **Tester class** | **Output** |
| --- | --- |
| public class TesterVehicle{  public static void main(String [] args){  Vehicle car = new Vehicle();  System.out.println("Attributes of car object:");  System.out.println(car.type);  System.out.println(car.wheels);  System.out.println(car.color);  System.out.println("=========");  Vehicle bike = new Vehicle();  bike.type="Motor bike";  bike.wheels=2;  bike.color="Red";  System.out.println("Attributes of bike object:");  System.out.println(bike.type);  System.out.println(bike.wheels);  System.out.println(bike.color);  }  } | Attributes of car object:  Car  4  White  =========  Attributes of bike object:  Motor bike  2  Red |

| **Driver Code** | **Output** |
| --- | --- |
| public class TesterCourse{  public static void main(String[] args) {  Course c1 = new Course();  Course c2 = new Course();    System.*out*.println("========== 1 ==========");  c1.createCourse("Programming Language I", "CSE110", 3);  c1.displayCourse();    System.*out*.println("========== 2 ==========");  c2.createCourse("Data Structures", "CSE220", 3);  c2.displayCourse();    System.*out*.println("========== 3 ==========");  c1.updateCourse("Programming Language II", "CSE111", 3);  c1.displayCourse();  }  } | ========== 1 ==========  Course Name: Programming Language I  Course Code: CSE110  Course Credit: 3  ========== 2 ==========  Course Name: Data Structures  Course Code: CSE220  Course Credit: 3  ========== 3 ==========  Course Name: Programming Language II  Course Code: CSE111  Course Credit: 3 |

Create a **Dog** class so that the tester code generates the given output:

| **Driver Code** | **Expected Output** |
| --- | --- |
| public class TesterDog{  public static void main (String[] args) {  Dog scooby = new Dog();  Dog oldie = new Dog();  Dog goofy = new Dog();    scooby.changeName("Scooby");  goofy.changeName("Goofy");    System.*out*.println("1. ===============");  System.*out*.println(scooby.bark());  System.*out*.println("2. ===============");  System.*out*.println(oldie.bark());  System.*out*.println("3. ===============");  oldie.changeColor("White");  System.*out*.println("4. ===============");  System.*out*.println(oldie.bark());  System.*out*.println("5. ===============");  System.*out*.println(goofy.bark());  System.*out*.println("6. ===============");  scooby.changeColor("Brown");  System.*out*.println("7. ===============");  System.*out*.println(scooby.bark());  System.*out*.println("8. ===============");  goofy.changeColor("Black");  }  } | 1. ===============  Scooby is barking  2. ===============  A dog is barking  3. ===============  This dog is White  4. ===============  White dog is barking  5. ===============  Goofy is barking  6. ===============  Scooby is Brown  7. ===============  Scooby the Brown dog is barking  8. ===============  Goofy is Black |

Design the **Reader** class in such a way so that the following code provides the expected output.

* A reader will have a name, capacity to read and an array of books they are reading.
* The initial capacity of a reader will be 0. The initial name will be “New user”.
* A new array is created every time a reader’s capacity is increased, which replaces the initial array.

| **Driver Code** | **Expected Output** |
| --- | --- |
| public class Reader\_tester {  public static void main(String[] args){  Reader r1 = new Reader();  Reader r2 = new Reader();    r1.createReader("Albert", 2);  r2.createReader("Issac", 5);    System.*out*.println("1 ==========");  r1.readerInfo();    System.*out*.println("2 ==========");  r2.addBook("Java");  r2.addBook("Python");  r2.addBook("C++");  r2.readerInfo();    System.*out*.println("3 ==========");  r1.addBook("C#");  r1.addBook("Rust");  r1.addBook("GoLang");    System.*out*.println("4 ==========");  r1.increaseCapacity(5);  r1.addBook("Python");    System.*out*.println("5 ==========");  r1.readerInfo();  }  } | 1 ==========  Name: Albert  Capacity: 2  Books:  No books added yet  2 ==========  Name: Issac  Capacity: 5  Books:  Book 1: Java  Book 2: Python  Book 3: C++  3 ==========  No more space for new book  4 ==========  Albert's capacity increased to 5  5 ==========  Name: Messi  Capacity: 5  Books:  Book 1: C#  Book 2: Rust  Book 3: Python |

| **1** | public class Task11 { |
| --- | --- |
| **2** | public int p = 3, y = 2, sum; |
| **3** | public void methodA(){ |
| **4** | int x = 0, y = 0; |
| **5** | y = y + this.y; |
| **6** | x = sum + 2 + p; |
| **7** | sum = x + y + methodB(p, y); |
| **8** | System.*out*.println(x + " " + y+ " " + sum); |
| **9** | } |
| **10** | public int methodB(int p, int n){ |
| **11** | int x = 0; |
| **12** | y = y + (++p); |
| **13** | x = x + 2 + n; |
| **14** | sum = sum + x + y; |
| **15** | System.*out*.println(x + " " + y+ " " + sum); |
| **16** | return sum; |
| **17** | } |
| **18** | } |

**Driver code:**

| public class Tester11 {  public static void main(String [] args){  Task11 t1 = new Task11 ();  t1.methodA();  t1.methodA();  Task11 t2 = new Task11();  System.out.println(t2.methodB(2,3));  }  } | **Outputs** | | |
| --- | --- | --- | --- |
| **x** | **y** | **Sum** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| 1 | public class Test2 { |
| --- | --- |
| 2 | int x = 3, y = 1, z = -4; |
| 3 | double p = 2.5; |
| 4 | public void methodA(int n, int x) { |
| 5 | this.x = methodB(x, n); |
| 6 | p = this.x + n % x \* 2.0; |
| 7 | y = (z++) + methodB(z, (int) p) + (++z); |
| 8 | System.out.println(this.x + " " + (n + y) + " " + (x + z)) ; |
| 9 | } |
| 10 | public int methodB(int q, int n) { |
| 11 | int arr[] = {2, -5, 6}; |
| 12 | arr[0] = arr[2] - this.x + n; |
| 13 | arr[1] = q - arr[1]; |
| 14 | arr[2] = arr[q % 3] + arr[n % 2]; |
| 15 | System.out.println(arr[0] + " " + arr[1] + " " + arr[2]) ; |
| 16 | return arr[1] + arr[2] - arr[0]; |
| 17 | } |
| 18 | } |

| public class Tester2{  public static void main(String [] args){  Test2 t = new Test2();  t.methodA(3, 4);  }  } | **Outputs** | | |
| --- | --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Constructor with multiclass

|  |  |
| --- | --- |
| public class TeamTester {  public static void main(String[] args) {  Player p1 = new Player("Sakep", 37, 200);  Player p2 = new Player("Tamem", 35, 180);  p1.details();  System.*out*.println("=======1=======");  p2.details();  System.*out*.println("=======2=======");  p1.addRival(p2);  System.*out*.println("=======3=======");  p1.showRival();  System.*out*.println("=======4=======");  Team b = new Team();  b.updateName("Bangladesh");  System.*out*.println("=======5=======");  b.addPlayer(p1);  System.*out*.println("=======6=======");  b.addPlayer(p2);  System.*out*.println("=======7=======");  b.printDetails();  System.*out*.println("=======8=======");  Team a = new Team("Australia");  Player p3 = new Player("Ponting", 50, 300);  a.addPlayer(p3);  Player p4 = new Player("Lee", 49, 200);  a.addPlayer(p4);  a.printDetails();  }  } | Name: Sakep  Age: 37, Total matches: 200  =======1=======  Name: Tamem  Age: 35, Total matches: 180  =======2=======  =======3=======  Rival of Sakep is Tamem  Age: 35, Total matches: 180  =======4=======  =======5=======  Player Sakep added  =======6=======  Player Tamem added  =======7=======  Team: Bangladesh  List of players:  Name: Sakep  Age: 37, Total Matches: 200  Name: Tamem  Age: 35, Total Matches: 180  =======8=======  Player Ponting added  Player Lee added  Team: Australia  List of players:  Name: Ponting  Age: 50, Total Matches: 300  Name: Lee  Age: 49, Total Matches: 200 |

Write the “**Product**“ class to show the following output

**Note:** Make sure to use proper ***Encapsulation concepts* for the setter & getter methods**. All the attributes should have **Private** access.

| **Driver Code** | **Output** |
| --- | --- |
| public class ProductTester{  public static void main(String[] args) {  System.out.println("< —--—----1—-------->");  Product product1 = new Product();  product1.displayInfo();  System.out.println("< —--—----2—-------->");  Product product2 = new Product("Laptop", 1200.00);  product2.setQuantity(10);  product2.displayInfo(true);  System.out.println("< —--—----3—-------->");    System.out.println("Retrieved Quantity: " + product2.getQuantity());  }  } | < —--—----1—-------->  Product Name: Unknown  Price: $0.0  < —--—----2—-------->  Product Name: Laptop  Price: $1200.0  Quantity: 10  < —--—----3—-------->  Retrieved Price: $1200.0  Retrieved Quantity: 10 |

# INHERITANCE

| **Driver Code** | **Output** |
| --- | --- |
| public class Animal {  public String name;  public int age;  public String color;  public Animal(String name, int age, String color) {  this.name = name;  this.age = age;  this.color = color;  }  public void makeSound() {  System.out.println("Animal makes a sound");  }  public String info() {  return "Name: "+name+"\nAge: "+age+"\nColor: "+color+"\n";  } }  **//Tester Class**  public class AnimalTester {  public static void main(String[] args) {  Dog dog = new Dog("Buddy", 5, "Brown", "Bulldog");  Cat cat = new Cat("Kitty", 3, "White", "Persian");  System.out.println("1.========");  System.out.println(dog.info());  System.out.println("2.========");  System.out.println(cat.info());  System.out.println("3.========");  dog.makeSound();  System.out.println("4.========");  cat.makeSound();  } } | 1.========  Name: Buddy  Age: 5  Color: Brown  Breed: Bulldog  2.========  Name: Kitty  Age: 3  Color: White  Breed: Persian  3.========  Brown color Buddy is barking  4.========  White color Kitty is meowing |

| **Driver Code** | **Output** |
| --- | --- |
| public class Student{  private String name = "Just a Student";  private String department = "nothing";    public void updateDepartment(String dpt){  this.department = dpt;  }  public void updateName(String name){  this.name = name;  }  public void details(){  System.out.println("Name : " + name + " Department: " + department);  }  }  **//Tester Class**  public class TestStudent{  public static void main(String [] args){  BBAStudent b1 = new BBAStudent();  BBAStudent b2 = new BBAStudent("Humty Dumty");  BBAStudent b3 = new BBAStudent("Little Bo Peep");  b1.details();  System.out.println("1---------------");  b2.details();  System.out.println("2---------------");  b3.details();  }  } | Name : Default Department: BBA  1---------------  Name : Humty Dumty Department: BBA  2---------------  Name : Little Bo Peep Department: BBA |

| **1** | **public class A {** |
| --- | --- |
| **2** | **static int temp = 4;** |
| **3** | **static int x = -10;** |
| **4** | **int sum, y;** |
| **5** | **public A() {** |
| **6** | **y = temp - 2;** |
| **7** | **sum = temp + 1;** |
| **8** | **temp -= 2;** |
| **9** | **}** |
| **10** | **public void methodA(int m, int n) {** |
| **11** | **y = y + m + (temp++);** |
| **12** | **x = x + 1 + n;** |
| **13** | **sum = sum + x + y;** |
| **14** | **System.out.println(x + " " + y + " " + sum);** |
| **15** | **}** |
| **16** | **}** |
| **17** | **class B extends A {** |
| **18** | **static int x = 0;** |
| **19** | **int sum = -6;** |
| **20** | **public B() {** |
| **21** | **sum = 0;** |
| **22** | **y = temp + 3;** |
| **23** | **super.sum = 3 + temp + 2;** |
| **24** | **temp -= 2;** |
| **25** | **}** |
| **26** | **public B(B b) {** |
| **27** | **sum = b.sum + super.sum;** |
| **28** | **x = b.x + 1;** |
| **29** | **b.methodB(2, 3);** |
| **30** | **}** |
| **31** | **public void methodB(int m, int n) {** |
| **32** | **int y = 0;** |
| **33** | **y = y + this.y;** |
| **34** | **x = y + 2 + (++temp);** |
| **35** | **methodA(x, y);** |
| **36** | **sum = x + y + sum;** |
| **37** | **System.out.println(x + " " + y + " " + sum);** |
| **38** | **}** |
| **39** | **}** |

| **public class Tester {**  **public static void main(String [] args) {**  **A a1 = new A();**  **B b1 = new B();**  **B b2 = new B(b1);**  **b1.methodA(2, 3);**  **b2.methodB(3, 8);**  **}**  **}** | **Output:** | | |
| --- | --- | --- | --- |
| **x** | **y** | **sum** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Design the **SultansDine** class with the necessary property to produce the output from the given driver code.

Subtaks:

1. Create SultansDine class
2. Create 2 static variable and 1 static array
3. Create 1 static method
4. Calculation of branch sell is given below
   1. If sellQuantity < 10:
      1. Branch\_sell = quantity \* 300
   2. Else if sellQuantity < 20:
      1. Branch\_sell = quantity \* 350
   3. Else
      1. Branch\_sell = quantity \* 400
5. Calculation of branch’s sell percentage = (branch’s sell / total sell) \* 100

| **Driver Code** | **Output** |
| --- | --- |
| public class SultansDineTester {  public static void main(String[] args) {  SultansDine.details();  System.out.println("1===================");  SultansDine dhanmondi = new SultansDine("Dhanmondi");  dhanmondi.sellQuantity(25);  dhanmondi.branchInformation();  System.out.println("2===================");  SultansDine.details();  System.out.println("3===================");  SultansDine baily\_road=new SultansDine("Baily Road");  baily\_road.sellQuantity(15);  baily\_road.branchInformation();  System.out.println("4===================");  SultansDine.details();  System.out.println("5===================");  SultansDine gulshan = new SultansDine("Gulshan");  gulshan.sellQuantity(9);  gulshan.branchInformation();  System.out.println("6===================");  SultansDine.details();  }  } | Total Number of branch(s): 0  Total Sell: 0 Taka  1===================  Branch Name: Dhanmondi  Branch Sell: 10000 Taka  2===================  Total Number of branch(s): 1  Total Sell: 10000 Taka  Branch Name: Dhanmondi, Branch Sell: 10000 Taka  Branch consists of total sell's 100.00  3===================  Branch Name: Baily Road  Branch Sell: 5250 Taka  4===================  Total Number of branch(s): 2  Total Sell: 15250 Taka  Branch Name: Dhanmondi, Branch Sell: 10000 Taka  Branch consists of total sell's 65.57  Branch Name: Baily Road, Branch Sell: 5250 Taka  Branch consists of total sell's 34.43  5===================  Branch Name: Gulshan  Branch Sell: 2700 Taka  6===================  Total Number of branch(s): 3  Total Sell: 17950 Taka  Branch Name: Dhanmondi, Branch Sell: 10000 Taka  Branch consists of total sell's 55.71  Branch Name: Baily Road, Branch Sell: 5250 Taka  Branch consists of total sell's 29.25  Branch Name: Gulshan, Branch Sell: 2700 Taka  Branch consists of total sell's 15.04 |

Design the **Nokia** class derived from the Mobile class so that the following output is produced.

| **Parent Class** | |
| --- | --- |
| class Mobile {  public String model;  public String IMEICode;  public boolean simCardStatus;    public Mobile(String model, String IMEICode, boolean simCardStatus) {  this.model = model;  this.IMEICode = IMEICode;  this.simCardStatus = simCardStatus;  System.out.println("Model " + model + " is manufactured.");  }    public String getCountryName(String countryCode) {  if (countryCode.equals("880")) {  return "Bangladesh";  } else if (countryCode.equals("455")) {  return "USA";  }  return null;  }    public void activateSimCard() {  if (!simCardStatus) {  simCardStatus = true;  System.out.println("SIM card is activated successfully.");  }  }    @Override  public String toString() {  return "Mobile Phone Detail:\nModel: " + model + "\nIMEICode: " + IMEICode + "\nSIM Card Status: " + simCardStatus;  }  }  //Driver code below | |
| **Driver Code** | **Output** |
| public class MobileTester {  public static void main(String[] args) {  Nokia N3110 = new Nokia("N3110", true, "IMEI-102", 0);  System.out.println(N3110);  System.out.println("1========================");  Nokia N1100 = new Nokia("N1100", false, "IMEI-124", 100);  System.out.println(N1100);  System.out.println("2========================");  System.out.println(N3110.dialCall("88017196xxxx"));  System.out.println("3========================");  N3110.rechargeSIMCard(200);  N1100.rechargeSIMCard(300);  System.out.println("4========================");  System.out.println(N3110.dialCall("88017196xxxx"));  System.out.println("5========================");  System.out.println(N1100.dialCall("45517196xxxx"));  System.out.println("6========================");  N1100.activateSimCard();  System.out.println("7========================");  System.out.println(N1100.dialCall("45517196xxxx"));  System.out.println("8========================");  System.out.println(N1100.dialCall("96617196xxxx"));  }  } | Model N3110 is manufactured.  Mobile Phone Detail:  Model: N3110  IMEICode: IMEI-102  SIM Card Status: true  Balance: 0.0 TK  1========================  Model N1100 is manufactured.  Mobile Phone Detail:  Model: N1100  IMEICode: IMEI-124  SIM Card Status: false  Balance: 100.0 TK  2========================  Insufficient balance! Please recharge.  3========================  Recharge successful! Current balance 200.0 TK.  Recharge successful! Current balance 400.0 TK.  4========================  Dialing the number 88017196xxxx to Bangladesh region.  5========================  No SIM card available! Please check the SIM card connectivity.  6========================  SIM card is activated successfully.  7========================  Dialing the number 45517196xxxx to USA region.  8========================  Dialing is not allowed in this region. |

| 1 | public class TracingExample { |
| --- | --- |
| 2 | public static double counter = 3.0; |
| 3 | public static int a, b; |
| 4 | public TracingExample(int a, int b) { |
| 5 | this.a = a; |
| 6 | b = (a + b) % 2; |
| 7 | counter++; |
| 8 | } |
| 9 | public void methodA(int value) { |
| 10 | this.a += value; |
| 11 | int z = (a + b) % 3; |
| 12 | System.out.println(this.a + " " + z + counter); |
| 13 | if (a >= 15) { |
| 14 | this.methodB(this); |
| 15 | System.out.println(z + counter + " " + this.a); |
| 16 | } |
| 17 | } |
| 18 | public static void methodA(double value) { |
| 19 | a += value; |
| 20 | int z = (a + b) % 3; |
| 21 | System.out.println(a + " " + 1 + z + counter); |
| 22 | a += 2; |
| 23 | } |
| 24 | public static void methodB(TracingExample obj) { |
| 25 | double temp = (counter \* 2) + 2; |
| 26 | obj.a += temp; |
| 27 | System.out.println(obj.a + " " + temp); |
| 28 | } |
| 29 | } |

| public class TestTracing {  public static void main(String[ ] args) {  TracingExample t1 = new TracingExample(2, 3);  TracingExample t2 = new TracingExample(5, 7);  t1.methodA(TracingExample.counter);  t1.methodA(7);  TracingExample.methodA(3.0);  }  } |
| --- |