

Ques 1.

The screenshot shows the Eclipse IDE interface with several Java files listed in the top bar: Ques11.java, Ques12.java, Ques10.java, Ques1a.java, Ques2a.java, Ques3a.java, and Ques4a.java. The main focus is on the Ques7.java file, which contains the following code:

```
1 import java.util.*;
2
3 class Area1 {
4     int length;
5     int breadth;
6
7     public Area1(int length, int breadth) {
8         this.length = length;
9         this.breadth = breadth;
10    }
11
12    public int returnArea() {
13        return length * breadth;
14    }
15 }
16
17 public class Ques7 {
18     public static void main(String[] args) {
19         Scanner sc = new Scanner(System.in);
20         System.out.print("Enter length: ");
21         int length = sc.nextInt();
22         System.out.print("Enter breadth: ");
23         int breadth = sc.nextInt();
24
25         Area1 area = new Area1(length, breadth);
26         System.out.println("Area of rectangle: " + area.returnArea());
27         sc.close();
28     }
29 }
```

Below the code editor is the Eclipse Console window, which displays the following output:

```
Console X
<terminated> Ques7 [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-0940
Enter length: 123
Enter breadth: 123
Area of rectangle: 15129
```

Ques 2.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows tabs for Ques11.java, Ques12.java, Ques10.java, Ques1a.java (highlighted), Ques2a.java, and Ques3a.java.
- Code Editor:** Displays Java code for a class named Area1 and a main method in Ques7. The code uses Scanner to input length and breadth, and prints the calculated area.

```
1 import java.util.*;
2
3 class Area1 {
4     int length;
5     int breadth;
6
7     public Area1(int length, int breadth) {
8         this.length = length;
9         this.breadth = breadth;
10    }
11
12    public int returnArea() {
13        return length * breadth;
14    }
15 }
16
17 public class Ques7 {
18    public static void main(String[] args) {
19        Scanner sc = new Scanner(System.in);
20        System.out.print("Enter length: ");
21        int length = sc.nextInt();
22        System.out.print("Enter breadth: ");
23        int breadth = sc.nextInt();
24
25        Area1 area = new Area1(length, breadth);
26        System.out.println("Area of rectangle: " + area.returnArea());
27        sc.close();
28    }
29 }
```

- Console:** Shows the terminal output of the program execution.

Console Output:

```
<terminated> Ques7 [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx_x86_64
Enter length: 123
Enter breadth: 123
Area of rectangle: 15129
```

Ques 3.

The screenshot shows the Eclipse IDE interface with the Java editor open. The code is for a program that prints the perimeter and area of a triangle. The Java code is as follows:

```
1 //3. Write a program to print the area and perimeter of a triangle having sides
2 //creating a class named 'Triangle' without any parameter in its constructor.
3
4 import java.util.*;
5 class Triangle{
6     int a, b, c;
7     int base;
8     int height;
9
10    public Triangle() {
11        a = 3;
12        b = 4;
13        c = 5;
14    }
15
16    public void printPerimeter() {
17        int perimeter = a + b + c;
18        System.out.println("The perimeter of the triangle is: " + perimeter);
19    }
20
21    public void printArea(int base, int height) {
22        double area = (base* height) / 2.0;
23        System.out.println("Area:" + area);
24    }
25 }
26
27
28 public class Ques3 {
29    public static void main(String[] args) {
30        Scanner sc = new Scanner(System.in);
31        Triangle t = new Triangle();
32        t.printPerimeter();
33        t.printArea(23,123);
34
35
36
37 }
```

The console output shows the results of running the program:

```
<terminated> Ques3 [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre
The perimeter of the triangle is: 12
Area:1414.5
```

Ques 4.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows multiple Java files: Ques11.java, Ques12.java, Ques10.java, Ques1a.java, Ques2a.java, Ques3a.java, Ques4a.java, and Ques4.java (the active file).
- Code Editor:** Displays the Java code for Ques4.java. The code defines a class Triangle1 with methods to calculate perimeter and area, and a main method to test it. The code is as follows:

```
1 | class Triangle1{  
2 |     int a,b,c;  
3 |  
4 |     public Triangle1(int a, int b, int c) {  
5 |         this.a = a;  
6 |         this.b = b;  
7 |         this.c = c;  
8 |     }  
9 |  
10|     public void printPerimeter() {  
11|         int perimeter = a + b + c;  
12|         System.out.println("Perimeter of triangle: " + perimeter);  
13|     }  
14|  
15|     public void printArea() {  
16|         double area = (4+3)/ 2.0;  
17|         System.out.println("Area of the triangle is: " + area);  
18|     }  
19|  
20| }  
21|  
22| public class Ques4{  
23|     public static void main(String[] args) {  
24|         Triangle1 t1 = new Triangle1(3,4,5);  
25|         t1.printPerimeter();  
26|         t1.printArea();  
27|     }  
28| }  
29| }
```

- Console:** Shows the terminal output of the application. It prints:
<terminated> Ques3 [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre
The perimeter of the triangle is: 12
Area:1414.5

Ques 5.

The screenshot shows the Eclipse IDE interface with the following details:

- Project Explorer:** Shows multiple Java files: Ques4.java, Ques12.java, Ques10.java, Ques1a.java, Ques2a.java, Ques3a.java, and Ques4a.java.
- Code Editor:** The active file is Ques5.java. It contains the following code:

```
1 class Rectangle{  
2     int length;  
3     int breadth;  
4     int area;  
5     public Rectangle(int length, int breadth) {  
6         this.length = length;  
7         this.breadth = int length - Rectangle.Rectangle (  
8     }  
9     public void Area(){  
10        area = length * breadth;  
11        System.out.println("Area of the rectangle is: "+ area);  
12    }  
13 }  
14  
15  
16 public class Ques5 {  
17     public static void main(String[] args) {  
18         Rectangle r = new Rectangle(23,21);  
19         Rectangle r2 = new Rectangle(43,12);  
20  
21         r.Area();  
22         r2.Area();  
23     }  
24 }  
25 }  
26 }
```
- Code Completion:** A tooltip is displayed over the line `this.breadth = int length - Rectangle.Rectangle (`, showing the method signature: `int length - Rectangle.Rectangle (int, int)`.
- Console:** The console output shows the execution results:

```
<terminated> Ques5 [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-14  
Area of the rectangle is: 483  
Area of the rectangle is: 516
```

Ques 6.

```
1 Ques11.java    2 Ques12.java    3 Ques10.java    4 Ques1a.java    5 Ques2a.java    6 Ques3
1 import java.util.*;
2
3 class Area{
4     int length;
5     int breadth;
6
7     public void setDim() {
8         Scanner sc = new Scanner(System.in);
9
10
11         System.out.println("Enter the breadth");
12         breadth = sc.nextInt();
13         System.out.println("Enter the length");
14         length = sc.nextInt();
15
16     }
17     public int getArea() {
18         return length * breadth;
19     }
20 }
21
22 public class Ques6 {
23     public static void main(String args[]) {
24         Area a = new Area();
25         a.setDim();
26         System.out.println("Area" + a.getArea());
27
28     }
29 }
30
31 }
32
```

Console X

<terminated> Ques6 [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macos

```
Enter the breadth
23
Enter the length
23
Area529
```

Ques 7.

```
1 import java.util.*;
2
3 class Area1 {
4     int length;
5     int breadth;
6
7     public Area1(int length, int breadth) {
8         this.length = length;
9         this.breadth = breadth;
10    }
11
12    public int returnArea() {
13        return length * breadth;
14    }
15 }
16
17 public class Ques7 {
18    public static void main(String[] args) {
19        Scanner sc = new Scanner(System.in);
20        System.out.print("Enter length: ");
21        int length = sc.nextInt();
22        System.out.print("Enter breadth: ");
23        int breadth = sc.nextInt();
24
25        Area1 area = new Area1(length, breadth);
26        System.out.println("Area of rectangle: " + area.returnArea());
27        sc.close();
28    }
29 }
```

Console X

```
<terminated> Ques7 [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250
Enter length: 231
Enter breadth: 112
Area of rectangle: 25872
```

Ques 8.

```
J Ques11.java J Ques12.java J Ques10.java J Ques1a.java xJ Ques2a.java xJ Ques3a.java J Ques4a.java
1 //8. Print the average of three numbers entered by the user by creating a class
2 //having a method to calculate and print the average.
3
4 import java.util.Scanner;
5
6 class Average{
7     int a, b, c;
8     int aver;
9
10    public double printAvg() {
11        Scanner sc = new Scanner(System.in);
12
13        System.out.println("Enter three nums");
14        a = sc.nextInt();
15        b = sc.nextInt();
16        c = sc.nextInt();
17        double avg = (a+b+c) /3.0;
18        return avg;
19
20
21
22    }
23 }
24
25
26 public class Ques8 {
27    public static void main(String[] args) {
28        Average a = new Average();
29        System.out.println("Average of three nums: " + a.printAvg());
30    }
31 }
32
33
```



```
Console X
<terminated> Ques8 [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-14
Enter three nums
123
213
12
Average of three nums: 116.0
```

Ques 9.

```
Ques11.java Ques12.java Ques10.java Ques1a.java Ques2a.java Ques3a.java
1 //  
2 //9. Print the sum, difference and product of two complex numbers  
3 //'Complex' with separate methods for each operation whose real a  
4 //the user.  
5  
6  
7 import java.util.*;  
8  
9 class Complex {  
10     double real, imag;  
11  
12     Complex(double r, double i) {  
13         real = r;  
14         imag = i;  
15     }  
16  
17     Complex add(Complex o) {  
18         return new Complex(real + o.real, imag + o.imag);  
19     }  
20  
21     Complex sub(Complex o) {  
22         return new Complex(real - o.real, imag - o.imag);  
23     }  
24  
25     Complex mul(Complex o) {  
26         double realPart = real * o.real - imag * o.imag;  
27         double imagPart = real * o.imag + imag * o.real;  
28         return new Complex(realPart, imagPart);  
29     }  
30  
31     void print() {  
32         System.out.println(real + " + " + imag + "i");  
33     }  
34 }  
35 public class Ques9 {  
36     public static void main(String[] args) {  
37         Scanner sc = new Scanner(System.in);
```

Console X

```
<terminated> Ques9 [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64  
123  
123  
24  
123  
147.0 + 246.0i  
99.0 + 0.0i  
-12177.0 + 18081.0i
```

Ques 10.

```
Ques11.java Ques12.java Ques10.java X Ques1a.java Ques2a.java Ques3a.java Ques4a.java Ques5a.java  
1 //10. Write a program that would print the information (name, year of joining, sal  
2 //three employees by creating a class named 'Employee'. The output should be as fo  
3 //Name Year of joining Address  
4 //Robert 1994 64C- WallsStreat  
5 //Sam 2000 68D- WallsStreat  
6 //John 1999 26B- WallsStreat  
7  
8  
9 class Employee {  
10     String name;  
11     int yoj;  
12     String address;  
13  
14     Employee(String name, int yoj, String address) {  
15         this.name = name;  
16         this.yoj = yoj;  
17         this.address = address;  
18     }  
19     void print() {  
20         System.out.println(name + " " + yoj + " " + address);  
21     }  
22 }  
23  
24 public class Ques10 {  
25     public static void main(String args[]) {  
26         System.out.println("Name      Year of joining  Address");  
27         Employee em = new Employee("Robert", 1994, "64C- WallsStreat");  
28         Employee em1 = new Employee("Sam", 2000, "68D- WallsStreat");  
29         Employee em2 = new Employee("John", 1999, "26B- WallsStreat");  
30  
31         em.print();  
32         em1.print();  
33         em2.print();  
34     }  
35 }
```



```
Console X  
<terminated> Ques9 [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin  
123  
123  
24  
123  
147.0 + 246.0i  
99.0 + 0.0i  
-12177.0 + 18081.0i
```

Ques 11.

```
49
50 import java.util.Scanner;
51
52 class Clock {
53     private int hours;    // 0-23
54     private int minutes; // 0-59
55     private int seconds; // 0-59
56
57     // Default constructor: 12:00:00
58     public Clock() {
59         this.hours = 12;
60         this.minutes = 0;
61         this.seconds = 0;
62     }
63
64     // Constructor with hours, minutes, seconds
65     public Clock(int hours, int minutes, int seconds) {
66         setHours(hours);
67         setMinutes(minutes);
68         setSeconds(seconds);
69     }
70
71     // Constructor with seconds since midnight
72     public Clock(int secondsSinceMidnight) {
73         setClock(secondsSinceMidnight);
74     }
75 }
```

```
// setClock: set time from seconds since midnight
public void setClock(int secondsSinceMidnight) {
    if (secondsSinceMidnight < 0) secondsSinceMidnight = 0;
    this.hours = (secondsSinceMidnight / 3600) % 24;
    this.minutes = (secondsSinceMidnight % 3600) / 60;
    this.seconds = secondsSinceMidnight % 60;
}

// Getters
public int getHours() { return hours; }
public int getMinutes() { return minutes; }
public int getSeconds() { return seconds; }

// Setters
public void setHours(int hours) {
    this.hours = (hours >= 0 && hours < 24) ? hours : 0;
}
public void setMinutes(int minutes) {
    this.minutes = (minutes >= 0 && minutes < 60) ? minutes : 0;
}
public void setSeconds(int seconds) {
    this.seconds = (seconds >= 0 && seconds < 60) ? seconds : 0;
}
```

```
// Increment time by one second
public void tick() {
    seconds++;
    if (seconds == 60) {
        seconds = 0;
        minutes++;
        if (minutes == 60) {
            minutes = 0;
            hours = (hours + 1) % 24;
        }
    }
}

// Decrement time by one second
public void tickDown() {
    seconds--;
    if (seconds < 0) {
        seconds = 59;
        minutes--;
        if (minutes < 0) {
            minutes = 59;
            hours = (hours - 1 + 24) % 24;
        }
    }
}

// Add another clock's time to this clock
public void addClock(Clock other) {
    int totalSeconds = this.toSeconds() + other.toSeconds();
    setClock(totalSeconds % (24 * 3600)); // wrap around 24 hours
}
```

```
// Subtract another clock's time from this clock, return as new Clock
public Clock subtractClock(Clock other) {
    int diff = this.toSeconds() - other.toSeconds();
    if (diff < 0) diff += 24 * 3600; // wrap around
    return new Clock(diff);
}

// Convert current time to seconds since midnight
private int toSeconds() {
    return hours * 3600 + minutes * 60 + seconds;
}

// String representation
public String toString() {
    return String.format("%02d:%02d:%02d", hours, minutes, seconds);
}

/ Demo class
public class Ques11 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Instantiate firstClock using seconds since midnight
        System.out.print("Enter seconds since midnight for firstClock: ");
        int secSinceMidnight = sc.nextInt();
        Clock firstClock = new Clock(secSinceMidnight);

        // Tick firstClock 10 times
        System.out.println("Ticking firstClock 10 times:");
        for (int i = 0; i < 10; i++) {
            firstClock.tick();
```

```
// Instantiate secondClock using hours, minutes, seconds
System.out.print("Enter hours for secondClock: ");
int h = sc.nextInt();
System.out.print("Enter minutes for secondClock: ");
int m = sc.nextInt();
System.out.print("Enter seconds for secondClock: ");
int s = sc.nextInt();
Clock secondClock = new Clock(h, m, s);

// Tick secondClock 10 times
System.out.println("Ticking secondClock 10 times:");
for (int i = 0; i < 10; i++) {
    secondClock.tick();
    System.out.println(secondClock);
}

// Add secondClock to firstClock
firstClock.addClock(secondClock);

// Print both clocks
System.out.println("firstClock: " + firstClock);
System.out.println("secondClock: " + secondClock);

// Subtract secondClock from firstClock
Clock thirdClock = firstClock.subtractClock(secondClock);
System.out.println("thirdClock (firstClock - secondClock): " + thirdClock);

sc.close();
}
```

```
J Ques11.java J Ques10.java Ques1a.java X Ques2a.java Ques3a.java Ques4a.java J Ques5a.java
1 class Car {
2     String brand;
3     String model;
4     double price;
5
6     // Constructor 1: Only brand
7     Car(String brand) {
8         this.brand = brand;
9         System.out.println("Brand: " + brand);
10    }
11
12     // Constructor 2: Brand and model
13     Car(String brand, String model) {
14         this.brand; // Calls constructor 1
15         this.model = model;
16         System.out.println("Model: " + model);
17    }
18
19     // Constructor 3: Brand, model, and price
20     Car(String brand, String model, double price) {
21         this.brand, model); // Calls constructor 2
22         this.price = price;
23         System.out.println("Price: " + price);
24    }
25 }
26
27 public class Ques1a {
28     public static void main(String[] args) {
29         Car c1 = new Car("Toyota");
30         System.out.println("___");
31         Car c2 = new Car("Honda", "Civic");
32         System.out.println("___");
33         Car c3 = new Car("Ford", "Mustang", 35000.0);
34     }
35 }
```

Console X

```
<terminated> Ques1a [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724
Brand: Toyota
---
Brand: Honda
Model: Civic
---
Brand: Ford
Model: Mustang
Price: 35000.0
```

```
Ques11.java Ques10.java Ques1a.java Ques2a.java X Ques3a.java Ques4.java Ques5.java Ques7.java Ques3
1 class Person {
2     String name;
3     int age;
4
5     Person(String name, int age) {
6         this.name = name;
7         this.age = age;
8         System.out.println("Person constructor: Name = " + name + ", Age = " + age);
9     }
10 }
11
12 class Student1 extends Person {
13     String course;
14
15     Student1(String name, int age, String course) {
16         super(name, age); // Calls Person constructor
17         this.course = course;
18         System.out.println("Student constructor: Course = " + course);
19     }
20 }
21
22 public class Ques2a {
23     public static void main(String[] args) {
24         Student1 s = new Student1("Alice", 20, "Computer Science");
25     }
26 }
```

Console X

```
<terminated> Ques2a [Java Application] /Users/proxim/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java (Sep 13, 2025, 7:54:24 PM)
Person constructor: Name = Alice, Age = 20
Student constructor: Course = Computer Science
```

Ques 3a.

The screenshot shows the Eclipse IDE interface with the following details:

- Top Bar:** Shows various icons for file operations like Open, Save, Find, and Run.
- Toolbar:** Shows icons for New, Open, Save, Cut, Copy, Paste, and others.
- Project Explorer:** Shows multiple Java files: Ques11.java, Ques10.java, Ques1a.java, Ques2a.java, Ques3a.java (highlighted), Ques4a.java, Ques5a.java, and Ques7.java.
- Code Editor:** Displays the Java code for Ques3a.java. The code defines three classes: Vehicle, FourWheeler, and Car. The Vehicle class has a String type field and a constructor that prints its type. The FourWheeler class extends Vehicle and adds a String brand field, calling the Vehicle constructor and printing its brand. The Car class extends FourWheeler and adds a String model and double price field, calling the FourWheeler constructor and printing its model and price. A tooltip "FourWheeler" is visible over the class name in the editor.
- Console:** Shows the output of the program. It includes the terminal prompt <terminated>, the application name Ques2a [Java Application], the path /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java, and the date (Sep 13, 2025). The console also displays two lines of output: "Person constructor: Name = Alice, Age = 20" and "Student constructor: Course = Computer Science".

```
1 class Vehicle {
2     String type;
3
4     Vehicle(String type) {
5         this.type = type;
6         System.out.println("Vehicle constructor: Type = " + type);
7     }
8 }
9
10 class FourWheeler extends Vehicle {
11     String brand;
12
13     FourWheeler(String type, String brand) {
14         super(type); // Calls Vehicle constructor
15         this.brand = brand;
16         System.out.println("FourWheeler constructor: Brand = " + brand);
17     }
18 }
19
20 class Car extends FourWheeler {
21     String model;
22     double price;
23
24     Car(String type, String brand, String model, double price) {
25         super(type, brand); // Calls FourWheeler constructor
26         this.model = model;
27         this.price = price;
28         System.out.println("Car constructor: Model = " + model + ", Price = " + price);
29     }
30 }
31
32 public class Ques3a {
33     public static void main(String[] args) {
34         Car c = new Car("Automobile", "Toyota", "Camry", 25000.0);
35     }
36 }
```

```
Console X
<terminated> Ques2a [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java (Sep 13, 2025)
Person constructor: Name = Alice, Age = 20
Student constructor: Course = Computer Science
```

Ques 4a.

```
1 class Order {
2     int orderId;
3     String customerName;
4     double totalAmount;
5
6     // Constructor 1: Only orderId
7     Order(int orderId) {
8         this.orderId = orderId;
9         System.out.println("OrderId: " + orderId);
10    }
11
12     // Constructor 2: orderId and customerName
13     Order(int orderId, String customerName) {
14         this(orderId); // Calls constructor 1
15         this.customerName = customerName;
16         System.out.println("Customer Name: " + customerName);
17    }
18
19     // Constructor 3: orderId, customerName, and totalAmount
20     Order(int orderId, String customerName, double totalAmount) {
21         this(orderId, customerName); // Calls constructor 2
22         this.totalAmount = totalAmount;
23         System.out.println("Total Amount: " + totalAmount);
24    }
25 }
26
27 public class Ques4a {
28     public static void main(String[] args) {
29         Order o1 = new Order(101);
30         System.out.println("----");
31         Order o2 = new Order(102, "Alice");
32         System.out.println("----");
33         Order o3 = new Order(103, "Bob", 2500.50);
34     }
35 }
```

Console X

```
<terminated> Ques2a [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java (Sep
Person constructor: Name = Alice, Age = 20
Student constructor: Course = Computer Science
```

Ques 5a.

```
Ques11.java Ques10.java Ques1a.java Ques2a.java Ques3a.java Ques4a.java Ques5a.java X Ques7.java
3  String holderName;
4  double balance;
5
6  // Constructor 1: Only accountNumber
7@ BankAccount(int accountNumber) {
8      this.accountNumber = accountNumber;
9      System.out.println("Account Number: " + accountNumber);
10 }
11
12 // Constructor 2: accountNumber and holderName
13@ BankAccount(int accountNumber, String holderName) {
14     this(accountNumber); // Calls constructor 1
15     this.holderName = holderName;
16     System.out.println("Holder Name: " + holderName);
17 }
18
19 // Constructor 3: accountNumber, holderName, and balance
20@ BankAccount(int accountNumber, String holderName, double balance) {
21     this(accountNumber, holderName); // Calls constructor 2
22     this.balance = balance;
23     System.out.println("Balance: " + balance);
24 }
25
26@ void display() {
27     System.out.println("Account Number: " + accountNumber +
28                         ", Holder Name: " + holderName +
29                         ", Balance: " + balance);
30 }
31 }
32
33 public class Ques5a {
34@   public static void main(String[] args) {
35       BankAccount b1 = new BankAccount(1001);
36       b1.display();
37       System.out.println("___");
38       BankAccount b2 = new BankAccount(1002, "Alice");
39       b2.display();
40       System.out.println("___");
41       BankAccount b3 = new BankAccount(1003, "Bob", 5000.0);
42       b3.display();
43   }
44 }
```

Console X

```
<terminated> Ques5a [Java Application] /Users/proxim/p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_21.0.8.v20250724-1412/jre/bin/java
Account Number: 1001
Account Number: 1001, Holder Name: null, Balance: 0.0
---
Account Number: 1002
Holder Name: Alice
Account Number: 1002, Holder Name: Alice, Balance: 0.0
---
Account Number: 1003
Holder Name: Bob
Balance: 5000.0
Account Number: 1003, Holder Name: Bob, Balance: 5000.0
```