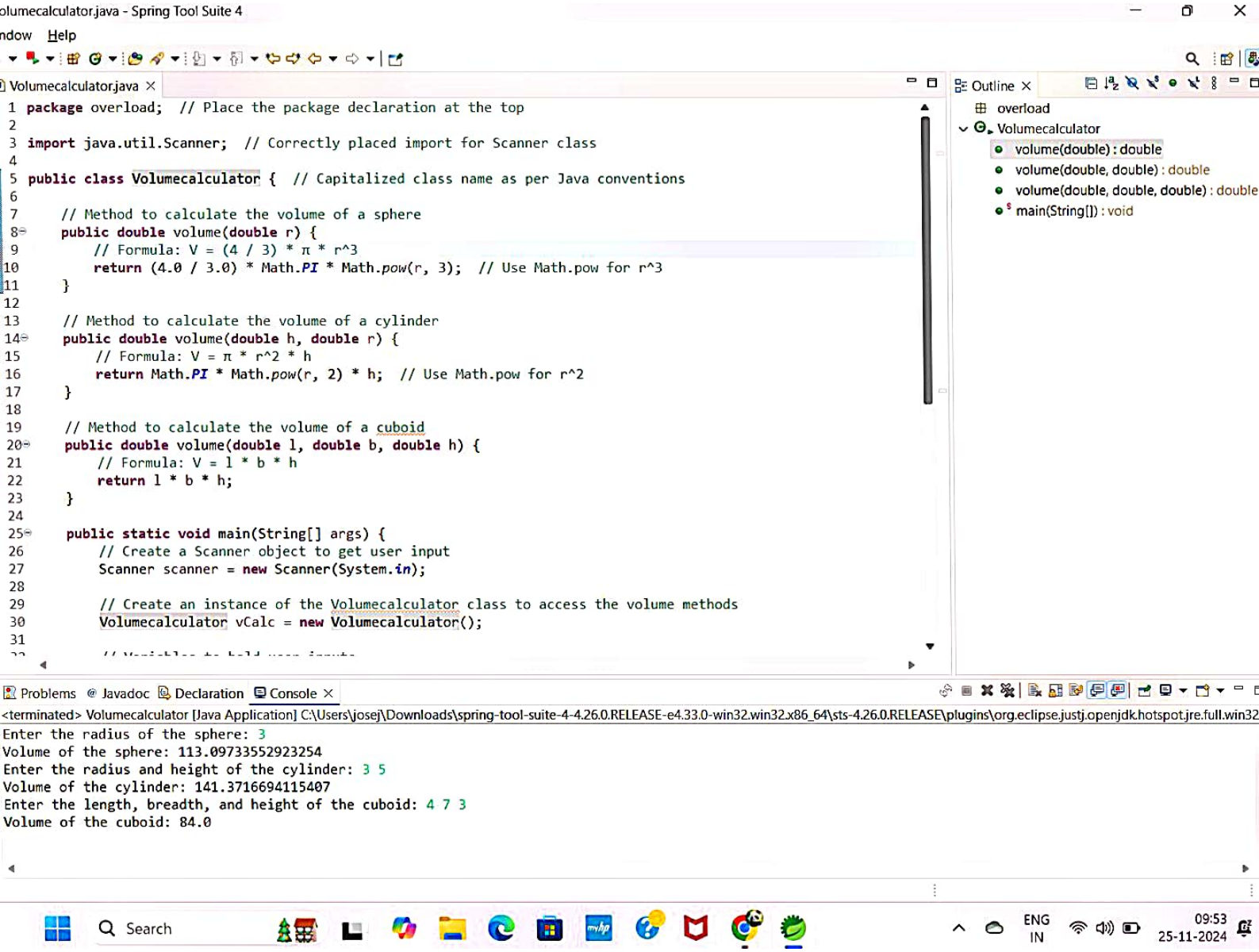
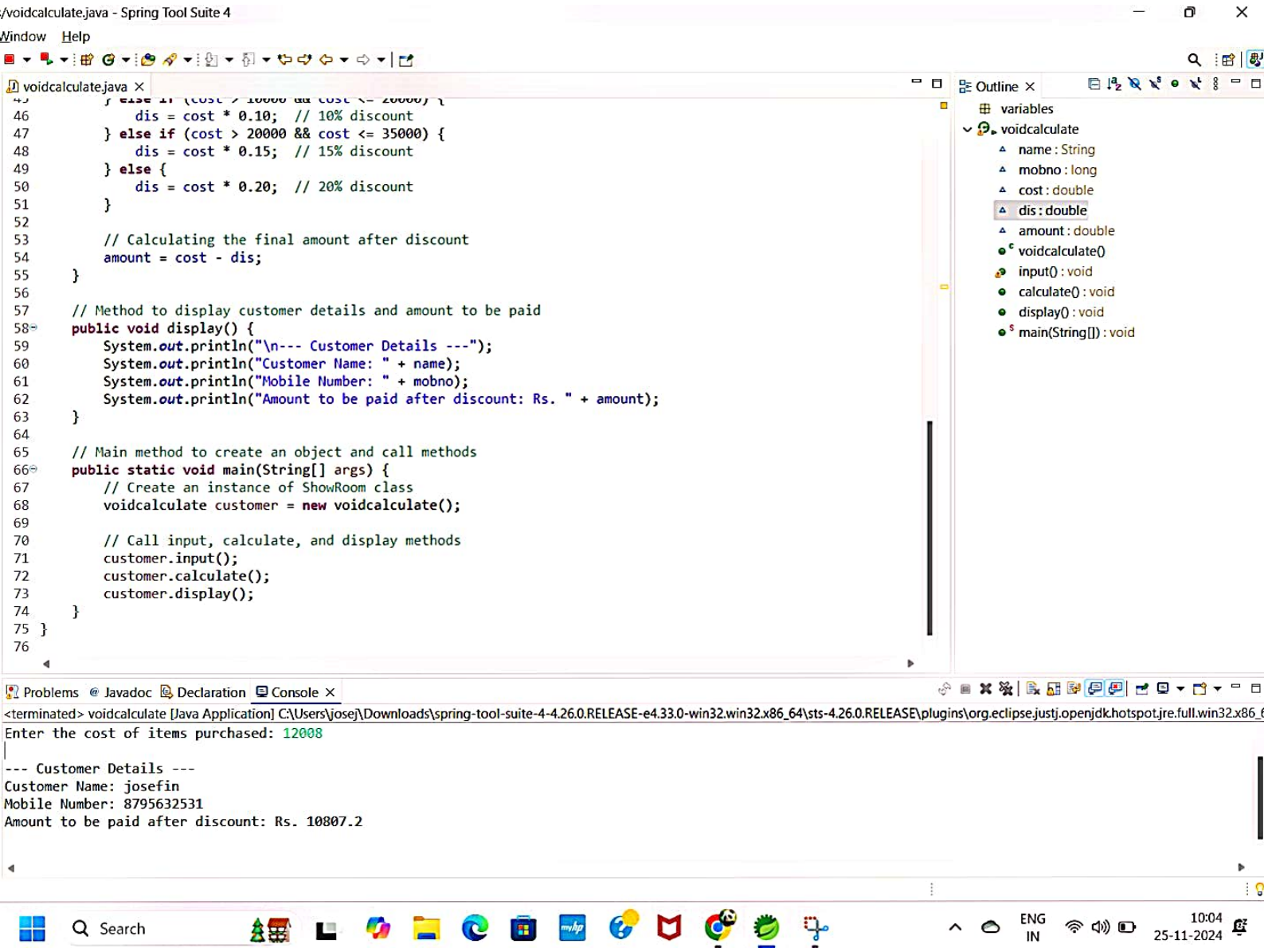
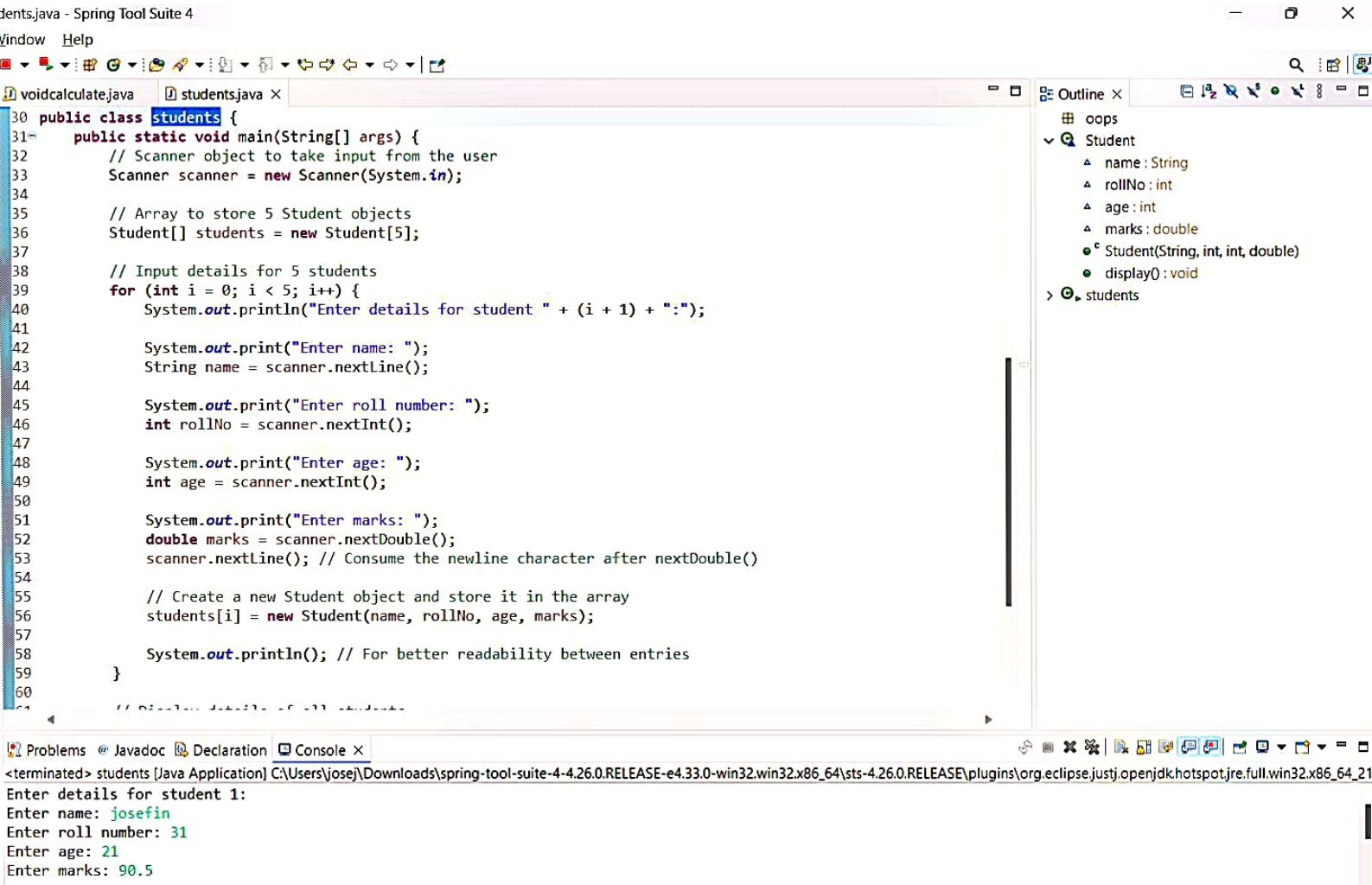
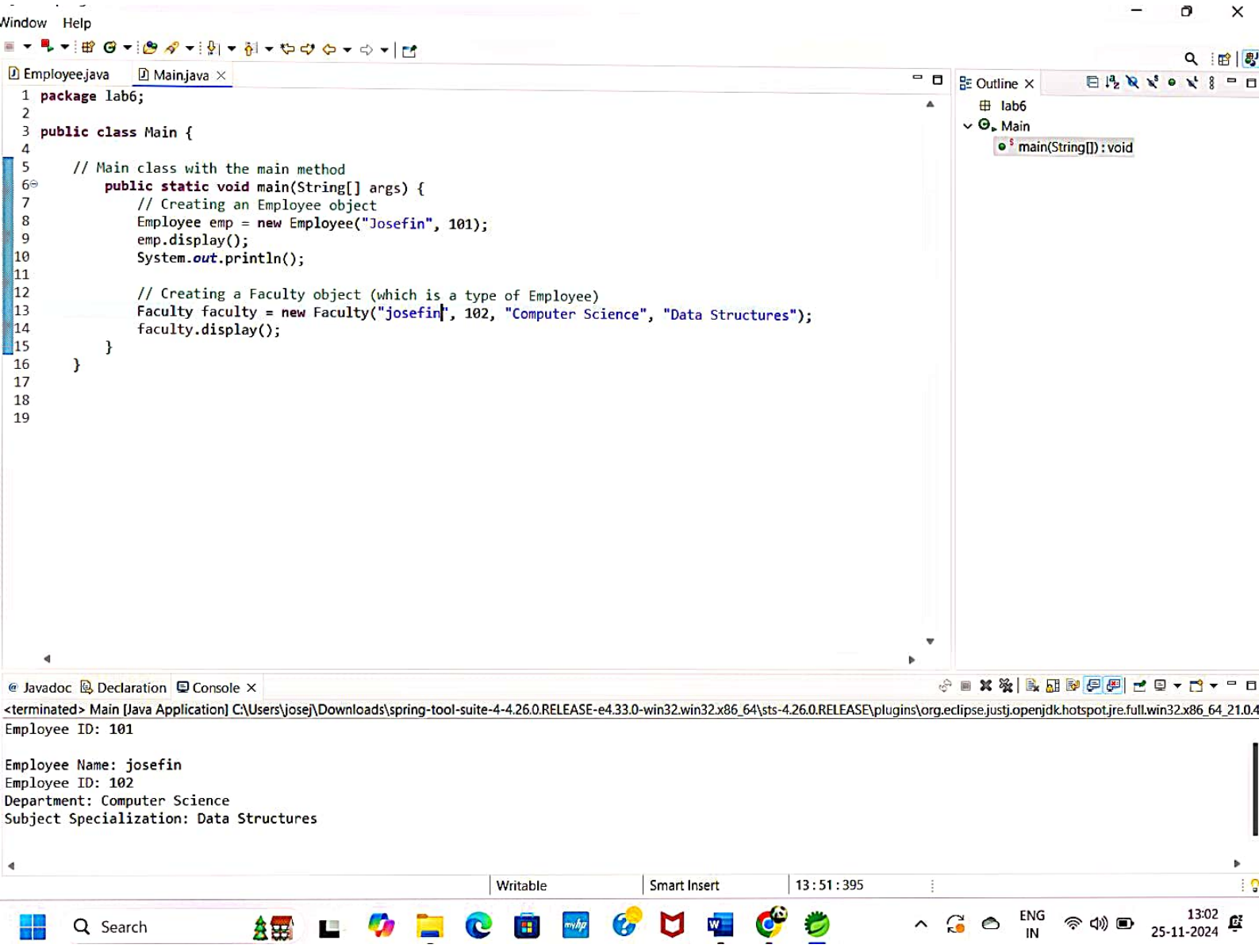

olumecalculator.java - Spring Tool Suite 4
ndow Help
Volumecalculator.java
1 package overload; // Place the package declaration at the top
2
3 import java.util.Scanner; // Correctly placed import for Scanner class
4
5 public class Volumecalculator { // Capitalized class name as per Java conventions
6
7 // Method to calculate the volume of a sphere
8 public double volume(double r) {
9 // Formula: V = (4 / 3) * π * r^3
10 return (4.0 / 3.0) * Math.PI * Math.pow(r, 3); // Use Math.pow for r^3
11 }
12
13 // Method to calculate the volume of a cylinder
14 public double volume(double h, double r) {
15 // Formula: V = π * r^2 * h
16 return Math.PI * Math.pow(r, 2) * h; // Use Math.pow for r^2
17 }
18
19 // Method to calculate the volume of a cuboid
20 public double volume(double l, double b, double h) {
21 // Formula: V = l * b * h
22 return l * b * h;
23 }
24
25 public static void main(String[] args) {
26 // Create a Scanner object to get user input
27 Scanner scanner = new Scanner(System.in);
28
29 // Create an instance of the Volumecalculator class to access the volume methods
30 Volumecalculator vCalc = new Volumecalculator();
31
Outline
overload
Volumecalculator
volume(double) : double
volume(double, double) : double
volume(double, double, double) : double
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> Volumecalculator [Java Application] C:\Users\josej\Downloads\spring-tool-suite-4-4.26.0.RELEASE-e4.33.0-win32.win32.x86_64\sts-4.26.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32
Enter the radius of the sphere: 3
Volume of the sphere: 113.09733552923254
Enter the radius and height of the cylinder: 3 5
Volume of the cylinder: 141.3716694115407
Enter the length, breadth, and height of the cuboid: 4 7 3
Volume of the cuboid: 84.0
Search
ENG
IN
09:53
25-11-2024


/voidcalculate.java - Spring Tool Suite 4
Window Help
voidcalculate.java
} else if (cost > 10000 && cost <= 20000) {
dis = cost * 0.10; // 10% discount
} else if (cost > 20000 && cost <= 35000) {
dis = cost * 0.15; // 15% discount
} else {
dis = cost * 0.20; // 20% discount
}
// Calculating the final amount after discount
amount = cost - dis;
}
// Method to display customer details and amount to be paid
public void display() {
System.out.println("\n--- Customer Details ---");
System.out.println("Customer Name: " + name);
System.out.println("Mobile Number: " + mobno);
System.out.println("Amount to be paid after discount: Rs. " + amount);
}
// Main method to create an object and call methods
public static void main(String[] args) {
// Create an instance of ShowRoom class
voidcalculate customer = new voidcalculate();
// Call input, calculate, and display methods
customer.input();
customer.calculate();
customer.display();
}
}
Outline
variables
voidcalculate
name : String
mobno : long
cost : double
dis : double
amount : double
voidcalculate()
input() : void
calculate() : void
display() : void
main(String[]) : void
Problems Javadoc Declaration Console
<terminated> voidcalculate [Java Application] C:\Users\josej\Downloads\spring-tool-suite-4-4.26.0.RELEASE-e4.33.0-win32.win32.x86_64\sts-4.26.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_6
Enter the cost of items purchased: 12008
--- Customer Details ---
Customer Name: josefin
Mobile Number: 8795632531
Amount to be paid after discount: Rs. 10807.2
Search
ENG
IN
10:04
25-11-2024




```
30 public class students {
31     public static void main(String[] args) {
32         // Scanner object to take input from the user
33         Scanner scanner = new Scanner(System.in);
34 
35         // Array to store 5 Student objects
36         Student[] students = new Student[5];
37 
38         // Input details for 5 students
39         for (int i = 0; i < 5; i++) {
40             System.out.println("Enter details for student " + (i + 1) + ":");
41 
42             System.out.print("Enter name: ");
43             String name = scanner.nextLine();
44 
45             System.out.print("Enter roll number: ");
46             int rollNo = scanner.nextInt();
47 
48             System.out.print("Enter age: ");
49             int age = scanner.nextInt();
50 
51             System.out.print("Enter marks: ");
52             double marks = scanner.nextDouble();
53             scanner.nextLine(); // Consume the newline character after nextDouble()
54 
55             // Create a new Student object and store it in the array
56             students[i] = new Student(name, rollNo, age, marks);
57 
58             System.out.println(); // For better readability between entries
59         }
60 
```


Window Help
Employee.java
Main.java
1 package lab6;
2
3 public class Main {
4
5 // Main class with the main method
6 public static void main(String[] args) {
7 // Creating an Employee object
8 Employee emp = new Employee("Josefin", 101);
9 emp.display();
10 System.out.println();
11
12 // Creating a Faculty object (which is a type of Employee)
13 Faculty faculty = new Faculty("josefin", 102, "Computer Science", "Data Structures");
14 faculty.display();
15 }
16 }
17
18
19
Outline
lab6
Main
main(String[]) : void
Javadoc
Declaration
Console
<terminated> Main [Java Application] C:\Users\josej\Downloads\spring-tool-suite-4-4.26.0.RELEASE-e4.33.0-win32.win32.x86_64\sts-4.26.0.RELEASE\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.4
Employee ID: 101
Employee Name: josefin
Employee ID: 102
Department: Computer Science
Subject Specialization: Data Structures
Writable
Smart Insert
13 : 51 : 395
Search
ENG
IN
13:02
25-11-2024