

Java Lab Assignment – 3

Praneesh R V

CB.SC.U4CYS23036

Qn1.

Code:



```
1  import java.util.Scanner;
2  class Wall {
3      private double length;
4      private double height;
5      Wall(double length, double height) {
6          this.length = length;
7          this.height = height;
8      }
9      public double calculateArea() {
10         return length * height;
11     }
12 }
13 public class qn1 {
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         System.out.println("Enter length of wall 1");
17         double length1 = sc.nextDouble();
18         System.out.println("Enter height of wall 1");
19         double height1 = sc.nextDouble();
20         Wall wall1 = new Wall(length1, height1);
21         System.out.println("Enter length of wall 2");
22         double length2 = sc.nextDouble();
23         System.out.println("Enter height of wall 2");
24         double height2 = sc.nextDouble();
25         Wall wall2 = new Wall(length2, height2);
26
27         System.out.printf(format:"Area of Wall 1: %.2f\n", wall1.calculateArea());
28         System.out.printf(format:"Area of Wall 2: %.2f\n", wall2.calculateArea());
29
30         sc.close();
31     }
32 }
```

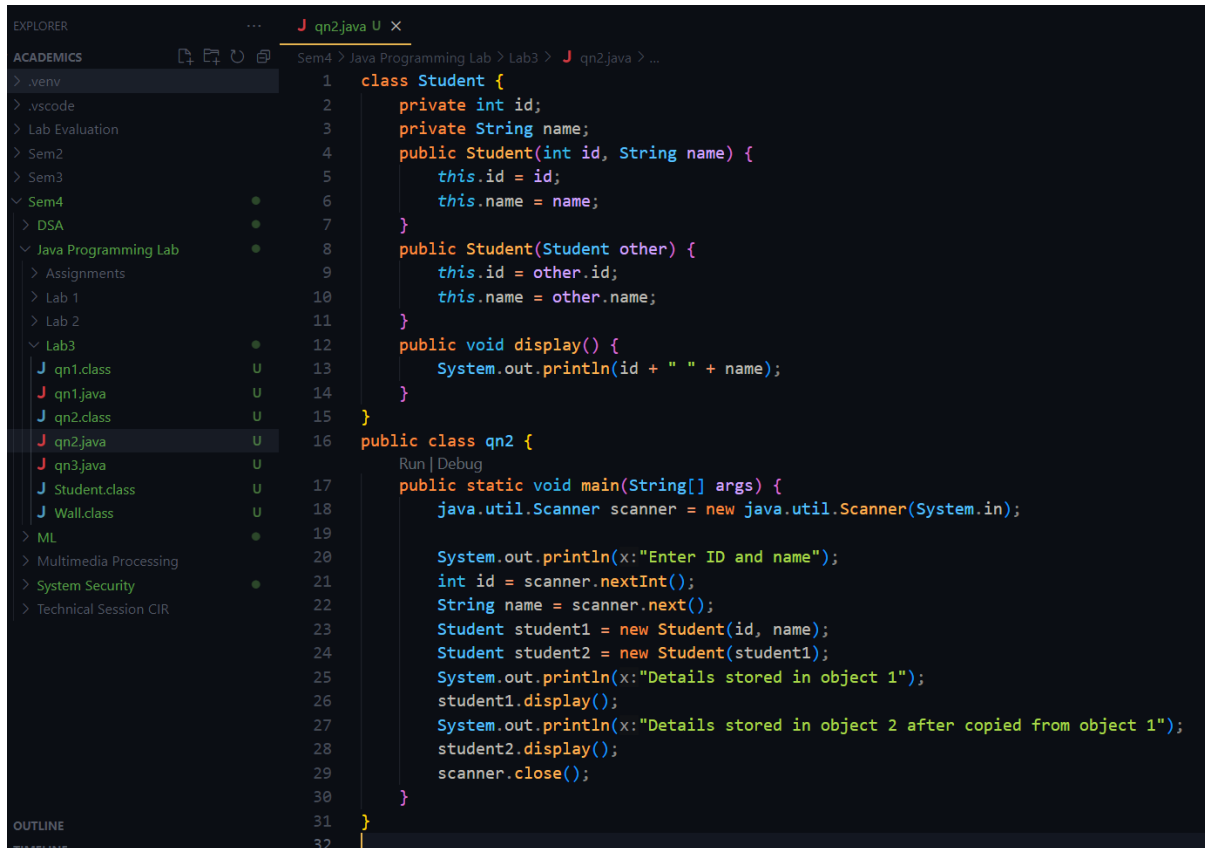
Output:



```
PS D:\Academics> cd "d:\Academics\Sem4\Java Programming Lab\Lab3\" ; if ($?) { javac qn1.java } ; if ($?) { java qn1 }
Enter length of wall 1
32
Enter height of wall 1
25
Enter length of wall 2
15
Enter height of wall 2
30
Area of Wall 1: 800.00
Area of Wall 2: 450.00
PS D:\Academics\Sem4\Java Programming Lab\Lab3>
```

Qn2,

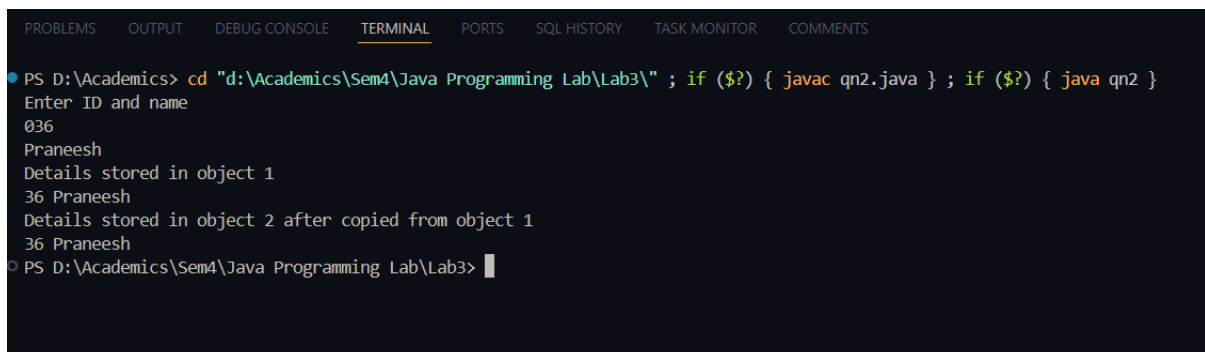
Code:



The screenshot shows the VS Code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project structure with folders like .venv, .vscode, Lab Evaluation, Sem2, Sem3, Sem4, DSA, Java Programming Lab, Assignments, Lab 1, Lab 2, Lab3, and files like qn1.class, qn1.java, qn2.class, qn2.java, qn3.java, Student.class, and Wall.class. The code editor shows the following Java code:

```
1 class Student {
2     private int id;
3     private String name;
4     public Student(int id, String name) {
5         this.id = id;
6         this.name = name;
7     }
8     public Student(Student other) {
9         this.id = other.id;
10        this.name = other.name;
11    }
12    public void display() {
13        System.out.println(id + " " + name);
14    }
15 }
16 public class qn2 {
17     public static void main(String[] args) {
18         java.util.Scanner scanner = new java.util.Scanner(System.in);
19
20         System.out.println(x:"Enter ID and name");
21         int id = scanner.nextInt();
22         String name = scanner.next();
23         Student student1 = new Student(id, name);
24         Student student2 = new Student(student1);
25         System.out.println(x:"Details stored in object 1");
26         student1.display();
27         System.out.println(x:"Details stored in object 2 after copied from object 1");
28         student2.display();
29         scanner.close();
30     }
31 }
32
```

Output:

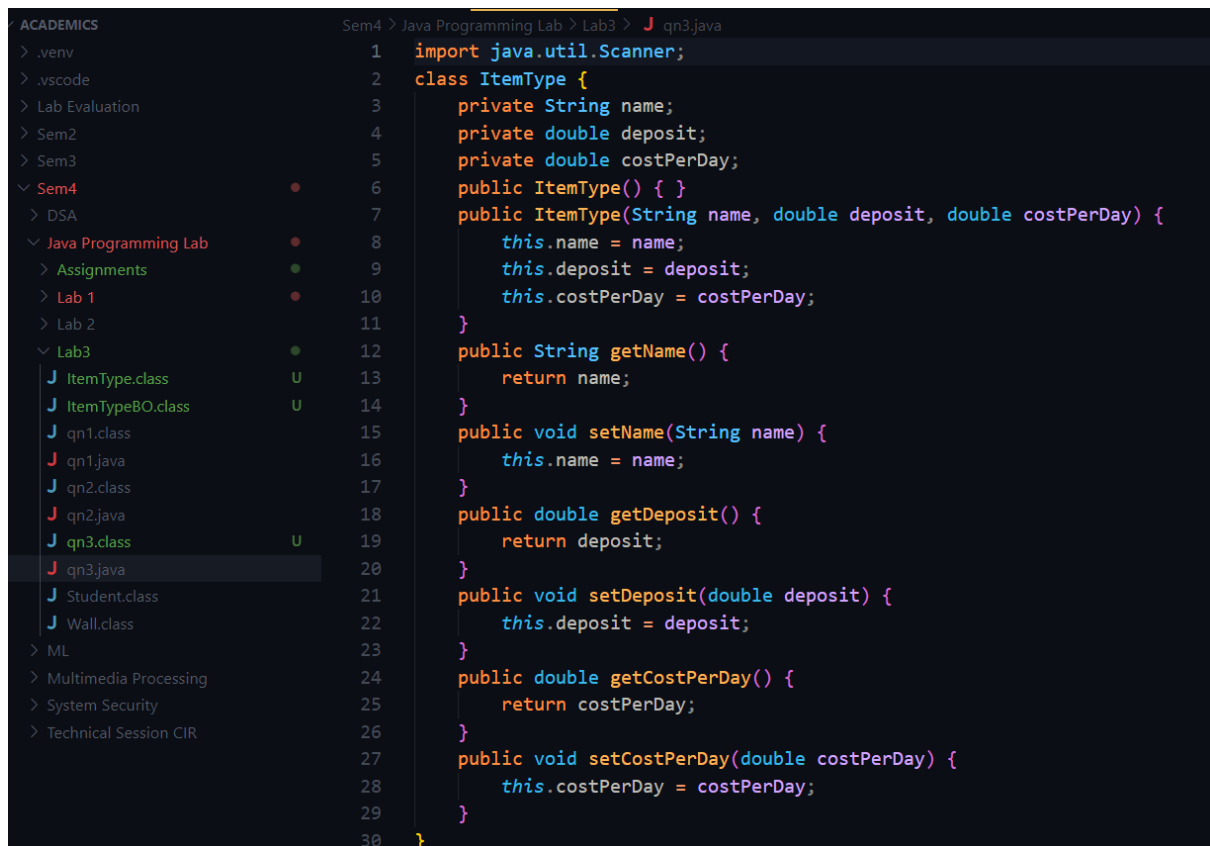


The screenshot shows the VS Code terminal with the following output:

```
PS D:\Academics> cd "d:\Academics\Sem4\Java Programming Lab\Lab3\" ; if ($?) { javac qn2.java } ; if ($?) { java qn2 }
Enter ID and name
036
Praneesh
Details stored in object 1
36 Praneesh
Details stored in object 2 after copied from object 1
36 Praneesh
PS D:\Academics\Sem4\Java Programming Lab\Lab3>
```

Qn3,

Code:



```
1  import java.util.Scanner;
2  class ItemType {
3      private String name;
4      private double deposit;
5      private double costPerDay;
6      public ItemType() { }
7      public ItemType(String name, double deposit, double costPerDay) {
8          this.name = name;
9          this.deposit = deposit;
10         this.costPerDay = costPerDay;
11     }
12     public String getName() {
13         return name;
14     }
15     public void setName(String name) {
16         this.name = name;
17     }
18     public double getDeposit() {
19         return deposit;
20     }
21     public void setDeposit(double deposit) {
22         this.deposit = deposit;
23     }
24     public double getCostPerDay() {
25         return costPerDay;
26     }
27     public void setCostPerDay(double costPerDay) {
28         this.costPerDay = costPerDay;
29     }
30 }
```




```
31 class ItemTypeBO {
32     public void searchDetail(String search, ItemType[] item, int n) {
33         boolean found = false;
34         for(int i = 0; i < n; i++) {
35             if(item[i].getName().equals(search)) {
36                 System.out.printf(format: "%s %.1f %.1f\n", item[i].getName(), item[i].getDeposit(), item[i].getCostPerDay());
37                 found = true;
38                 break;
39             }
40         }
41         if(!found) {
42             System.out.println(x: "Searched item Type not found");
43         }
44     }
45 }
46
47 public void display(ItemType[] item, int n) {
48     for(int i = 0; i < n; i++) {
49         System.out.printf(format: "%s %.1f %.1f\n", item[i].getName(), item[i].getDeposit(), item[i].getCostPerDay());
50     }
51 }
52
53 public class qn3 {
54     Run | Debug
55     public static void main(String[] args) {
56         Scanner sc = new Scanner(System.in);
57         ItemType[] items = new ItemType[10];
58     }
59 }
```



```
53 public class qn3 {
54     public static void main(String[] args) {
55         Scanner sc = new Scanner(System.in);
56         ItemType[] items = new ItemType[10];
57
58         int n = sc.nextInt();
59         for(int i = 0; i < n; i++) {
60             String name = sc.next();
61             double deposit = sc.nextDouble();
62             double costPerDay = sc.nextDouble();
63             items[i] = new ItemType(name, deposit, costPerDay);
64         }
65
66         String search = sc.next();
67         ItemTypeBO bo = new ItemTypeBO();
68         bo.searchDetail(search, items, n);
69         bo.display(items, n);
70
71         sc.close();
72     }
73 }
74
```

Output:



```
PS D:\Academics> cd "d:\Academics\Sem4\Java Programming Lab\Lab3\" ; if ($?) { javac qn3.java } ; if ($?) { java qn3 }
3
electronics
2500
150
chemicals
500
50
construction
2000
500
construction
construction 2000.0 500.0
electronics 2500.0 150.0
chemicals 500.0 50.0
construction 2000.0 500.0
```

Qn4,

Code:

The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like .env, .vscode, Lab Evaluation, Sem2, Sem3, Sem4, DSA, Java Programming Lab, Assignments, Lab 1, Lab 2, Lab 3, and files like ItemType.class, ItemTypeBO.class, qn1.class, qn1.java, qn2.class, qn2.java, qn3.class, qn3.java, qn4.java, Student.class, and Wall.class. The code editor shows the StallCategory class definition in qn4.java, which includes a Scanner import, class definition with attributes name and detail, and methods for constructor, getters, and setters.

```
1 import java.util.Scanner;
2 class StallCategory {
3     public String name;
4     public String detail;
5
6     public StallCategory() {
7         name = "";
8         detail = "";
9     }
10    public StallCategory(String name, String detail) {
11        this.name = name;
12        this.detail = detail;
13    }
14    public StallCategory(StallCategory object) {
15        this.name = object.name;
16        this.detail = object.detail;
17    }
18    public String getName() {
19        return name;
20    }
21    public String getDetail() {
22        return detail;
23    }
24    public void setName(String name) {
25        this.name = name;
26    }
27    public void setDetail(String detail) {
28        this.detail = detail;
29    }
30 }
31
```

The screenshot shows the VS Code editor with the file explorer on the left and the code editor on the right. The file explorer shows a project structure with folders like Sem2, Sem3, Sem4, DSA, Java Programming Lab, Assignments, Lab 1, Lab 2, Lab 3, and files like ItemType.class, ItemTypeBO.class, qn1.class, qn1.java, qn2.class, qn2.java, qn3.class, qn3.java, qn4.java, Student.class, and Wall.class. The code editor shows the main method in qn4.java, which uses a Scanner to read input and creates StallCategory objects using both parameterized and copy constructors.

```
29 }
30 }
31
32 public class qn4 {
33     Run | Debug
34     public static void main(String[] args) {
35         Scanner sc = new Scanner(System.in);
36
37         String name = sc.nextLine();
38         String detail = sc.nextLine();
39
40         StallCategory stall1 = new StallCategory(name, detail);
41
42         StallCategory stall2 = new StallCategory(stall1);
43
44         System.out.println(x:"Using Parameterized Constructor:");
45         System.out.println("Name:" + stall1.getName());
46         System.out.println("Detail:" + stall1.getDetail());
47
48         System.out.println(x:"\nUsing Copy Constructor:");
49         System.out.println("Name:" + stall2.getName());
50         System.out.println("Detail:" + stall2.getDetail());
51
52         sc.close();
53     }
54 }
55
```

Output:

```
PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS SQL HISTORY TASK MONITOR COMMENTS
PS D:\Academics> cd "d:\Academics\Sem4\Java Programming Lab\Lab3\" ; if ($?) { javac qn4.java } ; if ($?) { java qn4 }
Harry Potter
Book
Using Parameterized Constructor:
Name:Harry Potter
Detail:Book

Using Copy Constructor:
Name:Harry Potter
Detail:Book
PS D:\Academics\Sem4\Java Programming Lab\Lab3> |
```

Qn5,

Code:

```
EXPLORER qn5.java U x
ACADEMICS Sem4 > Java Programming Lab > Lab3 > J qn5.java > ...
> .venv
> .vscode
> Lab Evaluation
> Sem2
> Sem3
> Sem4
> DSA
> Java Programming Lab
> Assignments
> Lab 1
> Lab 2
> Lab3
J ItemType.class U
J ItemTypeBO.class U
J Overloading.class U
J qn1.class
J qn1.java
J qn2.class
J qn2.java
J qn3.class U
J qn3.java
J qn4.class U
J qn4.java U
J qn5.class U
J qn5.java U
J StallCategory.class U
J Student.class
J Wall.class

1 import java.util.Scanner;
2
3 class Overloading {
4     private String name;
5     private String day;
6     private int temp;
7
8     public Overloading() {
9         this.name = "Argentina";
10        this.day = "Yesterday";
11        this.temp = 29;
12    }
13
14    public Overloading(String name, int temp) {
15        this.name = name;
16        this.day = "Today";
17        this.temp = temp;
18    }
19
20    public Overloading(String name, String day, int temp) {
21        this.name = name;
22        this.day = day;
23        this.temp = temp;
24    }
25
26    public void displayDetails() {
27        System.out.println(name + " " + day + " Temperature: " + temp + "\u00B0");
28    }
29 }
```

File Explorer	Line Numbers	Code
> Lab Evaluation	30	Run Debug
> Sem2	31	<code>public static void main(String[] args) {</code>
> Sem3	32	<code>Scanner scanner = new Scanner(System.in);</code>
> Sem4	33	
> DSA	34	<code>String[] input1 = scanner.nextLine().split(regex:" ");</code>
> Java Programming Lab	35	<code>String place1 = input1[0];</code>
> Assignments	36	<code>int temp1 = Integer.parseInt(input1[1]);</code>
> Lab 1	37	
> Lab 2	38	<code>String[] input2 = scanner.nextLine().split(regex:" ");</code>
> Lab3	39	<code>String place2 = input2[0];</code>
J ItemType.class	40	<code>String day2 = input2[1];</code>
J ItemTypeBO.class	41	<code>int temp2 = Integer.parseInt(input2[2]);</code>
J Overloading.class	42	
J qn1.class	43	<code>Overloading obj1 = new Overloading();</code>
J qn1.java	44	<code>Overloading obj2 = new Overloading(place1, temp1);</code>
J qn2.class	45	<code>Overloading obj3 = new Overloading(place2, day2, temp2);</code>
J qn2.java	46	
J qn3.class	47	<code>obj1.displayDetails();</code>
J qn3.java	48	<code>obj2.displayDetails();</code>
J qn4.class	49	<code>obj3.displayDetails();</code>
J qn4.java	50	
J qn5.class	51	<code>scanner.close();</code>
J qn5.java	52	<code>}</code>
J StallCategory.class	53	
J Student.class		

Output:

Terminal
PS D:\Academics> cd "d:\Academics\Sem4\Java Programming Lab\Lab3\" ; if (\$?) { javac qn5.java } ; if (\$?) { java qn5 }
Coimbatore
27
Chennai
Today
37
Argentina Yesterday Temperature: 29°
Coimbatore Today Temperature: 27°
Chennai Today Temperature: 37°
PS D:\Academics\Sem4\Java Programming Lab\Lab3>