Se. Arivumathi

CB.SC.U4CYS23004

LAB – 3

1)

```java
import java.util.Scanner;

class Wall{
    private double length;
    private double height;

    public Wall (double length, double height){
        this.length = length;
        this.height = height;
    }
    public double calculateArea(){
        return length * height;
    }
}
public class WallAreaCalculator{
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print(s:"Enter the length of the first wall: ");
        double length1 = scanner.nextDouble();
        System.out.print(s:"Enter the height of the first wall: ");
        double height1 = scanner.nextDouble();
        Wall wall1 = new Wall(length1, height1);
        double area1 = wall1.calculateArea();
        System.out.print("Area of the first wall is: " + area1 + "\n");
        System.out.print(s:"Enter the length of the second wall: ");
        double length2 = scanner.nextDouble();
        System.out.print(s:"Enter the height of the second wall: ");
        double height2 = scanner.nextDouble();
        Wall wall2 = new Wall (length2, height2);
        double area2 = wall2.calculateArea();
        System.out.print("Area of the second wall is: " + area2);

        scanner.close();
    }
}
```

```
C:\Users\arivu\Desktop\2nd year 4th sem\Java programming\Java programming lab 2> cm
deDetailsInExceptionMessages -cp "C:\Users\arivu\AppData\Roaming\Code\User\workspac
lab 2_68c5ffa1\bin" WallAreaCalculator "
Enter the length of the first wall: 20
Enter the height of the first wall: 30
Area of the first wall is: 600.0
Enter the length of the second wall: 10
Enter the height of the second wall: 20
Area of the second wall is: 200.0
```

2)

```java
import java.util.Scanner;

class Person{
    private String name;
    private int age;
    public Person(String name, int age){
        this.name = name;
        this.age = age;
    }
    public Person (Person another){
        this.name = another.name;
        this.age = another.age;
    }
    public void display(){
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

public class CopyConstructor {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s:"Enter name: ");
        String name = scanner.nextLine();
        System.out.print(s:"Enter the age: ");
        int age = scanner.nextInt();
        Person originalPerson = new Person(name, age);
        System.out.println(x:"Original Person: ");
        originalPerson.display();
        Person copiedPerson = new Person(originalPerson);
        System.out.println(x:"Copied Person: ");
        copiedPerson.display();

        scanner.close();
    }
}
```

```
C:\Users\arivu\Desktop\2nd year 4th sem\Java programming\Java programming lab 2> cmd /C ""C:\
deDetailsInExceptionMessages -cp "C:\Users\arivu\AppData\Roaming\Code\User\workspaceStorage\4
lab 2_68c5ffa1\bin" CopyConstructor "
Enter name: Arivumathi
Enter the age: 19
Original Person:
Name: Arivumathi, Age: 19
Copied Person:
Name: Arivumathi, Age: 19
```

3)

```java
import java.util.Scanner;

class ItemType {
    private String name;
    private double deposit;
    private double costPerDay;
    public ItemType() {
    }
    public ItemType(String name, double deposit, double costPerDay) {
        this.name = name;
        this.deposit = deposit;
        this.costPerDay = costPerDay;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public double getDeposit() {
        return deposit;
    }
    public void setDeposit(double deposit) {
        this.deposit = deposit;
    }
    public double getCostPerDay() {
        return costPerDay;
    }
    public void setCostPerDay(double costPerDay) {
        this.costPerDay = costPerDay;
    }
}
class ItemTypeBO {
    public void searchDetail(String search, ItemType[] items, int n) {
        boolean found = false;
        for (int i = 0; i < n; i++) {
            if (items[i].getName().equalsIgnoreCase(search)) {
                System.out.printf(format:"Name: %s\nDeposit: %.1f\nCost Per Day: %.1f\n",
                items[i].getName(), items[i].getDeposit(), items[i].getCostPerDay());
                found = true;
                break;
            }
        }
        if (!found) {
            System.out.println(x:"Item not found");
        }
    }
    public void display(ItemType[] items, int n) {
        for (int i = 0; i < n; i++) {
            System.out.printf(format:"Name: %s\nDeposit: %.1f\nCost Per Day: %.1f\n",
            items[i].getName(), items[i].getDeposit(), items[i].getCostPerDay());
        }
    }
}
public class Main {
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        ItemType[] items = new ItemType[10];
        int count = 0;
        ItemTypeBO itemTypeBO = new ItemTypeBO();

        while (true) {
            System.out.println(x:"1. Add Item\n2. Search Item\n3. Display Items\n4. Exit");
            System.out.print(s:"Enter your choice: ");
            int choice = scanner.nextInt();
            scanner.nextLine();
            switch (choice) {
                case 1:
                    if (count >= 10) {
                        System.out.println(x:"Array is full. Cannot add more items.");
```

```
68                  case 1:
69                      if (count >= 10) {
70                          System.out.println(x:"Array is full. Cannot add more items.");
71                      } else {
72                          System.out.print(s:"Enter name: ");
73                          String name = scanner.nextLine();
74                          System.out.print(s:"Enter deposit: ");
75                          double deposit = scanner.nextDouble();
76                          System.out.print(s:"Enter cost per day: ");
77                          double costPerDay = scanner.nextDouble();
78                          items[count] = new ItemType(name, deposit, costPerDay);
79                          count++;
80                          System.out.println(x:"Item added successfully.");
81                      }
82                      break;
83                  case 2:
84                      System.out.print(s:"Enter the Name of the item to be searched: ");
85                      String searchName = scanner.nextLine();
86                      itemTypeBO.searchDetail(searchName, items, count);
87                      break;
88                  case 3:
89                      itemTypeBO.display(items, count);
90                      break;
91                  case 4:
92                      System.out.println(x:"Exiting...");
93                      scanner.close();
94                      return;
95                  default:
96                      System.out.println(x:"Invalid choice. Please try again.");
97              }
98          }
99      }
100 }
101
```

```
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 1
Enter name: Arivumathi
Enter deposit: 2000
Enter cost per day: 100
Item added successfully.
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 1
Enter name: Jeevan
Enter deposit: 1000
Enter cost per day: 300
Item added successfully.
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 2
Enter the Name of the item to be searched: Sanjay
Item not found
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 2
Enter the Name of the item to be searched: Arivumathi
Name: Arivumathi
Deposit: 2000.0
Cost Per Day: 100.0
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 2
Enter the Name of the item to be searched: Arivumathi
Name: Arivumathi
Deposit: 2000.0
Cost Per Day: 100.0
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 3
Name: Arivumathi
Deposit: 2000.0
Cost Per Day: 100.0
Name: Jeevan
Deposit: 1000.0
Cost Per Day: 300.0
1. Add Item
2. Search Item
3. Display Items
4. Exit
Enter your choice: 4
Exiting...
```

4)

```java
import java.util.Scanner;

class StallCategory{
    public String name;
    public String detail;

    public StallCategory(){
        this.name = "Default Name";
        this.detail = "Default Detail";
    }

    public StallCategory(String name, String detail){
        this.name = name;
        this.detail = detail;
    }

    public StallCategory (StallCategory object){
        this.name = object.name;
        this.detail = object.detail;
    }

    public String getName(){
        return name;
    }

    public void setName(String name){
        this.name = name;
    }

    public String getDetail(){
        return detail;
    }

    public void setDetail(String detail){
        this.detail = detail;
    }
}
```

```java
}

public class Main1{
    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        StallCategory defaultStall = new StallCategory();

        System.out.println(x:"Using Default Constructor: ");
        System.out.println("Name: " + defaultStall.getName());
        System.out.println("Details: " + defaultStall.getDetail());

        Scanner scanner = new Scanner(System.in);
        System.out.println(x:"Enter the Stall name: ");
        String name = scanner.nextLine();
        System.out.println(x:"Enter Stall details: ");
        String detail = scanner.nextLine();

        StallCategory parameterizedStall = new StallCategory(name, detail);

        System.out.println(x:"Using parameterized Constructor: ");
        System.out.println("Name: " + parameterizedStall.getName());
        System.out.println("Detail: " + parameterizedStall.getDetail());

        StallCategory copiedStall = new StallCategory(parameterizedStall);

        System.out.println(x:"\nUsing copy constructor: ");
        System.out.println("Name: " + copiedStall.getName());
        System.out.println("Details" + copiedStall.getDetail());

        scanner.close();
    }
}
```

```
Using Default Constructor:
Name: Default Name
Details: Default Detail
Enter the Stall name:
Arivumathi
Enter Stall details:
Cutlet
Using parameterized Constructor:
Name: Arivumathi
Detail: Cutlet

Using copy constructor:
Name: Arivumathi
Details: Cutlet
```

5)

```java
import java.util.Scanner;

public class Overloading{
    private String name;
    private String day;
    private int temp;

    public Overloading(){
        this.name = "Argentina";
        this.day = "";
        this.temp = 29;
        System.out.println("Default constructor called: " + name + ", " + day + ", " + temp);
    }

    public Overloading(String name, int temp){
        this.name = name;
        this.day = "";
        this.temp = temp;
        System.out.println("Parameterized constructor 1 called: " + name + ", " + day + ", " + temp);
    }

    public Overloading(String name, String day, int temp){
        this.name = name;
        this.day = day;
        this.temp = temp;
        System.out.println("Parameterized constructor 2 called: " + name + ", " + day + ", " + temp);
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        Overloading obj1 = new Overloading();

        System.out.println(x:"Enter the name and the temperature for parameterized constructor 1: ");
        String name1 = scanner.nextLine();
        int temp1 = scanner.nextInt();

        int temp1 = scanner.nextInt();

        scanner.nextLine();

        Overloading obj2 = new Overloading(name1, temp1);

        System.out.println(x:"Enter the name, day and temperature for parameterized constructor 2: ");
        String name2 = scanner.nextLine();
        String day2 = scanner.nextLine();
        int temp2 = scanner.nextInt();

        scanner.nextLine();

        Overloading obj3 = new Overloading(name2, day2, temp2);

        scanner.close();
    }
}
```

```
Default constructor called: Argentina, , 29
Enter the name and the temperature for parameterized constructor 1:
India
20
Parameterized constructor 1 called: India, , 20
Enter the name, day and temperature for parameterized constructor 2:
America
Monday
18
Parameterized constructor 2 called: America, Monday, 18
```