Name:	Debjit Ghosal
UID:	2023300065
Experiment No.	4A

AIM:	Write a program to demonstrate arrays of objects
Program 1	
PROBLEM STATEMENT :	Write a program in Java to maintain the information of Movies which includes the information of name of movie , type of movie(action , thriller , comedy ,drama) , Hero name , Heroine , budget in Rs. Write a program to accept the information of movies from user and sort them according to the budget of the film.

PROGRAM:	Import java.util.*;
	public class Hotel {
	  private int hotelId;
	  private String location;
	  private int[] rooms; // Array format: [suites, deluxe, standard]
	
	  public Hotel(int hotelId, String location, int[] rooms) {
	  this.hotelId = hotelId;
	  this.location = location;
	  this.rooms = rooms;
	  }
	
	  public int getHotelId() {
	  return hotelId;
	  }
	
	  public String getLocation() {
	  return location;
	  }
	
	  public int[] getRooms() {
	  return rooms;
	  }
	
	  public static void listHotels(Hotel[] hotels, String location) {

```
        System.out.println("Hotels in " + location + ":");
        for (Hotel hotel : hotels) {
            if (hotel.getLocation().equalsIgnoreCase(location)) {
                System.out.println("Hotel ID: " + hotel.getHotelId() + ", Location: " +
hotel.getLocation() + ", Rooms: " + arrayToString(hotel.getRooms()));
            }
        }
    }

    public static void listHotels(Hotel[] hotels, int roomType) {
        String[] roomTypes = {"Suite", "Deluxe", "Standard"};
        System.out.println("Total number of rooms of type " +
roomTypes[roomType] + " in each hotel:");
        for (Hotel hotel : hotels) {
            System.out.println("Hotel ID: " + hotel.getHotelId() + ", Location: " +
hotel.getLocation() + ", " + roomTypes[roomType] + " rooms: " +
hotel.getRooms()[roomType]);
        }
    }

    private static String arrayToString(int[] array) {
        StringBuilder stringBuilder = new StringBuilder();
        stringBuilder.append("[");
        for (int i = 0; i < array.length; i++) {
            stringBuilder.append(array[i]);
            if (i < array.length - 1) {
                stringBuilder.append(", ");
            }
        }
        stringBuilder.append("]");
        return stringBuilder.toString();
    }

    public static void main(String[] args) {
        // Creating hotels
        Hotel[] hotels = new Hotel[3];
        hotels[0] = new Hotel(1, "Mumbai", new int[]{20, 40, 60});
        hotels[1] = new Hotel(2, "Delhi", new int[]{30, 50, 70});
        hotels[2] = new Hotel(3, "Bangalore", new int[]{25, 45, 65});
```

```
        // Listing hotels by location
        Hotel.listHotels(hotels, "Mumbai");

        // Listing total number of rooms of a certain type in each hotel
        Hotel.listHotels(hotels, 1); // Type 1 represents "Deluxe" rooms
    }
}
```

RESULT:

```
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ javac Hotel.java
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java Hotel
Hotels in Mumbai:
Hotel ID: 1, Location: Mumbai, Rooms: [20, 40, 60]
Total number of rooms of type Deluxe in each hotel:
Hotel ID: 1, Location: Mumbai, Deluxe rooms: 40
Hotel ID: 2, Location: Delhi, Deluxe rooms: 50
Hotel ID: 3, Location: Bangalore, Deluxe rooms: 45
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$
```

Program 2

PROBLEM STATEMENT :

Your swim school has two swimming instructors, Jeff and Anna. Their current schedules are shown below. An “X” denotes a one-hour time slot that is occupied with a lesson.

Write a program with array(s) capable of storing the schedules. Create a main menu that allows the user to mark a time slot as busy or free for either instructor. Also, add an option to output the schedules to the screen. Next, add an option to output all time slots available for individual lessons (slots when at least one instructor is free). Finally, add an option to output all time slots available for group lessons (when both instructors are free).

PROGRAM:

```
import java.util.Scanner;

class SwimSchool {
 int[][] jeffSchedule = new int[4][3];
 int[][] annaSchedule = new int[4][3];
 Scanner sc = new Scanner(System.in);
```

```
public static void main(String[] args) {
 SwimSchool swimSchool = new SwimSchool();
 swimSchool.start();
}

void start() {
 int choice;
 do {
  printMenu();
  choice = getUserChoice();
  switch (choice) {
   case 1:
    markBusy();
    break;
   case 2:
    markFree();
    break;
   case 3:
    printSchedules();
    break;
   case 4:
    showIndividualAvailability();
    break;
   case 5:
    showGroupAvailability();
    break;
   case 6:
    System.out.println("Exiting...");
    break;
   default:
    System.out.println("Invalid choice. Please enter a number between 1 and 6.");
  }
 } while (choice != 6);
}

void printMenu() {
 System.out.println("\nMenu:");
 System.out.println("1. Schedule a lesson");
 System.out.println("2. Cancel a lesson");
 System.out.println("3. View schedules");
```

```
 System.out.println("4. Show available slots for individual lessons");
 System.out.println("5. Show available slots for group lessons");
 System.out.println("6. Exit");
}

int getUserChoice() {
 System.out.print("Enter your choice: ");
 return sc.nextInt();
}

void markBusy() {
 System.out.println("Which instructor?");
 System.out.println("1. Jeff");
 System.out.println("2. Anna");
 int instructor = sc.nextInt();

 System.out.println("Which day?");
 int day = sc.nextInt() - 1;

 System.out.println("Which hour?");
 int hour = sc.nextInt() - 1;
 if (instructor == 1)
  jeffSchedule[day][hour] = 1;
 else if (instructor == 2)
  annaSchedule[day][hour] = 1;
 else
  System.out.println("Invalid choice.");
}

void markFree() {
 System.out.println("Which instructor?");
 System.out.println("1. Jeff");
 System.out.println("2. Anna");
 int instructor = sc.nextInt();

 System.out.println("Which day?");
 int day = sc.nextInt() - 1;

 System.out.println("Which hour?");
 int hour = sc.nextInt() - 1;
```

```
 if (instructor == 1)
  jeffSchedule[day][hour] = 0;
 else if (instructor == 2)
  annaSchedule[day][hour] = 0;
 else
  System.out.println("Invalid choice.");
}

void printSchedules() {
 System.out.println("\nJeff's Schedule:");
 printSchedule(jeffSchedule);

 System.out.println("\nAnna's Schedule:");
 printSchedule(annaSchedule);
}

void printSchedule(int[][] schedule) {
 for (int i = 0; i < 4; i++) {
  for (int j = 0; j < 3; j++) {
   System.out.print(schedule[i][j] == 1 ? "X " : "- ");
  }
  System.out.println();
 }
}

void showIndividualAvailability() {
 System.out.println("Available slots for individual lessons (Jeff):");
 showAvailability(jeffSchedule);
 System.out.println("Available slots for individual lessons (Anna):");
 showAvailability(annaSchedule);
}

void showAvailability(int[][] schedule) {
 System.out.println("Day\tHour");
 for (int i = 0; i < 4; i++) {
  for (int j = 0; j < 3; j++) {
   if (schedule[i][j] == 0) {
    System.out.println((i + 1) + "\t" + (j + 1));
   }
  }
```

	}
	}
	
	void showGroupAvailability() {
	System.out.println("Available slots for group lessons:");
	System.out.println("Day\tHour");
	for (int i = 0; i < 4; i++) {
	for (int j = 0; j < 3; j++) {
	if (jeffSchedule[i][j] == 0 && annaSchedule[i][j] == 0) {
	System.out.println((i + 1) + "\t" + (j + 1));
	}
	}
	}
	}
	}

RESULT:	

```
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ javac SwimSchool.java
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$ java SwimSchool

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 1
Which instructor?
1. Jeff
2. Anna
1
Which day?
1
Which hour?
1

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 2
Which instructor?
1. Jeff
2. Anna
2
Which day?
2
Which hour?
2

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 3
```

```
Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 3

Jeff's Schedule:
X - -
- - -
- - -
- - -

Anna's Schedule:
- - -
- - -
- - -
- - -

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 4
Available slots for individual lessons (Jeff):
Day     Hour
1       2
1       3
2       1
2       2
2       3
3       1
3       2
3       3
4       1
4       2
4       3
Available slots for individual lessons (Anna):
Day     Hour
1       1
1       2
```

```
1       1
1       2
1       3
2       1
2       2
2       3
3       1
3       2
3       3
4       1
4       2
4       3

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 5
Available slots for group lessons:
Day     Hour
1       2
1       3
2       1
2       2
2       3
3       1
3       2
3       3
4       1
4       2
4       3

Menu:
1. Schedule a lesson
2. Cancel a lesson
3. View schedules
4. Show available slots for individual lessons
5. Show available slots for group lessons
6. Exit
Enter your choice: 6
Exiting...
lenovo@lenovo-ThinkCentre-neo-50s-Gen-3:~/Desktop/2023300065$
```

CONCLUSION:	I have learnt about the uses and types of Arrays. I have learnt both about 1D array and 2D array and the use of private class for protecting sensitive information.