

Week - 4

Laya Gollamandala

1,Write a program to manage a list of students using ArrayList. Include methods to add, remove, and display students.

Program:

```
import java.util.ArrayList;

public class StudentManager {
    private ArrayList<String> students;

    public StudentManager() {
        students = new ArrayList<>();
    }

    public void addStudent(String name) {
        students.add(name);
        System.out.println(name + " has been added.");
    }

    public void removeStudent(String name) {
        if (students.remove(name)) {
            System.out.println(name + " has been removed.");
        } else {
            System.out.println(name + " was not found.");
        }
    }

    public void displayStudents() {
        System.out.println("Student List:");
        for (String student : students) {
            System.out.println(student);
        }
    }

    public static void main(String[] args) {
        StudentManager manager = new StudentManager();
        manager.addStudent("Laya");
        manager.addStudent("Harika");
        manager.displayStudents();
        manager.removeStudent("Laya");
    }
}
```

```

        manager.displayStudents();
    }
}

```

Output:

Laya has been added.
 Harika has been added.
 Student List:
 Laya
 Harika
 Laya has been removed.
 Student List:
 Harika

2, Create a program that uses a HashMap to store and retrieve employee details based on their employee ID.

Program:

```

import java.util.HashMap;
import java.util.Map;
public class EmployeeManager {
    private HashMap<String, String> employeeMap;
    public EmployeeManager() {
        employeeMap = new HashMap<>();
    }
    public void addEmployee(String id, String name) {
        employeeMap.put(id, name);
        System.out.println("Employee added/updated: ID = " + id + ", Name = " + name);
    }
    public void removeEmployee(String id) {
        if (employeeMap.remove(id) != null) {
            System.out.println("Employee with ID " + id + " has been removed.");
        } else {
            System.out.println("Employee with ID " + id + " not found.");
        }
    }
    public void getEmployee(String id) {
        String name = employeeMap.get(id);
        if (name != null) {
            System.out.println("Employee ID: " + id + ", Name: " + name);
        } else {
            System.out.println("Employee with ID " + id + " not found.");
        }
    }
    public void displayAllEmployees() {
        System.out.println("Employee List:");
    }
}

```

```

        for (Map.Entry<String, String> entry : employeeMap.entrySet()) {
            System.out.println("ID: " + entry.getKey() + ", Name: " + entry.getValue());
        }
    }

    public static void main(String[] args) {
        EmployeeManager manager = new EmployeeManager();
        manager.addEmployee("1", "Harika");
        manager.addEmployee("2", "Laya");
        manager.getEmployee("1");
        manager.displayAllEmployees();
        manager.removeEmployee("2");
        manager.displayAllEmployees();
    }
}

```

Output:

Employee added/updated: ID = 1, Name = Harika
 Employee added/updated: ID = 2, Name = Laya
 Employee ID: 1, Name: Harika
 Employee List:
 ID: 1, Name: Harika
 ID: 2, Name: Laya
 Employee with ID 2 has been removed.
 Employee List:
 ID: 1, Name: Harika

3.Implement a program that handles multiple exceptions (e.g., `ArithmeticException`, `NullPointerException`) and uses custom exceptions

Program:

```

class InvalidAgeException extends Exception {
    public InvalidAgeException(String message) {
        super(message);
    }
}

public class ExceptionHandlingDemo {
    public static void processData(int number, String name, int age) throws InvalidAgeException {
        try {
            int result = 10 / number;
            System.out.println("Division result: " + result);
        } catch (ArithmeticException e) {
            System.out.println("Error: Division by zero is not allowed.");
        }
        try {

```

```

        System.out.println("Length of name: " + name.length());
    } catch (NullPointerException e) {
        System.out.println("Error: The name is null.");
    }
    if (age < 0) {
        throw new InvalidAgeException("Age cannot be negative.");
    }
    System.out.println("Age is valid: " + age);
}

public static void main(String[] args) {
    try {
        processData(2, "Laya", 24);
        processData(0, null, -5);
    } catch (InvalidAgeException e) {
        System.out.println("Custom Exception: " + e.getMessage());
    }
}
}

```

Output:

Division result: 5

Length of name: 4

ERROR!

Age is valid: 24

Error: Division by zero is not allowed.

ERROR!

Error: The name is null.

Custom Exception: Age cannot be negative.