CAP:680 JAVA PROGRAMMING LANGUAGE LABORATORY

ASSIGNMENT-5

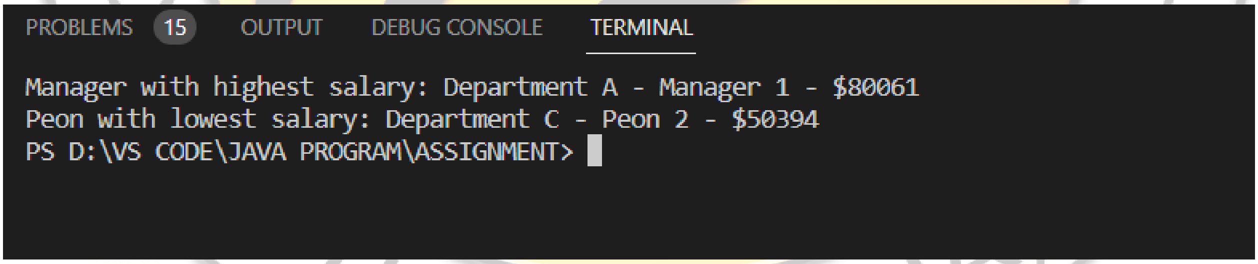
ST_NAME: - **EKHLAKH AHMAD**
REG NO.: - **12209166**
ROLL NO.: - **RD2215B50**
SECTION: - **D2215**
GROUP: - **2**

Q.1. a Company has four different departments containing 10 employee each of the department these employees are sharing some designation as well like manager, supervisor, store keeper, clerk, peon etc., Now being a boss of the company i would like to see who is the manager getting highest salary and who is the peon getting lowest salary.

Ans: -

```
import java.util.ArrayList;

import java.util.List;

import java.util.Comparator;


public class Main {


    public static void main(String[] args) {


        // Create employees

        List<Employee> employees = new ArrayList<>();


        // Define the departments and designations

        String[] departments = {"Department A", "Department C", "Department B", "Department D"};

        String[] designations = {"Manager", "Supervisor", "Store Keeper", "Clerk", "Peon"};


        // Define the number of employees for each designation
        int[] numEmployees = {1, 2, 3, 2, 2};

        // Create employees for each department and designation
        for (int i = 0; i < departments.length; i++) {
          for (int j = 0; j < designations.length; j++) {
            for (int k = 0; k < numEmployees[j]; k++) {
              String name = departments[i] + " - " + designations[j] + " " + (k + 1);
              int salary = (int) (Math.random() * 50000 + 50000); // random salary between 50,000 and 100,000
              employees.add(new Employee(name, departments[i], designations[j], salary));
            }
          }
        }

        // Find the manager with the highest salary
        Employee managerWithHighestSalary = employees.stream()
            .filter(e -> e.getDesignation().equals("Manager"))
            .max(Comparator.comparing(Employee::getSalary))
            .orElse(null);
        if (managerWithHighestSalary != null) {
          System.out.println("Manager with highest salary: " + managerWithHighestSalary.getName() + " - $" + managerWithHighestSalary.getSalary());
        }

        // Find the peon with the lowest salary
        Employee peonWithLowestSalary = employees.stream()
            .filter(e -> e.getDesignation().equals("Peon"))
```

```
                .min(Comparator.comparing(Employee::getSalary))
                .orElse(null);
        if (peonWithLowestSalary != null) {
            System.out.println("Peon with lowest salary: " + peonWithLowestSalary.getName() + " - $" + peonWithLowestSalary.getSalary());
        }
    }

    private static class Employee {
        private String name;
        private String department;
        private String designation;
        private int salary;

        public Employee(String name, String department, String designation, int salary) {
            this.name = name;
            this.department = department;
            this.designation = designation;
            this.salary = salary;
        }

        public String getName() {
            return name;
        }

        public String getDepartment() {
            return department;
        }
        public String getDesignation() {
            return designation;
        }

        public int getSalary() {
            return salary;
        }

        public String toString() {
            return String.format("%s (%s, %s) - $%d", name, department, designation, salary);
        }
    }
}
```



```
Manager with highest salary: Department A - Manager 1 - $80061
Peon with lowest salary: Department C - Peon 2 - $50394
PS D:\VS CODE\JAVA PROGRAM\ASSIGNMENT> 
```