Automatic evaluation[+]
ProjectAllocationRefactoring/src/project/Employee.java

```
1 package project;
2 public class Employee
3 {
4 private String employeeId;
5 private String employeeName;
6 private String emailId;
7 private String designation;
8 public Employee(String employeeId, String employeeName,
9 String emailId, String designation) {
10 this.employeeId = employeeId;
11 this.employeeName = employeeName;
12 this.emailId = emailId;
13 this.designation = designation;
14 }
15 public String getEmployeeId() {
16 return employeeId;
17 }
18 public void setEmployeeId(String employeeId) {
19 this.employeeId = employeeId;
20 }
21 public String getEmployeeName() {
22 return employeeName;
23 }
24 public void setEmployeeName(String employeeName) {
25 this.employeeName = employeeName;
26 }
27 public String getEmailId() {
28 return emailId;
29 }
30 public void setEmailId(String emailId) {
31 this.emailId = emailId;
32 }
33 public String getDesignation() {
34 return designation;
35 }
36 public void setDesignation(String designation) {
37 this.designation = designation;
38 }
39 @Override
40 public String toString() {
41 return "Employee [employeeId=" + employeeId + ",employeeName=" +
employeeName + ", emailId=" + emailId+ ", designation=" + designation + "]";
42 }
43 }
```

ProjectAllocationRefactoring/src/project/EmployeeDAO.java

```
1 package project;
2 import java.util.ArrayList;
3 import java.util.List;
4 public class EmployeeDAO {
5 private final List<Employee> employeeList = new
6 ArrayList<>();
7 public void addEmployee(Employee employee) {
8 employeeList.add(employee);
9 }
10 public void removeEmployee(Employee employee) {
```

```
11 employeeList.remove(employee);
12 }
13 public void viewEmployee() {
14 for (Employee employee : employeeList) {
15 System.out.println("Employee Id:" +
16 employee.getEmployeeId());
17 System.out.println("Employee Name:" +
18 employee.getEmployeeName());
19 System.out.println("Email Id:" +
20 employee.getEmailId());
21 System.out.println("Designation: " +
22 employee.getDesignation());
23 }
24 }
25 }
```

ProjectAllocationRefactoring/src/project/Project.java

```
 1 package project;
 2
 3 public class Project{
 4     String projectId;
 5     String projectName = new String("");
 6     String projectManagerName;
 7     int duration;
 8     String startDate;
 9     String endDate;
10
11     public Project(){
12
13     }
14     public Project(String projectId, String projectName, String projectManagerName, int
duration, String startDate,String endDate) {
15             super();
16             this.projectId = projectId;
17             this.projectName = projectName;
18             this.projectManagerName = projectManagerName;
19             this.duration = duration;
20             this.startDate = startDate;
21             this.endDate = endDate;
22     }
23 public String getProjectId() {
24 return projectId;
25 }
26 public void setProjectId(String projectId) {
27 this.projectId = projectId;
28 }
29 public String getProjectName() {
30 return projectName;
31 }
32 public void setProjectName(String projectName) {
33 this.projectName = projectName;
34 }
35 public String getProjectManagerName() {
36 return projectManagerName;
37 }
38 public void setProjectManagerName(String projectManagerName)
39 {
40 this.projectManagerName = projectManagerName;
```

```
41 }
42 public int getDuration() {
43 return duration;
44 }
45 public void setDuration(int duration) {
46 this.duration = duration;
47 }
48 public String getStartDate() {
49 return startDate;
50 }
51 public void setStartDate(String startDate) {
52 this.startDate = startDate;
53 }
54 public String getEndDate() {
55 return endDate;
56 }
57 public void setEndDate(String endDate) {
58 this.endDate = endDate;
59 }
60 @Override
61 public String toString() {
62 return "Project [projectId=" + projectId + ",projectName=" + projectName + ", projectManagerName="+ projectManagerName + ", duration=" +duration + ", startDate=" + startDate + ", endDate=" + endDate+ "]";
63 }
64 }
65
66
67
68
69
ProjectAllocationRefactoring/src/project/ProjectAllocation.java
 1 package project;
 2 import java.util.Date;
 3 public class ProjectAllocation {
 4 private Employee employee;
 5 private Project project;
 6 private int projectAllocationId;
 7 private String moduleName;
 8 private static final int NO_OF_PROJECTS_WORKING_IN_PARALLEL
 9 = 0;
10 private Date allocationDate;
11 private static final int NO_OF_HOURS_ALLOCATED = 160;
12 public ProjectAllocation(Employee employee, Project project,
13 int projectAllocationId, String moduleName,
14 Date allocationDate) {
15 this.employee = employee;
16 this.project = project;
17 this.projectAllocationId = projectAllocationId;
18 this.moduleName = moduleName;
19 this.allocationDate = allocationDate;
20 }
21 public Employee getEmployee() {
22 return employee;
23 }
24 public void setEmployee(Employee employee) {
25 this.employee = employee;
```

```
26 }
27 public Project getProject() {
28 return project;
29 }
30 public void setProject(Project project) {
31 this.project = project;
32 }
33 public int getProjectAllocationId() {
34 return projectAllocationId;
35 }
36 public void setProjectAllocationId(int projectAllocationId)
37 {
38 this.projectAllocationId = projectAllocationId;
39 }
40 public String getModuleName() {
41 return moduleName;
42 }
43 public void setModuleName(String moduleName) {
44 this.moduleName = moduleName;
45 }
46 public Date getAllocationDate() {
47 return allocationDate;
48 }
49 public void setAllocationDate(Date allocationDate) {
50 this.allocationDate = allocationDate;
51 }
52 public static int getNoOfProjectsWorkingInParallel() {
53 return NO_OF_PROJECTS_WORKING_IN_PARALLEL;
54 }
55 public static int getNoOfHoursAllocated() {
56 return NO_OF_HOURS_ALLOCATED;
57 }
58 @Override
59 public String toString() {
60 return "ProjectAllocation [employee=" + employee + ",project=" + project + ",
projectAllocationId="+ projectAllocationId + ", moduleName=" +moduleName + ",
allocationDate=" + allocationDate + "]";
61 }
62 }
```

ProjectAllocationRefactoring/src/project/ProjectAllocationDAO.java

```
1 package project;
2 import java.util.ArrayList;
3 import java.util.List;
4 public class ProjectAllocationDAO {
5 private final List<ProjectAllocation> projectAllocationList
6 = new ArrayList<>();
7 public void addProjectAllocation(ProjectAllocation
8 projectAllocation) {
9 projectAllocationList.add(projectAllocation);
10 }
11 public void removeProjectAllocation(ProjectAllocation
12 projectAllocation) {
13 projectAllocationList.remove(projectAllocation);
14 }
15 public void viewProjectAllocation() {
16 if (projectAllocationList.isEmpty())
17 {
```

```
18 System.out.println("Project Allocation List is empty");
19 }
20 else {
21 for (ProjectAllocation projectAllocation :
22 projectAllocationList) {
23 System.out.println("Project Allocation Id:"
24 + projectAllocation.getProjectAllocationId());
25 System.out.println("Project Id:" +
26 projectAllocation.getProject().getProjectId());
27 System.out.println("Employee Id:" +
28 projectAllocation.getEmployee().getEmployeeId());
29 System.out.println("Allocation Date:" +
30 projectAllocation.getAllocationDate());
31 System.out.println("Module Name:" +
32 projectAllocation.getModuleName());
33 }
34 }
35 }
36 }
```

ProjectAllocationRefactoring/src/project/ProjectDAO.java

```
1 package project;
2 import java.util.ArrayList;
3 import java.util.List;
4 public class ProjectDAO {
5 private final List<Project> projectList = new ArrayList<>();
6 public void addProject(Project project) {
7 projectList.add(project);
8 }
9 public void removeProject(Project project) {
10 projectList.remove(project);
11 }
12 public void viewProject() {
13 for (Project project : projectList) {
14 System.out.println("Project Id:" +
15 project.getProjectId());
16 System.out.println("Project Name:" +
17 project.getProjectName());
18 System.out.println("Project Manager Name:" +
19 project.getProjectManagerName());
20 System.out.println("Duration:" +
21 project.getDuration());
22 System.out.println("Start Date:" +
23 project.getStartDate());
24 System.out.println("End Date:" +
25 project.getEndDate());
26 }
27 }
28 }
```

Grade

Reviewed on Thursday, 11 March 2021, 10:30 AM by Automatic grade

Grade 92.73 / 100

Assessment report

[+]SOURCE CODE ANALYZER REPORT

[+]Grading and Feedback