# DEPARTMENT OF
# COMPUTER SCIENCE & ENGINEERING

**Assignment-1**

**Student Name:** Aadity Anand
**UID:** 23BCS10446
**Branch:** CSE
**Section/Group:** —
**Semester:** 6th
**Subject Code:** 23CSH-314
**Subject Name:** System Design

---

## Q1: Explain SRP and OCP in detail with proper examples.

**Solution:**

Single Responsibility Principle (SRP) and Open-Closed Principle (OCP) are important software design principles that help in developing clean and maintainable systems.

### 1. Single Responsibility Principle (SRP)

**What is SRP?**

SRP states that a class should have only one responsibility and one reason to change. A class should perform only one main task.

**Role of SRP:**

- Improves code readability
- Makes maintenance easy
- Reduces errors
- Helps in testing
- Improves system design

**Example Code (SRP Violation):**

```
class StudentManager {

 void saveStudent(){}

 void generateReport(){}

 void sendEmail(){}
}
```

This class performs many tasks, so it violates SRP.

**Correct SRP Implementation:**

```
class Student {}

class StudentRepository {
 void save(Student s){}
}

class ReportService {
 void generate(Student s){}
}

class EmailService {
 void send(Student s){}
}
```

Each class has only one responsibility.

## 2. Open-Closed Principle (OCP)

### What is OCP?

OCP states that software entities should be open for extension but closed for modification.

**Role of OCP:**

- Protects existing code
- Improves stability
- Supports scalability
- Reduces testing cost
- Avoids bugs

**Example Code (OCP Violation):**

```
class Payment {

 double pay(String type,double amt){

  if(type=="Card")
   return amt*0.9;

  else if(type=="UPI")
   return amt*0.95;

  else
   return amt;
 }
}
```

**Correct OCP Implementation:**

```
interface Payment {
 double pay(double amt);
}

class CardPayment implements Payment{
 public double pay(double amt){
  return amt*0.9;
 }
}

class UPIPayment implements Payment{
 public double pay(double amt){
  return amt*0.95;
 }
}
```

**HLD for Q1:**

Layered Architecture:

User → Controller → Service → Database

## Q2: Discuss violations in SRP and OCP along with their fixes.

**Solution:**

Violations occur when design principles are not followed properly. This results in complex and unreliable systems.

### 1. SRP Violations

SRP violations occur when one class handles many responsibilities.

**Examples:**

- Database + UI in one class
- Business logic + Email
- Multiple services together

**Problems:**

- Difficult maintenance
- More bugs
- Poor readability
- Low reusability

**Fix for SRP Violations:**

- Divide big classes
- Create separate modules
- Use layered architecture

### 2. OCP Violations

OCP violations occur due to frequent modification of code.

**Examples:**

- Too many if-else
- Switch statements
- Hardcoding values

**Problems:**

- Code breaks easily
- High testing cost
- Risk of failure

**Fix for OCP Violations:**

- Use interfaces
- Use abstract classes
- Apply polymorphism
- Use design patterns

**Example Code:**

```
abstract class Notification {
 abstract void send();
}

class Email extends Notification {
 void send(){}
}

class SMS extends Notification {
 void send(){}
}
```

**HLD for Q2:**
Wrong Design → One Big Class Correct Design → Separate Modules

## Q3: Design an HLD for an Online Examination System applying these principles.

**Solution:**

An Online Examination System allows students to give exams through the internet and provides automatic evaluation.

### 1. Objectives

- Conduct online exams
- Reduce manual work
- Improve accuracy
- Provide fast results

### 2. Main Modules

1. User Management
2. Authentication
3. Exam Module
4. Question Bank
5. Evaluation
6. Result
7. Notification

### 3. Module Responsibilities (SRP)

- User $\rightarrow$ Manage users
- Auth $\rightarrow$ Login security
- Exam $\rightarrow$ Handle exams
- Evaluation $\rightarrow$ Check answers
- Result $\rightarrow$ Generate reports

## 4. Extensibility (OCP)

New features can be added without modifying old code.

- New exam type
- New login system
- New notification service

## 5. System Workflow

1. Registration
2. Login
3. Select Exam
4. Attempt Test
5. Submit
6. Evaluation
7. Result

## 6. Example Code

```
interface ExamType {
 void conductExam();
}

class MCQExam implements ExamType{
 public void conductExam(){}
}

class CodingExam implements ExamType{
 public void conductExam(){}
}
```

## 7. HLD for Online Examination System

Student/Admin $\rightarrow$ Web App $\rightarrow$ Server $\rightarrow$ Modules $\rightarrow$ Database

## Conclusion

Single Responsibility Principle and Open-Closed Principle are essential in software development.

SRP improves organization and clarity. OCP improves flexibility and scalability.

By applying these principles, the Online Examination System becomes reliable and maintainable.

Hence, SRP and OCP are very important for modern software systems.