ASSIGNMENT – 1

Student Information System (SIS)

SHRUTHI C

Task 1: Define Classes

Student.java

```
package com.hexaware.sis.model;

import java.util.*;

public class Student {     private
int studentId;     private String
firstName;     private String
lastName;     private Date
dateOfBirth;     private String
email;     private String
phoneNumber;

   private List<Enrollment> enrollments = new ArrayList<>();
private List<Payment> payments = new ArrayList<>();

   public Student(int studentId, String firstName, String lastName, Date dateOfBirth, String email,
String phoneNumber) {
       this.studentId     =     studentId;
this.firstName       =       firstName;
this.lastName        =        lastName;
this.dateOfBirth = dateOfBirth;
       this.email = email;
       this.phoneNumber = phoneNumber;
```

```
    }

}
```

Course.java

```
package com.hexaware.sis.model;

import java.util.*;

public class Course {     private
int courseId;     private String
courseName;     private String
courseCode;     private String
instructorName;     private
Teacher teacher;

    private List<Enrollment> enrollments = new ArrayList<>();

    public Course(int courseId, String courseName, String courseCode, String instructorName) {
        this.courseId = courseId;
this.courseName = courseName;
this.courseCode = courseCode;
this.instructorName = instructorName;
    }
}
```

Enrollment.java

```
package com.hexaware.sis.model;
```

```
import java.util.*;

public class Enrollment {
private int enrollmentId;
private Student student;
private Course course;
private Date enrollmentDate;

   public Enrollment(int enrollmentId, Student student, Course course, Date enrollmentDate) {
this.enrollmentId = enrollmentId;         this.student = student;         this.course = course;
this.enrollmentDate = enrollmentDate;
   }

}
```

Teacher.java

```
package com.hexaware.sis.model;
import java.util.*; public
class Teacher {     private
int teacherId;     private
String firstName;     private
String lastName;     private
String email;

   private List<Course> assignedCourses = new ArrayList<>();

   public Teacher(int teacherId, String firstName, String lastName, String email) { this.teacherId
      = teacherId;
```

```
        this.firstName = firstName;
this.lastName = lastName;         this.email
= email;
    } }
```

Payment.java

```
package com.hexaware.sis.model;

import java.util.*;

public class Payment {
private int paymentId;
private Student student;
private double amount;
private Date paymentDate;

    public Payment(int paymentId, Student student, double amount, Date paymentDate) {
this.paymentId = paymentId;          this.student = student;          this.amount = amount;
this.paymentDate = paymentDate;
    }
}
```

Task 2: Implement Constructors

Already covered in each class above.

Task 3: Implement Methods

Student.java

```
package com.hexaware.sis.model;

import java.time.LocalDate;
import java.util.ArrayList; import
java.util.List;

public class Student {     private
int studentId;     private String
firstName;     private String
lastName;     private LocalDate
dateOfBirth;     private String
email;
    private String phoneNumber;     private List<Enrollment>
enrollments = new ArrayList<>();     private List<Payment>
payments = new ArrayList<>();

    public Student(int studentId, String firstName, String lastName, LocalDate dateOfBirth, String email,
String phoneNumber) {
        this.studentId     =     studentId;
this.firstName        =        firstName;
this.lastName        =        lastName;
this.dateOfBirth = dateOfBirth;
        this.email = email;
        this.phoneNumber = phoneNumber;
    }
    public void enrollInCourse(Course course, int enrollmentId, LocalDate enrollmentDate) {
Enrollment enrollment = new Enrollment(enrollmentId, this, course, enrollmentDate);
enrollments.add(enrollment);        course.getEnrollments().add(enrollment);
    }

    public void updateStudentInfo(String firstName, String lastName, LocalDate dateOfBirth, String
email, String phoneNumber) {        this.firstName = firstName;        this.lastName = lastName;
this.dateOfBirth = dateOfBirth;        this.email = email;        this.phoneNumber = phoneNumber;
    }

    public void makePayment(int paymentId, double amount, LocalDate paymentDate) {
Payment payment = new Payment(paymentId, this, amount, paymentDate);
payments.add(payment);
    }

    public void displayStudentInfo() {
        System.out.println("Student ID: " + studentId);
        System.out.println("Name: " + firstName + " " + lastName);
        System.out.println("DOB: " + dateOfBirth);
        System.out.println("Email: " + email);
        System.out.println("Phone: " + phoneNumber);
    }

    public List<Course> getEnrolledCourses() {
        List<Course> courses = new ArrayList<>();
for (Enrollment enrollment : enrollments) {
courses.add(enrollment.getCourse());
        }
        return courses;
    }
    public List<Payment> getPaymentHistory() {
return payments;
    }

    public int getStudentId() {
return studentId;
    }

    public List<Enrollment> getEnrollments() {
return enrollments;
    }
}
```

Course.java

```
package com.hexaware.sis.model;

import java.util.ArrayList; import
java.util.List;

public class Course {
private int courseId;
private String courseName;
private String courseCode;
private Teacher instructor;
private List<Enrollment>
enrollments = new
ArrayList<>();
```

```
  public Course(int courseId, String courseName, String courseCode, Teacher instructor) {
this.courseId = courseId;          this.courseName = courseName;          this.courseCode =
courseCode;        this.instructor = instructor;
  }

  public void assignTeacher(Teacher teacher) {
this.instructor                =                teacher;
teacher.getAssignedCourses().add(this);
  }

  public void updateCourseInfo(String courseCode, String courseName, String instructorName) {
this.courseCode = courseCode;        this.courseName = courseName;
  }

  public void displayCourseInfo() {
    System.out.println("Course ID: " + courseId);
    System.out.println("Name: " + courseName);
System.out.println("Code: " + courseCode);
    if (instructor != null) {
      System.out.println("Instructor: " + instructor.getFirstName() + " " + instructor.getLastName());
    }
  }

  public List<Enrollment> getEnrollments() {
return enrollments;

  }

  public Teacher getTeacher() {
return instructor;
  }
```

```
   public String getCourseName() {
return courseName;
   }

   public String getCourseCode() {
return courseCode;
   }
}
```

Enrollment.java

```
package com.hexaware.sis.model;

import java.time.LocalDate;

public class Enrollment {     private
int enrollmentId;     private Student
student;     private Course course;
private LocalDate enrollmentDate;

   public Enrollment(int enrollmentId, Student student, Course course, LocalDate enrollmentDate) {
this.enrollmentId = enrollmentId;         this.student = student;         this.course = course;
      this.enrollmentDate = enrollmentDate;
   }

   public Student getStudent() {
return student;
   }

   public Course getCourse() {
return course;
```

```
    }

    public int getEnrollmentId() {
return enrollmentId;
    }

    public LocalDate getEnrollmentDate() {
return enrollmentDate;
    }
}
```

Teacher.java

```
package com.hexaware.sis.model;

import java.util.ArrayList; import
java.util.List;

public class Teacher {
private int teacherId;
private String firstName;
private String lastName;
private String email;
private String expertise;
private List<Course>
assignedCourses = new
ArrayList<>();

    public Teacher(int teacherId, String firstName, String lastName, String email, String expertise) {
this.teacherId = teacherId;         this.firstName = firstName;         this.lastName = lastName;
this.email = email;         this.expertise = expertise;
```

```
    }

    public void updateTeacherInfo(String name, String email, String expertise) {
String[] names = name.split(" ");        this.firstName = names[0];
this.lastName = names.length > 1 ? names[1] : "";
        this.email = email;
this.expertise = expertise;
    }

    public void displayTeacherInfo() {
        System.out.println("Teacher ID: " + teacherId);
        System.out.println("Name: " + firstName + " " + lastName);
        System.out.println("Email: " + email);
        System.out.println("Expertise: " + expertise);
    }

    public List<Course> getAssignedCourses() {
return assignedCourses;
    }

    public String getFirstName() {
return firstName;
    }

    public String getLastName() {
return lastName;
    }
}
```

Payment.java

```
package com.hexaware.sis.model;

import java.time.LocalDate;

public class Payment {     private
int paymentId;     private Student
student;     private double
amount;     private LocalDate
paymentDate;

    public Payment(int paymentId, Student student, double amount, LocalDate paymentDate) {
this.paymentId = paymentId;             this.student = student;             this.amount = amount;
this.paymentDate = paymentDate;
    }

    public Student getStudent() {
return student;
    }

    public double getPaymentAmount() {
        return amount;
    }

    public LocalDate getPaymentDate() {
return paymentDate;
    }
}
```

Task 4: Exceptions handling and Custom Exceptions

DuplicateEnrollmentException.java

```
package com.hexaware.sis.exception;

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

CourseNotFoundException.java

```
package com.hexaware.sis.exception;

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

StudentNotFoundException.java

```
package com.hexaware.sis.exception;

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

}

TeacherNotFoundException.java

```
package com.hexaware.sis.exception;

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

PaymentValidationException.java

```
package com.hexaware.sis.exception;

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

InvalidStudentDataException.java

```
package com.hexaware.sis.exception;

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

InvalidCourseDataException.java

```
package com.hexaware.sis.exception;

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

InvalidEnrollmentDataException.java

```
package com.hexaware.sis.exception;

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

InvalidTeacherDataException.java

```
package com.hexaware.sis.exception;

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

InsufficientFundsException.java

```
package com.hexaware.sis.exception;

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

Task 5: Collections

Student.java

```
package com.hexaware.sis.model;

import java.time.LocalDate;
import java.util.ArrayList; import
java.util.List;

public     class     Student     {
private int studentId; private
String  firstName;       private
String    lastName;    private
LocalDate          dateOfBirth;
private       String       email;
private                     String
phoneNumber;

   private List<Enrollment> enrollments;
private List<Payment> payments;
```

```
   public Student(int studentId, String firstName, String lastName, LocalDate dateOfBirth, String email,
String phoneNumber) {
      this.studentId = studentId;
this.firstName = firstName;
this.lastName = lastName;
this.dateOfBirth = dateOfBirth;
this.email = email;        this.phoneNumber
= phoneNumber;        this.enrollments =
new ArrayList<>();        this.payments =
new ArrayList<>();
   }

   public int getStudentId() { return studentId; }     public void
setStudentId(int studentId) { this.studentId = studentId; }

   public String getFirstName() { return firstName; }     public void
setFirstName(String firstName) { this.firstName = firstName; }

   public String getLastName() { return lastName; }     public void
setLastName(String lastName) { this.lastName = lastName; }

   public LocalDate getDateOfBirth() { return dateOfBirth; }
   public void setDateOfBirth(LocalDate dateOfBirth) { this.dateOfBirth = dateOfBirth; }

   public String getEmail() { return email; } public void
   setEmail(String email) { this.email = email; }

   public String getPhoneNumber() { return phoneNumber; }     public void
setPhoneNumber(String phoneNumber) { this.phoneNumber = phoneNumber; }
```

```
   public List<Enrollment> getEnrollments() { return enrollments; }     public void
setEnrollments(List<Enrollment> enrollments) { this.enrollments = enrollments; }

   public List<Payment> getPayments() { return payments; }     public void
setPayments(List<Payment> payments) { this.payments = payments; }

   public void addEnrollment(Enrollment enrollment) {
enrollments.add(enrollment);
   }

   public void addPayment(Payment payment) {
payments.add(payment);
   }
}
```

Course.java

```
package com.hexaware.sis.model;

import java.util.ArrayList; import
java.util.List;

public class Course {
   private int courseId;     private
String courseName; private
String courseCode; private String
instructorName;

   private List<Enrollment> enrollments;
   public Course(int courseId, String courseName, String courseCode, String instructorName) {
this.courseId = courseId;
       this.courseName = courseName;
this.courseCode = courseCode;
this.instructorName = instructorName;
this.enrollments = new ArrayList<>();
   }

   public int getCourseId() { return courseId; }     public void
setCourseId(int courseId) { this.courseId = courseId; }

   public String getCourseName() { return courseName; }     public void
setCourseName(String courseName) { this.courseName = courseName; }

   public String getCourseCode() { return courseCode; }     public void
setCourseCode(String courseCode) { this.courseCode = courseCode; }

   public String getInstructorName() { return instructorName; }     public void
setInstructorName(String instructorName) { this.instructorName = instructorName; }

   public List<Enrollment> getEnrollments() { return enrollments; }     public void
setEnrollments(List<Enrollment> enrollments) { this.enrollments = enrollments; }

   public void addEnrollment(Enrollment enrollment) {
       enrollments.add(enrollment);
   }
}
```

Enrollment.java

```
package com.hexaware.sis.model;
```

```
import java.time.LocalDate;

public class Enrollment {     private
int enrollmentId;     private Student
student;     private Course course;
private LocalDate enrollmentDate;

   public Enrollment(int enrollmentId, Student student, Course course, LocalDate enrollmentDate) {
this.enrollmentId = enrollmentId;         this.student = student;         this.course = course;
this.enrollmentDate = enrollmentDate;
   }

   public int getEnrollmentId() { return enrollmentId; }     public void
setEnrollmentId(int enrollmentId) { this.enrollmentId = enrollmentId; }

   public Student getStudent() { return student; }     public void
setStudent(Student student) { this.student = student; }

   public Course getCourse() { return course; }     public void
setCourse(Course course) { this.course = course; }     public
LocalDate getEnrollmentDate() { return enrollmentDate; }
   public void setEnrollmentDate(LocalDate enrollmentDate) { this.enrollmentDate = enrollmentDate;
}
}
```

Teacher.java

```
package com.hexaware.sis.model;

import java.util.ArrayList; import
java.util.List;
```

```
public class Teacher {
private int teacherId;
private String firstName;
private String lastName;
private String email;

   private List<Course> assignedCourses;

   public Teacher(int teacherId, String firstName, String lastName, String email) {
this.teacherId = teacherId;         this.firstName = firstName;         this.lastName =
lastName;         this.email = email;         this.assignedCourses = new ArrayList<>();
   }

   public int getTeacherId() { return teacherId; }     public void
setTeacherId(int teacherId) { this.teacherId = teacherId; }     public
String getFirstName() { return firstName; }     public void
setFirstName(String firstName) { this.firstName = firstName; }

   public String getLastName() { return lastName; }     public void
setLastName(String lastName) { this.lastName = lastName; }

   public String getEmail() { return email; }     public void
setEmail(String email) { this.email = email; }

   public List<Course> getAssignedCourses() { return assignedCourses; }
   public void setAssignedCourses(List<Course> assignedCourses) { this.assignedCourses =
assignedCourses; }

   public void assignCourse(Course course) {
assignedCourses.add(course);
```

```
   }
}
```

Payment.java

```
package com.hexaware.sis.model;

import java.time.LocalDate;

public class Payment {     private
int paymentId;     private Student
student;     private double
amount;     private LocalDate
paymentDate;

   public Payment(int paymentId, Student student, double amount, LocalDate paymentDate) {
this.paymentId = paymentId;          this.student = student;          this.amount = amount;
this.paymentDate = paymentDate;
   }

   public int getPaymentId() { return paymentId; }     public void
setPaymentId(int paymentId) { this.paymentId = paymentId; }

   public Student getStudent() { return student; }     public void
setStudent(Student student) { this.student = student; }

   public double getAmount() { return amount; }     public void
setAmount(double amount) { this.amount = amount; }

   public LocalDate getPaymentDate() { return paymentDate; }     public void
setPaymentDate(LocalDate paymentDate) { this.paymentDate = paymentDate; }
}
```

Task 6: Create Methods for Managing Relationships

SISService.java

```
package com.hexaware.sis.service;


import com.hexaware.sis.model.*; import
com.hexaware.sis.exception.*;


import java.time.LocalDate; import
java.util.*;

public class SISService {


    private List<Student> students;
private List<Course> courses;     private
List<Teacher> teachers;     private
List<Enrollment> enrollments;     private
List<Payment> payments;


    public SISService() {        students
= new ArrayList<>();        courses =
new ArrayList<>();        teachers =
new ArrayList<>();        enrollments =
new ArrayList<>();        payments =
new ArrayList<>();
    }
    public void addStudent(Student student) {
students.add(student);
    }

    public void addCourse(Course course) {
courses.add(course);
    }

    public void addTeacher(Teacher teacher) {
teachers.add(teacher);
    }

    public void addEnrollment(Student student, Course course, LocalDate enrollmentDate) throws
DuplicateEnrollmentException {        for (Enrollment e : enrollments) {
            if (e.getStudent().getStudentId() == student.getStudentId() &&
                e.getCourse().getCourseId() == course.getCourseId()) {                throw new
DuplicateEnrollmentException("Student already enrolled in this course.");
            }
        }

        Enrollment enrollment = new Enrollment(enrollments.size() + 1, student, course,
enrollmentDate);        enrollments.add(enrollment);
student.getEnrollments().add(enrollment);
course.getEnrollments().add(enrollment);
    }

    public void assignCourseToTeacher(Course course, Teacher teacher) {
course.setInstructorName(teacher.getFirstName() + " " + teacher.getLastName());
teacher.getAssignedCourses().add(course);
    }
    public void addPayment(Student student, double amount, LocalDate paymentDate) throws
PaymentValidationException {
        if (amount <= 0) {
            throw new PaymentValidationException("Payment amount must be greater than 0.");
        }

        Payment payment = new Payment(payments.size() + 1, student, amount, paymentDate);
payments.add(payment);         student.getPayments().add(payment);
    }

    public List<Enrollment> getEnrollmentsForStudent(Student student) {
List<Enrollment> result = new ArrayList<>();         for (Enrollment e :
enrollments) {
            if (e.getStudent().getStudentId() == student.getStudentId()) {
                result.add(e);
            }
        }
        return result;
    }

    public List<Course> getCoursesForTeacher(Teacher teacher) {
return teacher.getAssignedCourses();
    }

    // Utility methods to fetch entities      public Student getStudentById(int id)
throws StudentNotFoundException {
        return students.stream()
                .filter(s -> s.getStudentId() == id)
                .findFirst()
                .orElseThrow(() -> new StudentNotFoundException("Student with ID " + id + " not found."));
    }
    public Course getCourseByCode(String code) throws CourseNotFoundException {
        return courses.stream()
                .filter(c -> c.getCourseCode().equals(code))
                .findFirst()
                .orElseThrow(() -> new CourseNotFoundException("Course with code " + code + " not
found."));
    }

    public Teacher getTeacherByEmail(String email) throws TeacherNotFoundException {
return teachers.stream()
                .filter(t -> t.getEmail().equals(email))
                .findFirst()
                .orElseThrow(() -> new TeacherNotFoundException("Teacher with email " + email + " not
found."));
    }

    public List<Student> getAllStudents() {
return students;
    }

    public List<Course> getAllCourses() {
return courses;
    }

    public List<Enrollment> getAllEnrollments() {
return enrollments;
    }

    public List<Teacher> getAllTeachers() {
return teachers;
    }
    public List<Payment> getAllPayments() {
return payments;
    }
}
```

Task 7: Database Connectivity

1.DBUtil.java

```
package com.hexaware.sis.util;

import java.sql.Connection; import
java.sql.DriverManager; import
java.sql.SQLException;

public class DBUtil {

    private static final String url =
"jdbc:mysql://localhost:3306/sis?useSSL=false&allowPublicKeyRetrieval=true&serverTimezone=UTC"
;
    private static final String USERNAME = "root";     private static final String
PASSWORD = "deva1234"; // Replace with your password

    public static Connection getConnection() throws SQLException {
return DriverManager.getConnection(URL, USERNAME, PASSWORD);
    }
}
```

2.DatabaseInitializer.java

```
package com.hexaware.sis.dao;
```

```
import com.hexaware.sis.util.DBUtil;

import java.sql.Connection; import
java.sql.Statement;

public class DatabaseInitializer {

   public static void initializeDatabase() {        try (Connection conn = DBUtil.getConnection();
Statement stmt = conn.createStatement()) {

          String createStudentTable = "CREATE TABLE IF NOT EXISTS student (" +
                "student_id INT PRIMARY KEY AUTO_INCREMENT," +
                "first_name VARCHAR(50)," +
                "last_name VARCHAR(50)," +
                "dob DATE," +
                "email VARCHAR(100)," +
                "phone VARCHAR(20))";

          String createCourseTable = "CREATE TABLE IF NOT EXISTS course (" +
                "course_id INT PRIMARY KEY AUTO_INCREMENT," +
                "course_name VARCHAR(100)," +
                "course_code VARCHAR(20)," +
                "instructor_name VARCHAR(100))";

          String createTeacherTable = "CREATE TABLE IF NOT EXISTS teacher (" +
                "teacher_id INT PRIMARY KEY AUTO_INCREMENT," +
                "first_name VARCHAR(50)," +
                "last_name VARCHAR(50)," +
                "email VARCHAR(100)," +
                "expertise VARCHAR(100))";
```

```
      String createEnrollmentTable = "CREATE TABLE IF NOT EXISTS enrollment (" +
          "enrollment_id INT PRIMARY KEY AUTO_INCREMENT," +
          "student_id INT," +
          "course_id INT," +
          "enrollment_date DATE," +
          "FOREIGN KEY(student_id) REFERENCES student(student_id)," +
          "FOREIGN KEY(course_id) REFERENCES course(course_id))";

      String createPaymentTable = "CREATE TABLE IF NOT EXISTS payment (" +
          "payment_id INT PRIMARY KEY AUTO_INCREMENT," +
          "student_id INT," +
          "amount DECIMAL(10,2)," +
          "payment_date DATE," +
          "FOREIGN KEY(student_id) REFERENCES student(student_id))";

      stmt.execute(createStudentTable);
stmt.execute(createCourseTable);       stmt.execute(createTeacherTable);
stmt.execute(createEnrollmentTable);
stmt.execute(createPaymentTable);

      System.out.println("Database initialized successfully!");

    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

3.QueryBuilder

```
package com.hexaware.sis.dao;


public class QueryBuilder {


    public static String buildSelectQuery(String tableName, String[] columns, String condition, String orderBy) {
        StringBuilder query = new StringBuilder("SELECT ");


        if (columns == null || columns.length == 0) {
query.append("*");
        } else {
            query.append(String.join(", ", columns));
        }


        query.append(" FROM ").append(tableName);


        if (condition != null && !condition.trim().isEmpty()) {
query.append(" WHERE ").append(condition);
        }


        if (orderBy != null && !orderBy.trim().isEmpty()) {
query.append(" ORDER BY ").append(orderBy);
        }


        return query.toString();
    }
}
```

4.SISMain.java package

com.hexaware.sis.main;

```
import com.hexaware.sis.dao.DatabaseInitializer;


public class SISMain {     public static
void main(String[] args) {
       DatabaseInitializer.initializeDatabase();
    }
}
```

Task 8: Student Enrollment

StudentDAO.java

```
package com.hexaware.sis.dao;


import java.sql.*; // and other imports
import com.hexaware.sis.model.*; import
com.hexaware.sis.util.DBUtil;


public class StudentDAO {
   // class contents


             public int addStudent(Student student) {
    String  sql  =  "INSERT  INTO  student  (first_name,  last_name,  date_of_birth,  email,
phone_number) VALUES (?, ?, ?, ?, ?)";      try (Connection conn = DBUtil.getConnection();
                     PreparedStatement ps = conn.prepareStatement(sql,
Statement.RETURN_GENERATED_KEYS)) {                  ps.setString(1,
student.getFirstName());                   ps.setString(2, student.getLastName());
ps.setDate(3, new java.sql.Date(student.getDateOfBirth().getTime()));
ps.setString(4, student.getEmail());                ps.setString(5,
student.getPhoneNumber());


                    ps.executeUpdate();
```

```
                ResultSet rs = ps.getGeneratedKeys();                   if
(rs.next()) {                   return rs.getInt(1); // Return generated
student_id
                }
            } catch (SQLException e) {
                e.printStackTrace();
            }
            return -1;
        }

}
```

CourseDAO.java

```
package com.hexaware.sis.dao;

import java.sql.*; // and other imports
import com.hexaware.sis.model.*; import
com.hexaware.sis.util.DBUtil;

public class CourseDAO {
    // class contents
            public Course getCourseByName(String name) {
                String sql = "SELECT * FROM course WHERE course_name =
?";            try (Connection conn = DBUtil.getConnection();
PreparedStatement ps = conn.prepareStatement(sql)) {
ps.setString(1, name);          ResultSet rs = ps.executeQuery();           if
(rs.next()) {
```

```
                        return new Course(rs.getInt("course_id"), rs.getString("course_name"),
rs.getString("course_code"));

                    }

                } catch (SQLException e) {

                    e.printStackTrace();

                }

                return null;

            }

}
```

EnrollmentDAO.java package

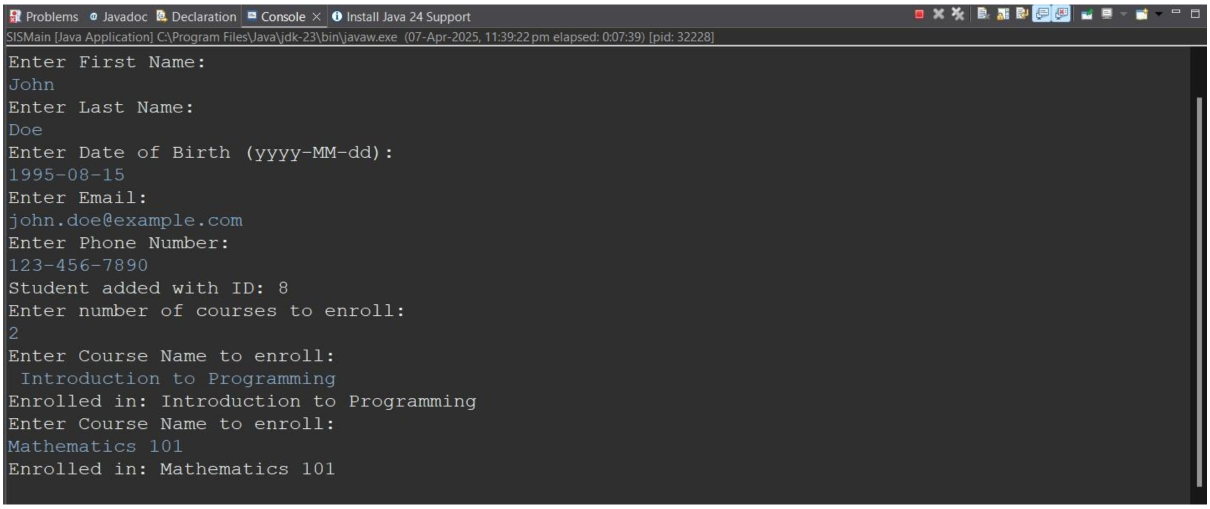
```
com.hexaware.sis.dao;


import java.sql.*; import

java.util.Date; import

com.hexaware.sis.util.DBUtil;


public class EnrollmentDAO {

   // class contents

             public void enrollStudent(int studentId, int courseId, Date date) {

                String sql = "INSERT INTO enrollment (student_id, course_id, enrollment_date) VALUES (?,
?, ?)";

                try (Connection conn = DBUtil.getConnection();

                     PreparedStatement ps = conn.prepareStatement(sql))

{                   ps.setInt(1, studentId);                ps.setInt(2, courseId);

       ps.setDate(3, new java.sql.Date(date.getTime()));

ps.executeUpdate();          } catch (SQLException e) {

                    e.printStackTrace();

                }
```

```
        }
}
```

SISMain.java

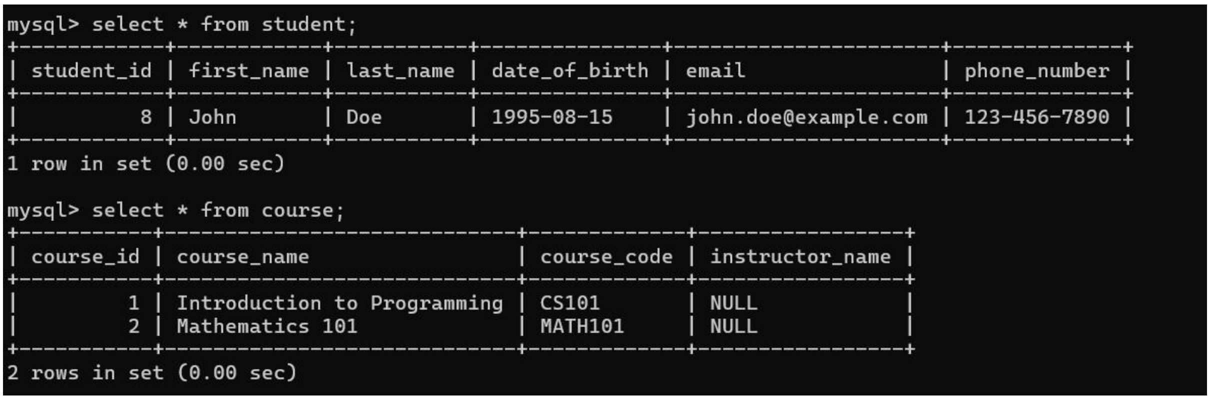
```
package com.hexaware.sis.main;
import com.hexaware.sis.dao.*; import
com.hexaware.sis.model.*;

import java.text.SimpleDateFormat;
import java.util.Date; import
java.util.Scanner;

public class SISMain {     public static
void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    StudentDAO studentDAO = new StudentDAO();
    CourseDAO courseDAO = new CourseDAO();
    EnrollmentDAO enrollmentDAO = new EnrollmentDAO();
     try
{
      // Input student details
      System.out.println("Enter First Name:");
      String firstName = sc.nextLine();

      System.out.println("Enter Last Name:");
      String lastName = sc.nextLine();

      System.out.println("Enter Date of Birth (yyyy-MM-dd):");
      String dobStr = sc.nextLine();
      Date dob = new SimpleDateFormat("yyyy-MM-dd").parse(dobStr);
      System.out.println("Enter Email:");
      String email = sc.nextLine();

      System.out.println("Enter Phone Number:");
      String phone = sc.nextLine();

      // Create student object and save to DB
      Student student = new Student(0, firstName, lastName, dob, email, phone);
int studentId = studentDAO.addStudent(student);
      System.out.println("Student added with ID: " + studentId);

      // Input course names to enroll
      System.out.println("Enter number of courses to enroll:");
int courseCount = Integer.parseInt(sc.nextLine());

      for (int i = 0; i < courseCount; i++) {
        System.out.println("Enter Course Name to enroll:");
        String courseName = sc.nextLine().trim();
        Course course = courseDAO.getCourseByName(courseName);
        if (course != null) {
          enrollmentDAO.enrollStudent(studentId, course.getCourseId(), new Date());
System.out.println("Enrolled in: " + courseName);
        } else {
          System.out.println("Course not found: " + courseName);
        }
      }
    } catch (Exception e) {
e.printStackTrace();     }
finally {       sc.close();
    }
    }
}
```

INPUT:



```
SISMain [Java Application] C:\Program Files\Java\jdk-23\bin\javaw.exe  (07-Apr-2025, 11:39:22 pm elapsed: 0:07:39) [pid: 32228]
Enter First Name:
John
Enter Last Name:
Doe
Enter Date of Birth (yyyy-MM-dd):
1995-08-15
Enter Email:
john.doe@example.com
Enter Phone Number:
123-456-7890
Student added with ID: 8
Enter number of courses to enroll:
2
Enter Course Name to enroll:
 Introduction to Programming
Enrolled in: Introduction to Programming
Enter Course Name to enroll:
Mathematics 101
Enrolled in: Mathematics 101
```

OUPUT:



```
mysql> select * from student;
+------------+------------+-----------+---------------+----------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email                | phone_number |
+------------+------------+-----------+---------------+----------------------+--------------+
|          8 | John       | Doe       | 1995-08-15    | john.doe@example.com | 123-456-7890 |
+------------+------------+-----------+---------------+----------------------+--------------+
1 row in set (0.00 sec)

mysql> select * from course;
+-----------+-----------------------------+-------------+-----------------+
| course_id | course_name                 | course_code | instructor_name |
+-----------+-----------------------------+-------------+-----------------+
|         1 | Introduction to Programming | CS101       | NULL            |
|         2 | Mathematics 101             | MATH101     | NULL            |
+-----------+-----------------------------+-------------+-----------------+
2 rows in set (0.00 sec)
```

Task 9: Teacher Assignment

TeacherDAO.java package

```
com.hexaware.sis.dao;


import com.hexaware.sis.model.Teacher; import
com.hexaware.sis.util.DBUtil;


import java.sql.*;


public class TeacherDAO {


            public int addTeacher(Teacher teacher) {
int generatedId = -1;         try (Connection conn =
DBUtil.getConnection();
                    PreparedStatement ps = conn.prepareStatement(
                       "INSERT INTO teacher (first_name, last_name, email) VALUES (?, ?, ?)",
                       Statement.RETURN_GENERATED_KEYS)) {


                   ps.setString(1, teacher.getFirstName());
ps.setString(2, teacher.getLastName());        ps.setString(3,
teacher.getEmail());


                   int rows = ps.executeUpdate();
       if (rows > 0) {
        ResultSet rs = ps.getGeneratedKeys();           if
(rs.next()) {

                generatedId = rs.getInt(1);
              }
            }
        } catch (SQLException e) {
          e.printStackTrace();
        }
        return generatedId;
      }

}
```

CourseDAO.java

```
package com.hexaware.sis.dao;

import com.hexaware.sis.model.Course; import
com.hexaware.sis.util.DBUtil;

import java.sql.*; import
java.util.ArrayList; import
java.util.List;

public    class    CourseDAO    {              public    int
addCourse(Course course) {          int courseId = -1;
try (Connection conn = DBUtil.getConnection();
        PreparedStatement stmt = conn.prepareStatement(
          "INSERT INTO course (course_name, course_code, instructor_name) VALUES (?, ?, ?)",
Statement.RETURN_GENERATED_KEYS)) {
      stmt.setString(1, course.getCourseName());

      stmt.setString(2, course.getCourseCode());

stmt.setString(3, course.getInstructorName());


      int rows = stmt.executeUpdate();

      if (rows > 0) {

        ResultSet rs = stmt.getGeneratedKeys();

        if (rs.next()) {

courseId = rs.getInt(1);

        }

      }

    } catch (SQLException e) {

      e.printStackTrace();

    }

    return courseId;

  }


  public Course getCourseByName(String courseName) {

Course  course  =  null;          try  (Connection  conn  =

DBUtil.getConnection();

       PreparedStatement stmt = conn.prepareStatement("SELECT * FROM course WHERE
course_name = ?")) {


      stmt.setString(1, courseName);

ResultSet rs = stmt.executeQuery();

      if (rs.next()) {

        course = new Course(

rs.getInt("course_id"),

rs.getString("course_name"),

rs.getString("course_code"),

rs.getString("instructor_name")
```

```
        );
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
    return course;
  }

  public Course getCourseByCode(String courseCode) {
Course  course  =  null;              try  (Connection  conn  =
DBUtil.getConnection();
      PreparedStatement stmt = conn.prepareStatement("SELECT * FROM course WHERE
course_code = ?")) {

      stmt.setString(1, courseCode);
ResultSet rs = stmt.executeQuery();
      if (rs.next()) {                course = new
Course(                    rs.getInt("course_id"),
rs.getString("course_name"),
rs.getString("course_code"),
rs.getString("instructor_name")
        );
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
    return course;
  }

  public boolean assignTeacherToCourse(int courseId, String instructorName) {
boolean updated = false;
```

```
    try (Connection conn = DBUtil.getConnection();
         PreparedStatement stmt = conn.prepareStatement("UPDATE course SET instructor_name = ?
WHERE course_id = ?")) {

      stmt.setString(1, instructorName);
stmt.setInt(2, courseId);

      int rows = stmt.executeUpdate();
updated  =  rows  >  0;          }  catch
(SQLException e) {
      e.printStackTrace();
    }
    return updated;
  }

  public List<Course> getAllCourses() {
    List<Course> courseList = new ArrayList<>();
try (Connection conn = DBUtil.getConnection();
         PreparedStatement stmt = conn.prepareStatement("SELECT * FROM course");
ResultSet rs = stmt.executeQuery()) {

      while (rs.next()) {
        Course course = new Course(
rs.getInt("course_id"),
rs.getString("course_name"),
rs.getString("course_code"),
rs.getString("instructor_name")
        );
        courseList.add(course);
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
    return courseList;
  }
}
```

SISMain.java

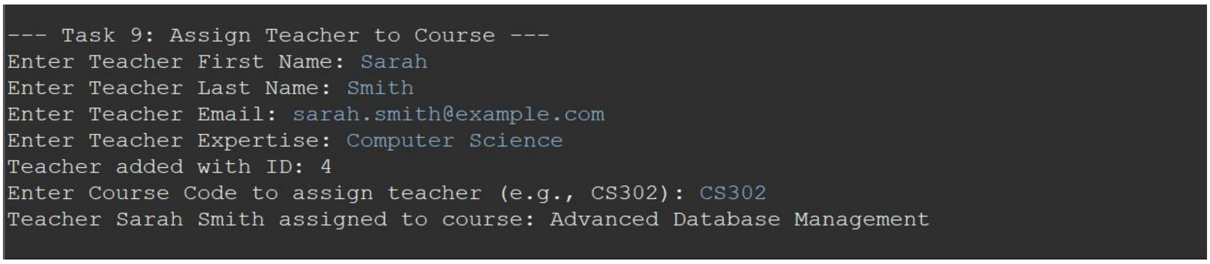
```
package com.hexaware.sis.main;

import com.hexaware.sis.dao.*; import
com.hexaware.sis.model.*;

import java.text.SimpleDateFormat;
import java.util.Date; import
java.util.Scanner;

public class SISMain {    public static
void main(String[] args) {

        //this line will call the DatabaseInitializer.java
        DatabaseInitializer.initializeDatabase();
    Scanner sc = new Scanner(System.in);
    StudentDAO studentDAO = new StudentDAO();
    CourseDAO courseDAO = new CourseDAO();
    EnrollmentDAO enrollmentDAO = new EnrollmentDAO();

    try {
      // ===== Task 8: Student Enrollment =====
      System.out.println("--- Task 8: Student Enrollment ---");

      // Input student details
```

```
        System.out.println("Enter First Name:");
        String firstName = sc.nextLine();

        System.out.println("Enter Last Name:");
        String lastName = sc.nextLine();

        System.out.println("Enter Date of Birth (yyyy-MM-dd):");
        String dobStr = sc.nextLine();
        Date dob = new SimpleDateFormat("yyyy-MM-dd").parse(dobStr);

        System.out.println("Enter Email:");
        String email = sc.nextLine();

        System.out.println("Enter Phone Number:");
        String phone = sc.nextLine();

        // Create student object and save to DB
        Student student = new Student(0, firstName, lastName, dob, email, phone);
int studentId = studentDAO.addStudent(student);
        System.out.println("Student added with ID: " + studentId);

        // Input course names to enroll
        System.out.println("Enter number of courses to enroll:");
int courseCount = Integer.parseInt(sc.nextLine());

        for (int i = 0; i < courseCount; i++) {
            System.out.println("Enter Course Name to enroll:");
            String courseName = sc.nextLine().trim();
            Course course = courseDAO.getCourseByName(courseName);
            if (course != null) {
```

```
            enrollmentDAO.enrollStudent(studentId, course.getCourseId(), new Date());
System.out.println("Enrolled in: " + courseName);
          } else {
            System.out.println("Course not found: " + courseName);
          }
        }

        // ===== Task 9: Teacher Assignment =====
        System.out.println("\n--- Task 9: Assign Teacher to Course ---");

        System.out.print("Enter Teacher First Name: ");
        String teacherFirstName = sc.nextLine();

        System.out.print("Enter Teacher Last Name: ");
        String teacherLastName = sc.nextLine();

        System.out.print("Enter Teacher Email: ");
        String teacherEmail = sc.nextLine();

        System.out.print("Enter Teacher Expertise: ");
        String expertise = sc.nextLine();

        Teacher teacher = new Teacher(0, teacherFirstName, teacherLastName, teacherEmail);
TeacherDAO teacherDAO = new TeacherDAO();                      int teacherId =
teacherDAO.addTeacher(teacher);
        if (teacherId != -1) {
teacher.setTeacherId(teacherId);
          System.out.println("Teacher added with ID: " + teacherId);
        } else {
          System.out.println("Failed to add teacher.");
          return;
```

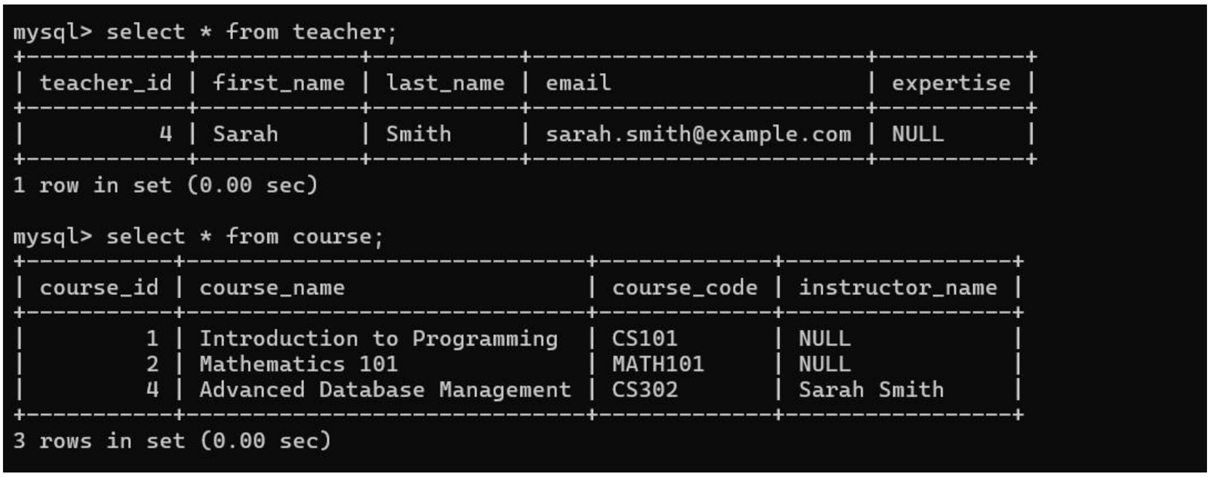
```
            }


            System.out.print("Enter Course Code to assign teacher (e.g., CS302): ");
String courseCode = sc.nextLine();


            Course courseToUpdate = courseDAO.getCourseByCode(courseCode);


            if (courseToUpdate != null) {
                boolean updated = courseDAO.assignTeacherToCourse(courseToUpdate.getCourseId(),
teacher.getFullName());
                if (updated) {
                    System.out.println("Teacher " + teacher.getFullName() +
                            " assigned to course: " + courseToUpdate.getCourseName());
                } else {
                    System.out.println("Failed to assign teacher to course.");
                }
            } else {
                System.out.println("Course not found with code: " + courseCode);
            }


        } catch (Exception e) {
            e.printStackTrace();
} finally {            sc.close();
        }
    }
}
```

INPUT:



```
--- Task 9: Assign Teacher to Course ---
Enter Teacher First Name: Sarah
Enter Teacher Last Name: Smith
Enter Teacher Email: sarah.smith@example.com
Enter Teacher Expertise: Computer Science
Teacher added with ID: 4
Enter Course Code to assign teacher (e.g., CS302): CS302
Teacher Sarah Smith assigned to course: Advanced Database Management
```

OUTPUT:



```
mysql> select * from teacher;
+------------+------------+-----------+-------------------------+-----------+
| teacher_id | first_name | last_name | email                   | expertise |
+------------+------------+-----------+-------------------------+-----------+
|          4 | Sarah      | Smith     | sarah.smith@example.com | NULL      |
+------------+------------+-----------+-------------------------+-----------+
1 row in set (0.00 sec)

mysql> select * from course;
+-----------+------------------------------+-------------+-----------------+
| course_id | course_name                  | course_code | instructor_name |
+-----------+------------------------------+-------------+-----------------+
|         1 | Introduction to Programming  | CS101       | NULL            |
|         2 | Mathematics 101              | MATH101     | NULL            |
|         4 | Advanced Database Management | CS302       | Sarah Smith     |
+-----------+------------------------------+-------------+-----------------+
3 rows in set (0.00 sec)
```

Task 10: Payment Record

Payment.java

```
package com.hexaware.sis.model;

import java.util.Date;

public class Payment {
   private     int     paymentId;
private       int       studentId;
private     double     amount;
private Date paymentDate;

   public Payment(int paymentId, int studentId, double amount, Date paymentDate) {
this.paymentId = paymentId;         this.studentId = studentId;         this.amount =
amount;         this.paymentDate = paymentDate;
   }

   // Getters and setters     public int getPaymentId() { return paymentId; }
public void setPaymentId(int paymentId) { this.paymentId = paymentId; }

   public int getStudentId() { return studentId; }     public void
setStudentId(int studentId) { this.studentId = studentId; }

   public double getAmount() { return amount; }     public void
setAmount(double amount) { this.amount = amount; }

   public Date getPaymentDate() { return paymentDate; }     public void
setPaymentDate(Date paymentDate) { this.paymentDate = paymentDate; }
}
```

Teacher.java

```
package com.hexaware.sis.model;

public class Teacher {
private int teacherId;
private String firstName;
private String lastName;
private String email;
private String expertise;

   //  Add this constructor
   public Teacher(int teacherId, String firstName, String lastName, String email, String
expertise) {        this.teacherId = teacherId;        this.firstName = firstName;
this.lastName = lastName;        this.email = email;        this.expertise = expertise;
   }

   // Getters and setters
public int getTeacherId() {
return teacherId;
   }

   public void setTeacherId(int teacherId) {
this.teacherId = teacherId;
   }

   public String getFirstName() {
return firstName;
   }

   public void setFirstName(String firstName) {
this.firstName = firstName;
   }

   public String getLastName() {
return lastName;
   }
```

```
    public void setLastName(String lastName) {
this.lastName = lastName;
    }

    public String getEmail() {
return email;
    }

    public void setEmail(String email) {
this.email = email;
    }

    public String getExpertise() {
return expertise;
    }

    public void setExpertise(String expertise) {
this.expertise = expertise;
    }

    // Optional: helper method for full name
public String getFullName() {         return
firstName + " " + lastName;
    }
}
```

PaymentDAO.java

```
package com.hexaware.sis.dao;

import com.hexaware.sis.model.Payment; import
com.hexaware.sis.util.DBUtil;

import java.sql.Connection; import
java.sql.PreparedStatement;

public class PaymentDAO {     public boolean
addPayment(Payment payment) {
        String sql = "INSERT INTO payment (student_id, amount, payment_date) VALUES (?, ?,
?)";
        try (Connection conn = DBUtil.getConnection();
             PreparedStatement ps = conn.prepareStatement(sql)) {          ps.setInt(1,
payment.getStudentId());                    ps.setDouble(2,  payment.getAmount());
ps.setDate(3,      new      java.sql.Date(payment.getPaymentDate().getTime()));
int rows = ps.executeUpdate();           return rows > 0;        } catch (Exception e)
{
            e.printStackTrace();
        }
        return false;
    }
}
```

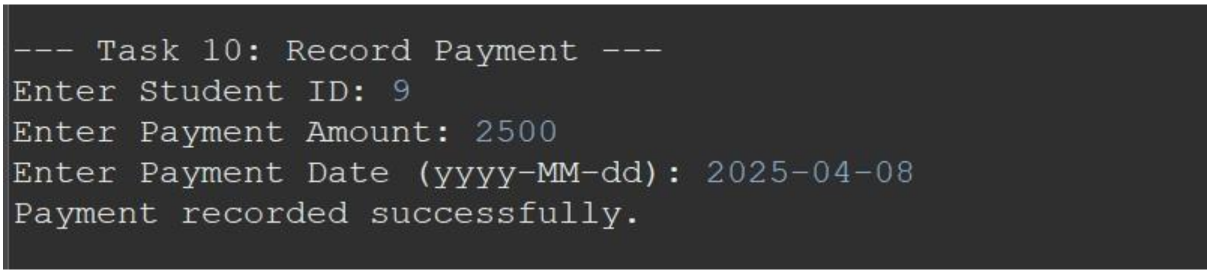
SISMain.java

```
package com.hexaware.sis.main;

import com.hexaware.sis.dao.*; import
com.hexaware.sis.model.*;
```

```
import java.text.SimpleDateFormat;

import java.util.Date; import

java.util.Scanner;


public class SISMain {     public static

void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        StudentDAO studentDAO = new StudentDAO();

        CourseDAO courseDAO = new CourseDAO();

        EnrollmentDAO enrollmentDAO = new EnrollmentDAO();

        TeacherDAO teacherDAO = new TeacherDAO();

        PaymentDAO paymentDAO = new PaymentDAO();


        try {

            // --- Task 8: Student Enrollment ---

            System.out.println("--- Task 8: Student Enrollment ---");
System.out.println("Enter First Name:");

            String firstName = sc.nextLine();


            System.out.println("Enter Last Name:");

            String lastName = sc.nextLine();


            System.out.println("Enter Date of Birth (yyyy-MM-dd):");

            String dobStr = sc.nextLine();

            Date dob = new SimpleDateFormat("yyyy-MM-dd").parse(dobStr);


            System.out.println("Enter Email:");

            String email = sc.nextLine();
            System.out.println("Enter Phone Number:");

            String phone = sc.nextLine();


            Student student = new Student(0, firstName, lastName, dob, email, phone);

int studentId = studentDAO.addStudent(student);

            System.out.println("Student added with ID: " + studentId);


            System.out.println("Enter number of courses to enroll:");

int courseCount = Integer.parseInt(sc.nextLine());


            for (int i = 0; i < courseCount; i++) {

              System.out.println("Enter Course Name to enroll:");

              String courseName = sc.nextLine().trim();

              Course course = courseDAO.getCourseByName(courseName);

              if (course != null) {

                enrollmentDAO.enrollStudent(studentId, course.getCourseId(), new Date());
System.out.println("Enrolled in: " + courseName);

              } else {

                System.out.println("Course not found: " + courseName);

              }

            }


            // --- Task 9: Assign Teacher to Course ---

            System.out.println("\n--- Task 9: Assign Teacher to Course ---");


            System.out.print("Enter Teacher First Name: ");

            String tFirstName = sc.nextLine();


            System.out.print("Enter Teacher Last Name: ");

            String tLastName = sc.nextLine();
            System.out.print("Enter Teacher Email: ");
            String tEmail = sc.nextLine();

            System.out.print("Enter Teacher Expertise: ");
            String expertise = sc.nextLine();

            Teacher teacher = new Teacher(0, tFirstName, tLastName, tEmail, expertise);
int teacherId = teacherDAO.addTeacher(teacher);
            if (teacherId != -1) {
teacher.setTeacherId(teacherId);
              System.out.println("Teacher added with ID: " + teacherId);
            } else {
              System.out.println("Failed to add teacher.");
              return;
            }

            System.out.print("Enter Course Code to assign teacher (e.g., CS302): ");
String courseCode = sc.nextLine();

            Course course = courseDAO.getCourseByCode(courseCode);

            if (course != null) {
              boolean updated = courseDAO.assignTeacherToCourse(course.getCourseId(),
teacher.getFirstName() + " " + teacher.getLastName());
              if (updated) {
                System.out.println("Teacher " + teacher.getFirstName() + " " +
teacher.getLastName() +
                " assigned to course: " + course.getCourseName());
              } else {
                System.out.println("Failed to assign teacher to course.");
                }
            } else {
                System.out.println("Course not found with code: " + courseCode);
            }

            // --- Task 10: Record Payment ---
            System.out.println("\n--- Task 10: Record Payment ---");

            System.out.print("Enter Student ID: ");            int
payStudentId = Integer.parseInt(sc.nextLine());

            System.out.print("Enter Payment Amount: ");
double amount = Double.parseDouble(sc.nextLine());
System.out.print("Enter Payment Date (yyyy-MM-dd): ");
            String paymentDateStr = sc.nextLine();
            Date paymentDate = new SimpleDateFormat("yyyy-MM-dd").parse(paymentDateStr);

            Payment payment = new Payment(0, payStudentId, amount, paymentDate);
boolean paymentSuccess = paymentDAO.addPayment(payment);
            if (paymentSuccess) {
                System.out.println("Payment recorded successfully.");
            } else {
                System.out.println("Failed to record payment.");
            }

          } catch (Exception e) {
            e.printStackTrace();
} finally {            sc.close();
```

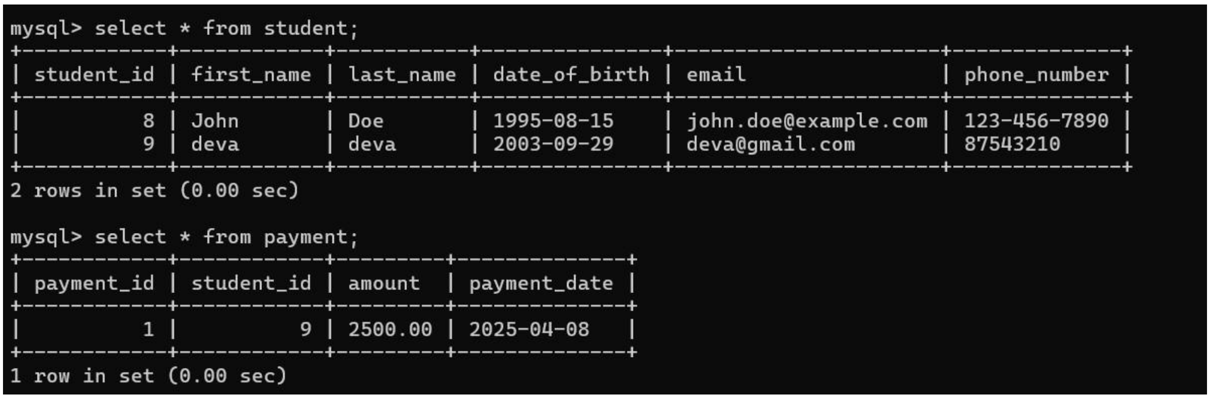
```
        }
    }
}
```

INPUT:



```
--- Task 10: Record Payment ---
Enter Student ID: 9
Enter Payment Amount: 2500
Enter Payment Date (yyyy-MM-dd): 2025-04-08
Payment recorded successfully.
```

OUTPUT:



```
mysql> select * from student;
+------------+------------+-----------+---------------+----------------------+--------------+
| student_id | first_name | last_name | date_of_birth | email                | phone_number |
+------------+------------+-----------+---------------+----------------------+--------------+
|          8 | John       | Doe       | 1995-08-15    | john.doe@example.com | 123-456-7890 |
|          9 | deva       | deva      | 2003-09-29    | deva@gmail.com       | 87543210     |
+------------+------------+-----------+---------------+----------------------+--------------+
2 rows in set (0.00 sec)

mysql> select * from payment;
+------------+------------+---------+--------------+
| payment_id | student_id | amount  | payment_date |
+------------+------------+---------+--------------+
|          1 |          9 | 2500.00 | 2025-04-08   |
+------------+------------+---------+--------------+
1 row in set (0.00 sec)
```

Task 11: Enrollment Report Generation

EnrollmentDAO.java

```
package com.hexaware.sis.dao;

import com.hexaware.sis.model.Student; import
com.hexaware.sis.util.DBUtil;
```

```
import java.sql.*; import
java.util.ArrayList; import
java.util.Date; import
java.util.List;

public class EnrollmentDAO {

   // Method to enroll a student in a course     public void
enrollStudent(int studentId, int courseId, Date date) {
       String sql = "INSERT INTO enrollment (student_id, course_id, enrollment_date) VALUES
(?, ?, ?)";
       try (Connection conn = DBUtil.getConnection();
            PreparedStatement ps = conn.prepareStatement(sql)) {
           ps.setInt(1,  studentId);                        ps.setInt(2,
courseId);                                            ps.setDate(3,    new
java.sql.Date(date.getTime()));
ps.executeUpdate();        } catch (SQLException e) {
           e.printStackTrace();
       }
   }

   // Method to retrieve students enrolled in a specific course by course name
public List<Student> getEnrolledStudentsByCourseName(String courseName) {
       List<Student> students = new ArrayList<>();
 String query = "SELECT s.student_id, s.first_name, s.last_name, s.email, s.phone_number " +
                    "FROM student s " +
                    "JOIN enrollment e ON s.student_id = e.student_id " +
                    "JOIN course c ON e.course_id = c.course_id " +
                    "WHERE c.course_name = ?";         try
(Connection conn = DBUtil.getConnection();
           PreparedStatement ps = conn.prepareStatement(query)) {
ps.setString(1,   courseName);                             ResultSet   rs   =
ps.executeQuery();
            while (rs.next()) {
                Student student = new Student(
                    rs.getInt("student_id"),
rs.getString("first_name"),
rs.getString("last_name"),                     null, // dob is
not required for this report
rs.getString("email"),
rs.getString("phone_number")
                );
                students.add(student);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
        return students;
    }
}
```

Student.java

```
package com.hexaware.sis.model;

import java.util.Date;

public class Student {     private
int studentId;     private String
```

```
firstName;     private String
lastName;     private Date
dateOfBirth;     private String
email;     private String
phoneNumber;

   // Full constructor
   public Student(int studentId, String firstName, String lastName, Date dateOfBirth, String
email, String phoneNumber) {
       this.studentId = studentId;
       this.firstName    =    firstName;
this.lastName        =        lastName;
this.dateOfBirth     =     dateOfBirth;
this.email = email;
       this.phoneNumber = phoneNumber;
   }

   // Constructor without studentId
   public Student(String firstName, String lastName, Date dateOfBirth, String email, String
phoneNumber) {
       this(0, firstName, lastName, dateOfBirth, email, phoneNumber);
   }

   //  Getters         public  int
getStudentId() {         return
studentId;
   }

   public String getFirstName() {
return firstName;
```

```
    }

    public String getLastName() {
return lastName;
    }

    public Date getDateOfBirth() {
return dateOfBirth;
    }

    public String getEmail() {
return email;
    }

    public String getPhoneNumber() {
return phoneNumber;
    }

    // Alias for consistency with SISMain
public String getPhone() {          return
phoneNumber;
    }

    // Setters
    public void setStudentId(int studentId) {
this.studentId = studentId;
    }

    public void setFirstName(String firstName) {
this.firstName = firstName;
```

```
   }

   public void setLastName(String lastName) {
this.lastName = lastName;
   }

   public void setDateOfBirth(Date dateOfBirth) {
this.dateOfBirth = dateOfBirth;
   }
   public void setEmail(String email) {
this.email = email;
   }

   public void setPhoneNumber(String phoneNumber) {
this.phoneNumber = phoneNumber;
   }
}
```

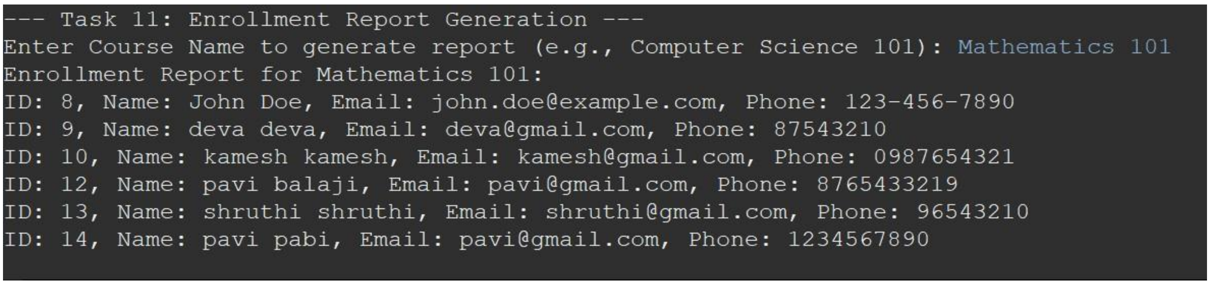
SISMain.java

```
package com.hexaware.sis.main;

import com.hexaware.sis.dao.*; import
com.hexaware.sis.model.*;

import java.text.SimpleDateFormat;
import java.util.Date; import
java.util.Scanner; import
java.util.List;
```

```
public class SISMain {     public static
void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        StudentDAO studentDAO = new StudentDAO();
        CourseDAO courseDAO = new CourseDAO();
        EnrollmentDAO enrollmentDAO = new EnrollmentDAO();        TeacherDAO teacherDAO =
new TeacherDAO();
        PaymentDAO paymentDAO = new PaymentDAO();

        try {
            // --- Task 8: Student Enrollment ---
            System.out.println("--- Task 8: Student Enrollment ---");

            System.out.println("Enter First Name:");
            String firstName = sc.nextLine();

            System.out.println("Enter Last Name:");
            String lastName = sc.nextLine();

            System.out.println("Enter Date of Birth (yyyy-MM-dd):");
            String dobStr = sc.nextLine();
            Date dob = new SimpleDateFormat("yyyy-MM-dd").parse(dobStr);

            System.out.println("Enter Email:");
            String email = sc.nextLine();

            System.out.println("Enter Phone Number:");
            String phone = sc.nextLine();
        Student student = new Student(0, firstName, lastName, dob, email, phone);
int studentId = studentDAO.addStudent(student);
        System.out.println("Student added with ID: " + studentId);

        System.out.println("Enter number of courses to enroll:");
int courseCount = Integer.parseInt(sc.nextLine());

        for (int i = 0; i < courseCount; i++) {
          System.out.println("Enter Course Name to enroll:");
          String courseName = sc.nextLine().trim();
          Course course = courseDAO.getCourseByName(courseName);
          if (course != null) {
            enrollmentDAO.enrollStudent(studentId, course.getCourseId(), new Date());
System.out.println("Enrolled in: " + courseName);
          } else {
            System.out.println("Course not found: " + courseName);
          }
        }

        // --- Task 9: Assign Teacher to Course ---
        System.out.println("\n--- Task 9: Assign Teacher to Course ---");

        System.out.print("Enter Teacher First Name: ");
        String tFirstName = sc.nextLine();

        System.out.print("Enter Teacher Last Name: ");
        String tLastName = sc.nextLine();

        System.out.print("Enter Teacher Email: ");
        String tEmail = sc.nextLine();
        System.out.print("Enter Teacher Expertise: ");
        String expertise = sc.nextLine();

        Teacher teacher = new Teacher(0, tFirstName, tLastName, tEmail, expertise);
        int teacherId = teacherDAO.addTeacher(teacher);
        if (teacherId != -1) {
teacher.setTeacherId(teacherId);
            System.out.println("Teacher added with ID: " + teacherId);
        } else {
            System.out.println("Failed to add teacher.");
            return;
        }

        System.out.print("Enter Course Code to assign teacher (e.g., CS302): ");
String courseCode = sc.nextLine();

        Course course = courseDAO.getCourseByCode(courseCode);

        if (course != null) {
            boolean updated = courseDAO.assignTeacherToCourse(course.getCourseId(),
teacher.getFirstName() + " " + teacher.getLastName());
            if (updated) {
                System.out.println("Teacher " + teacher.getFirstName() + " " +
teacher.getLastName() +
                        " assigned to course: " + course.getCourseName());
            } else {
                System.out.println("Failed to assign teacher to course.");
            }
        } else {
            System.out.println("Course not found with code: " + courseCode);
        }

        // --- Task 10: Record Payment ---
        System.out.println("\n--- Task 10: Record Payment ---");           System.out.print("Enter
Student ID: ");            int payStudentId = Integer.parseInt(sc.nextLine());

        System.out.print("Enter Payment Amount: ");
double amount = Double.parseDouble(sc.nextLine());

        System.out.print("Enter Payment Date (yyyy-MM-dd): ");
        String paymentDateStr = sc.nextLine();
        Date paymentDate = new SimpleDateFormat("yyyy-MM-dd").parse(paymentDateStr);

        Payment payment = new Payment(0, payStudentId, amount, paymentDate);
boolean paymentSuccess = paymentDAO.addPayment(payment);
        if (paymentSuccess) {
            System.out.println("Payment recorded successfully.");
        } else {
            System.out.println("Failed to record payment.");
        }

        System.out.println("--- Task 11: Enrollment Report Generation ---");
        System.out.print("Enter Course Name to generate report (e.g., Computer Science 101):
");
        String courseName = sc.nextLine().trim();

        List<Student> enrolledStudents =
enrollmentDAO.getEnrolledStudentsByCourseName(courseName);
```

```
            if (enrolledStudents.isEmpty()) {
                System.out.println("No students enrolled in: " + courseName);
            } else {
                System.out.println("Enrollment Report for " + courseName + ":");
for (Student s : enrolledStudents) {
                    System.out.println("ID: " + s.getStudentId() + ", Name: " + s.getFirstName() + " " +
s.getLastName()
                            + ", Email: " + s.getEmail() + ", Phone: " + s.getPhone());
                }
            }



        } catch (Exception e) {
            e.printStackTrace();
} finally {            sc.close();
        }
    }
}
```

OUTPUT:



```
--- Task 11: Enrollment Report Generation ---
Enter Course Name to generate report (e.g., Computer Science 101): Mathematics 101
Enrollment Report for Mathematics 101:
ID: 8, Name: John Doe, Email: john.doe@example.com, Phone: 123-456-7890
ID: 9, Name: deva deva, Email: deva@gmail.com, Phone: 87543210
ID: 10, Name: kamesh kamesh, Email: kamesh@gmail.com, Phone: 0987654321
ID: 12, Name: pavi balaji, Email: pavi@gmail.com, Phone: 8765433219
ID: 13, Name: shruthi shruthi, Email: shruthi@gmail.com, Phone: 96543210
ID: 14, Name: pavi pabi, Email: pavi@gmail.com, Phone: 1234567890
```