Student Management System - Java Study Plan 📚

📋 **Project Overview**

**Goal:** Build a Console-Based Student Management System in Java **Users:** Admin & Student with role-based functionalities **Timeline:** [Add your deadline here]

🎯 **Phase 1: Foundation & Planning (Days 1-2)**

**Core Java Concepts Review**

 **OOP Principles**

 Classes and Objects

 Inheritance, Polymorphism, Encapsulation  Abstract classes vs Interfaces

 **Collections Framework**

 ArrayList, HashMap, LinkedList  When to use which collection

 **File I/O Operations**

 Reading/Writing to files

 Data persistence strategies

**Project Planning**

 **System Design**

 Sketch user flow diagrams

 Define class structure (User, Admin, Student, Course, Subject, Exam)  Plan data storage approach (files vs in-memory)

 **Menu Structure Planning**

 Admin menu options

 Student menu options  Navigation flow

🏗 **Phase 2: Core Implementation (Days 3-5)**

**Day 3: User Management & Authentication**

 **Create Base Classes**

(parent class) and

User

Admin

Student

(child classes)

 Basic login/registration system

 **Implementation Tasks:**

 User registration with validation  Simple login mechanism

 Role-based menu display

**Day 4: Course & Subject Management**

 **Create Course System**

and classes

Course

Subject

 Admin functions: add/manage courses and subjects  Data storage solution

 **Implementation Tasks:**

 Add course functionality  Add subjects to courses  Display available courses

**Day 5: Student Features**

 **Student Registration & Course Selection ** Student profile creation

 Course browsing and selection

 Subject selection within chosen course  **Implementation Tasks:**

 Student registration form  Course selection menu

 Subject enrollment system

📝 **Phase 3: Exam System (Days 6-7)**

**Exam Implementation**

 **Create Exam Classes**

class (MCQ structure) class (collection of questions)

Question

Exam

class (score tracking)  **Implementation Tasks:**

Result

 Question bank creation (minimum 5 MCQs per subject)  Exam taking interface

 Score calculation and pass/fail logic  Result storage and retrieval

🔧 **Phase 4: Integration & Polish (Day 8)**

**System Integration**

 **Connect All Components**

 Ensure smooth navigation between menus  Test all user flows (Admin and Student)

 Handle edge cases and input validation  **Data Persistence**

 Implement file-based storage

 Ensure data survives program restarts  **Error Handling**

 Try-catch blocks for user inputs

 Graceful handling of invalid operations

🧪 **Phase 5: Testing & Documentation (Day 9-10)**

**Testing Checklist**

 **Admin Functions**

 ✅ Add courses and subjects

 ✅ View all registered students  ✅ View exam results

 **Student Functions**

 ✅ Registration process

 ✅ Course and subject selection

 ✅ Exam taking (5+ MCQs)  ✅ Result viewing

 **System Tests**

 ✅ Multiple user sessions  ✅ Data persistence

 ✅ Input validation

**Documentation**

 **Code Documentation**

 Add comments to complex methods

 Create README with setup instructions  **User Manual**

 How to run the application

 Feature walkthrough for both user types

💡 **Key Implementation Tips**

**Code Structure Suggestions**

ssrrcc//

├├──── mmooddeellss//

││ ├├────

││ ├├────

││ ├├────

││ ├├────

││ ├├────

││ └└────

UUsseerr..jjaavvaa AAddmmiinn..jjaavvaa SSttuuddeenntt..jjaavvaa CCoouurrssee..jjaavvaa SSuubbjjeecctt..jjaavvaa EExxaamm..jjaavvaa

├├──── sseerrvviicceess//

││ ├├────

││ ├├────

││ └└────

UUsseerrSSeerrvviiccee..jjaavvaa CCoouurrsseeSSeerrvviiccee..jjaavvaa EExxaammSSeerrvviiccee..jjaavvaa

├├──── uuttiillss//

││ ├├──── IInputValidator.javaavaj.rotadilaVtupn

││ └└──── FFileHandler.javaavaj.reldnaHeli

└└──── MMaaiinn..jjaavvaa

**Essential Java Features to Use**

**Scanner** for user input **ArrayList/HashMap** for data storage **File I/O** for persistence

**Exception handling** for robust code **Method overloading** where appropriate

📚 **Study Resources**

**Quick References**

 Oracle Java Documentation  Java Collections Tutorial

 File I/O in Java guides

**Practice Before Implementation**

 Create a simple login system

 Practice with ArrayLists and HashMaps  File reading/writing exercises

✅ **Daily Progress Tracker**

 Day 1: Foundation Review \_\_\_% complete  Day 2: Project Planning \_\_\_% complete  Day 3: User Management \_\_\_% complete  Day 4: Course System \_\_\_% complete

 Day 5: Student Features \_\_\_% complete  Day 6: Exam System Part 1 \_\_\_% complete

 Day 7: Exam System Part 2 \_\_\_% complete  Day 8: Integration \_\_\_% complete

 Day 9: Testing \_\_\_% complete

 Day 10: Final Polish \_\_\_% complete

**Remember:** Start coding early, test frequently, and don't hesitate to refactor as you learn!

OUTPUT



