

## Java OOP Student Manager – Task Sheet


### Project Info:

Project Name: StudentManagerSystem


Language: Java

Goal: Build a simple student management system using Java and Object-Oriented Programming concepts such as classes, constructors, encapsulation, collections, and method-based logic. This project helps strengthen your understanding of how real-world objects can be represented and manipulated in code.

### Project Steps (with GitHub):

 Suggested Repository Name: **StudentManagerSystem**

- `git init` → Start tracking your project with Git
- `git add .` → Prepare all files for commit
- `git commit -m "Initial commit"` → Save your first version
- `git remote add origin https://github.com/your-username/StudentManagerSystem.git` → Connect local repo to GitHub
- `git branch -M main` → Rename default branch to main
- `git push -u origin main` → Push your code to GitHub and track it
- `git status` → See what files are modified, staged, or new

 Example commit messages:

- `git commit -m "Add Student class with constructor and getters"`
- `git commit -m "Implement addStudent option in Main menu"`

 ***After each feature or function, commit your work to keep track clearly.***

### Project Menu Options:

1. Add Student
2. View All Students
3. Update Student Grade
4. Delete Student
5. Search Student by ID
6. Search Student by Name
7. Calculate Average Grade
8. Show Highest Grade
9. Show Lowest Grade

10. Count Total Students
11. Sort Students by Grade
12. Sort Students by Name
13. Clear All Students
14. Exit

## Code Structure (Functions Overview):

### *Student.java*

---

- Student(String name, int id, double grade) → Constructor to initialize student attributes
- getName() → Returns the student's name
- getId() → Returns the student's ID
- getGrade() → Returns the student's grade
- setGrade(double grade) → Updates the student's grade
- displayInfo() → Prints the student's details in a formatted string

### *Main.java*

---

```
public class Main {  
  
    public static void main(String[] args) {  
  
        Scanner scanner = new Scanner(System.in);  
  
        ArrayList<Student> students = new ArrayList<>();  
  
        boolean exit = false;  
  
        while (!exit) {  
  
            // 1. Display menu options  
  
            // 2. Take user input (switch case)  
  
            // 3. Perform add, update, delete, search, sort, etc.  
  
            // 4. Use methods like: addStudent(), viewStudents(), searchById(), etc.  
  
        }  
    }  
}
```

## Review & Coding Practice

### ✓ Review Yourself:

- Do you understand how classes and objects work?
- Can you explain why private is used in class attributes?
- What's the purpose of a constructor in Java?
- Why use ArrayList instead of a fixed-size array?
- When should you use scanner.next() vs scanner.nextLine()?
- How is a Lambda Expression used in this project?
- How does Comparator help sort the list?
- What's the difference between == and .equals() in Java?
- How would you add, update, and delete students in this system?
- What does each menu option represent logically?

### ✦ Bonus Missions (Challenge Yourself):

- Show students who failed (grade < 60)
- Add a feature to update a student's name using ID
- Create a confirmation prompt before deleting a student
- Count how many students have grades between 80 and 90
- Return the student with the second highest grade
- Add a login system before showing the menu (admin only)
- Create an export option to save student data in a file
- Group students by performance level (Excellent, Good, Fail)

✦ رأيي في كتابة الكود:

الكود الزين مش بس يشتغل، الكود الزين يشرح نفسه !  
أكتبه اليوم وأنا واثق إنني بعد أسبوع، أو أي شخص غيري، يقدر يفهمه بسهولة — لأن التعليق المناسب والتنسيق النظيف هم سر الكود الممتاز.

📝 مثال على تعليق بسيط وواضح في الكود:

```
// This method calculates the average grade of all students  
double average = calculateAverageGrade(students);
```