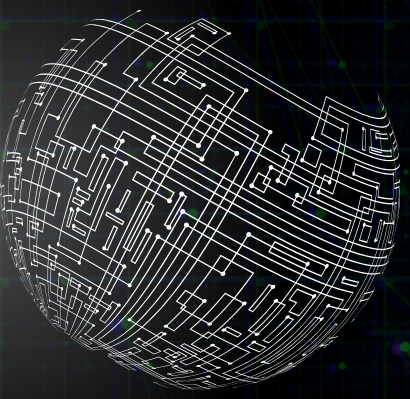

JAVA PROGRAMMING PROJECT LIST

@motioncut www.motioncut.in

Project Title: Student Skill Evaluator – Assess and Grade Skills with Java

To build a **Java console-based application** that allows evaluation of a student's performance based on input scores across multiple skill areas. The program will calculate averages, assign grades, and provide a basic performance analysis. This project strengthens understanding of arrays, input/output, conditionals, and modular programming.

1. Tasks / Requirements

Core Features to Include:

- Ask the user to enter:
  - Student's name and ID
  - Marks in different skill areas (e.g., Programming, Communication, Problem-Solving, Teamwork, Creativity)
- Calculate:
  - Total marks
  - Average score
  - Grade based on average
- Display:
  - Student info
  - Individual skill scores
  - Total, average, and grade
  - Personalized feedback message based on performance

Sample Grading Logic:

- **90 – 100**: Grade A (Excellent)

- **75 – 89**: Grade B (Good)
- **60 – 74**: Grade C (Average)
- **40 – 59**: Grade D (Needs Improvement)
- **Below 40**: Grade F (Fail)

Optional Enhancements:

- Evaluate multiple students in one session
- Store student data using arrays or ArrayList
- Display top performer among all entries
- Export results to a .txt file (file handling)
- Use color-coded console output (if supported)

2. Tools to Use

- **Language:** Java (SE 8 or higher)
- **IDE:** IntelliJ IDEA, Eclipse, BlueJ, VS Code, or Replit
- **Java Classes & Methods:**
  - Scanner for user input
  - Arrays or ArrayList for storing scores
  - Math class for calculations
  - System.out.printf() for formatted output
  - (Optional) FileWriter, BufferedWriter for saving results

3. Concepts Used in This Project

- Java program structure and syntax
- Arrays or ArrayLists for skill score storage
- Loops (for, while) for input and processing
- Conditional statements for grading logic
- Input handling with Scanner
- Basic mathematical operations and average calculation
- String formatting and structured console output
- Optional: File I/O operations and sorting logic
- Modular code using methods (e.g., calculateGrade(), printReport())