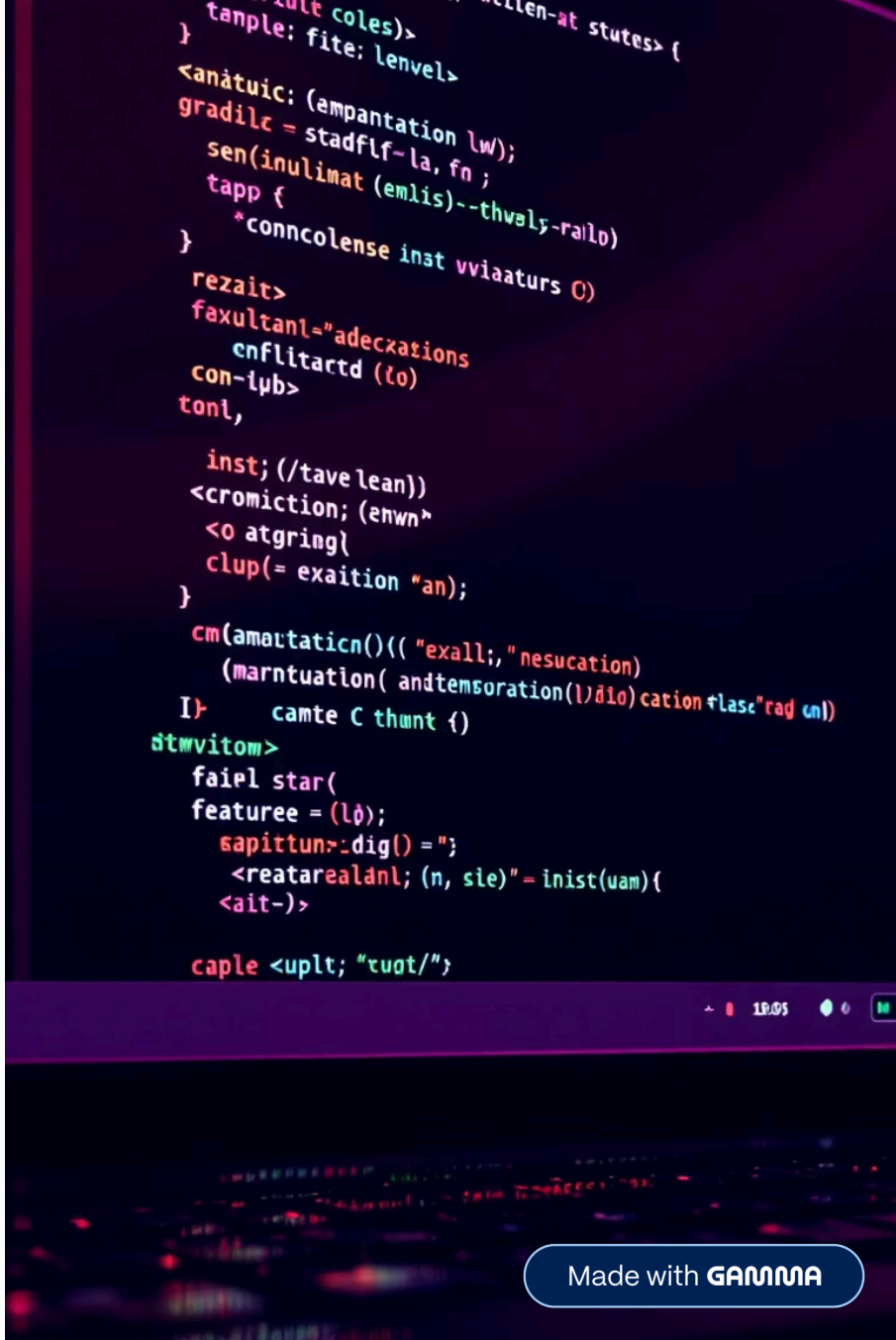
Grade Calculator Using C Programming

Sudhanshu Deshmukh
B.Tech CSE (Artificial Intelligence & Machine Learning)
1st Semester
Rungta International Skills University, Bhilai



INTRODUCTION

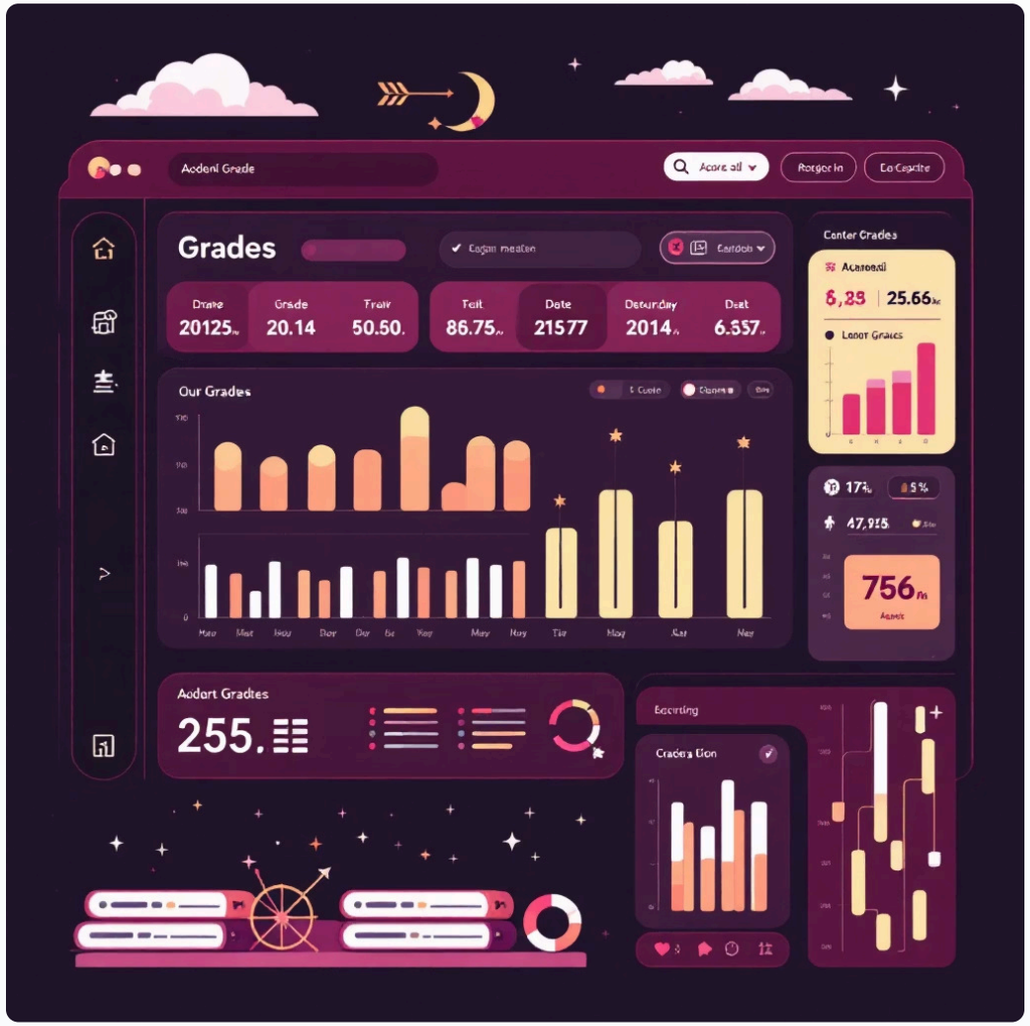
Understanding Grade Calculators

What is a Grade Calculator?

A grade calculator is an automated system that processes student marks, computes totals and percentages, and assigns appropriate grades based on predefined criteria. It eliminates manual calculations and reduces human error.

The Role of C Programming

C programming provides the foundational logic and control structures necessary for building efficient academic tools. Its array handling and conditional statements make it ideal for grade calculation systems.



Project Objectives

Accept Student Marks

Input marks for five subjects, each ranging from 0 to 100 marks, ensuring comprehensive assessment across multiple disciplines.

Perform Calculations

Automatically calculate total marks obtained out of 500 and compute the percentage score for overall performance evaluation.

Grade Assignment

Assign appropriate letter grades (A through F) based on percentage thresholds and determine pass or fail status accordingly.

PROBLEM STATEMENT

Defining the Challenge

Input Requirements

Accept marks for five subjects, with each subject's marks constrained between 0 and 100. The system must handle numerical input efficiently.

Validation Process

Implement robust input validation to ensure all marks fall within acceptable ranges, preventing invalid data from corrupting the calculation process.

Automated Results

Generate comprehensive results automatically, including total marks, percentage, assigned grade, and pass/fail status without manual intervention.

Algorithm Design

Initialisation

Begin programme execution and declare necessary variables including marks array, total, percentage, and grade.

Input & Validation

Accept marks for five subjects using a loop structure. Validate each entry to ensure marks are within 0–100 range.

Computation

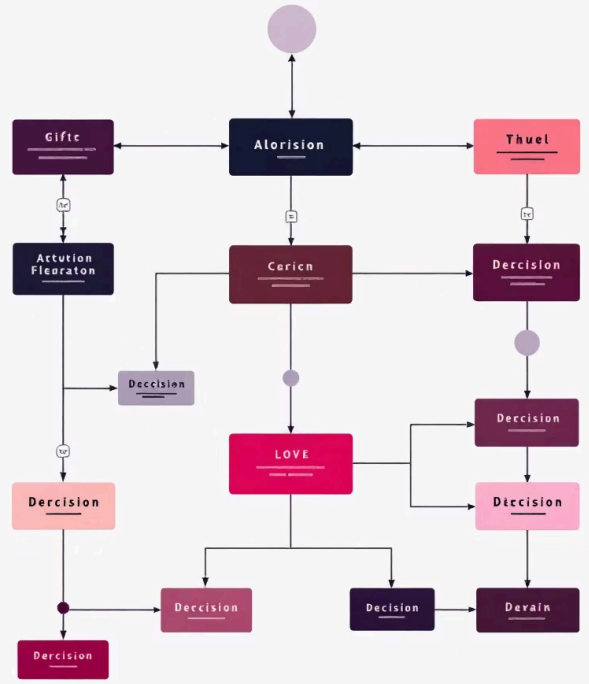
Calculate total marks by summing all subject marks. Compute percentage by dividing total by 5.

Grade Assignment

Use conditional statements to assign appropriate grade (A–F) based on percentage thresholds.

Output & Termination

Display comprehensive results including total, percentage, grade, and status. End programme execution.



PROGRAMME CODE

C Implementation

```
#include <stdio.h>

int main() {
 float marks[5], total = 0, percentage;
 char grade;
 int i;

 printf("=====================================\n");
 printf(" STUDENT GRADE CALCULATOR \n");
 printf("=====================================\n");

 printf("Enter marks of 5 subjects (out of 100):\n");

for(i = 0; i < 5; i++) {
printf("Subject %d: ", i + 1);
scanf("%f", &marks[i]);

if(marks[i] < 0 || marks[i] > 100) {
printf("Invalid marks! Enter between 0-100.\n");
return 0;
}
total += marks[i];
}

percentage = total / 5;

if(percentage >= 90) grade = 'A';
else if(percentage >= 80) grade = 'B';
else if(percentage >= 70) grade = 'C';
else if(percentage >= 60) grade = 'D';
else if(percentage >= 50) grade = 'E';
else grade = 'F';

 printf("\n=====================================\n");
 printf(" RESULT \n");
 printf("=====================================\n");
 printf("Total Marks : %.2f / 500\n", total);
 printf("Percentage : %.2f %%\n", percentage);
 printf("Grade : %c\n", grade);

 if(grade == 'F')
 printf("Status : FAIL\n");
 else
 printf("Status : PASS\n");

 printf("=====================================\n");

 return 0;
}
```

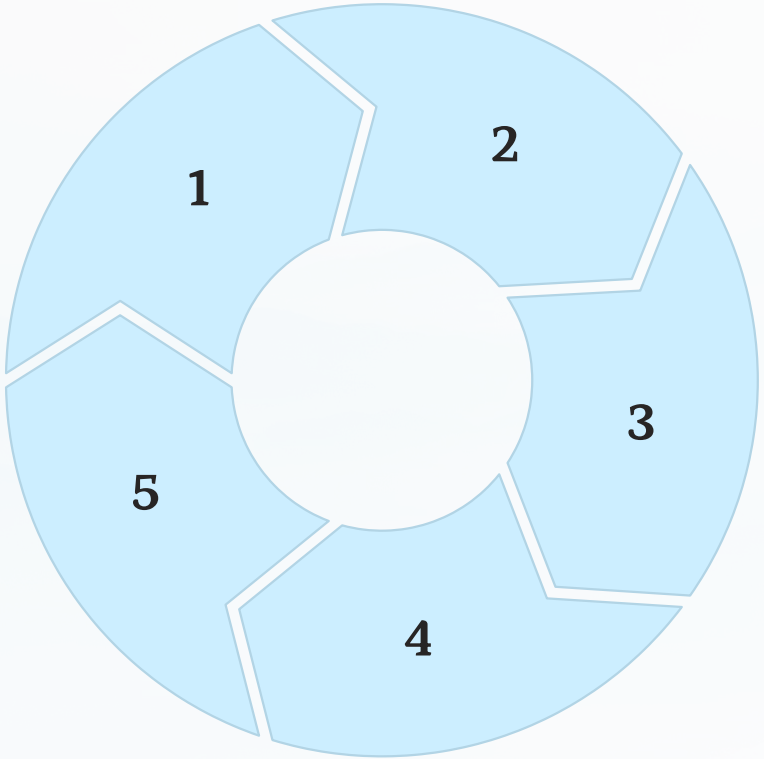
Code Explanation

Array Declaration

The `marks[5]` array efficiently stores marks for all five subjects, enabling systematic data management and manipulation throughout the programme.

Formatted Output

The `printf` statements with formatting specifiers display results clearly, using `%.2f` for two decimal precision in numerical outputs.



Input Loop

A `for` loop iterates five times, accepting marks for each subject. This approach reduces code redundancy and improves maintainability.

Validation Logic

The conditional statement `if(marks[i] < 0 || marks[i] > 100)` ensures data integrity by rejecting invalid entries immediately.

Grade Determination

Nested `if-else` statements implement the grading rubric, assigning letters A through F based on calculated percentage values.

Sample Execution

Input Values

Subject Marks Entered:

- Subject 1: **85**
- Subject 2: **78**
- Subject 3: **92**
- Subject 4: **88**
- Subject 5: **80**

Generated Output

Calculated Results:

- Total Marks: **423.00 / 500**
- Percentage: **84.60%**
- Grade: B
- Status: PASS

Advantages & Applications

Speed & Accuracy

Automated calculations eliminate manual errors and provide instant results, significantly reducing processing time from hours to seconds.

Reduced Manual Work

Minimises repetitive calculation tasks, allowing educators to focus on teaching whilst the system handles administrative grading processes.

Easy Modification

The grading thresholds and criteria can be quickly adjusted in the code, making the system adaptable to different assessment standards.

Wide Applications

Applicable in student result processing systems, academic management platforms, and institutional record-keeping databases.

KEY TAKEAWAYS

Project Summary: Grade Calculator

Automated Grading System

Developed a robust C-based programme to automate student grade calculation, minimizing manual effort and error.

Fundamental C Concepts

Successfully implemented core programming concepts: arrays for data storage, loops for input, conditionals for grading, and validation for data integrity.

Defined Grading Scale

The system accurately assigns grades based on percentage thresholds:

- A: 90-100%
- B: 80-89%
- C: 70-79%
- D: 60-69%
- E: 50-59%
- F: Below 50%

Impact & Benefits

Eliminates manual errors, significantly saves time in result processing, and provides a scalable solution for academic management.

Conclusion

> This project demonstrates the practical application of C programming fundamentals in solving real-world academic challenges through automated grade calculation.

Technical Skills

- Array manipulation
- Loop implementation
- Conditional logic
- Input validation

Practical Value

- Academic relevance
- Industry application
- Scalable solution
- Error prevention

Thank You

Questions and Discussion Welcome