
System Requirements Specification Index

For

Grade Calculator and Student Status Console Application

Version 1.0

IIHT Pvt. Ltd.
fullstack@iiht.com

TABLE OF CONTENTS

1	Project Abstract	
2	Business Requirements	
3	Error! Bookmark not defined.	
4	Template Code Structure	
5	Execution Steps to Follow	Error! Bookmark not defined.

Grade Calculator and Student Status Console

System Requirements Specification

1 PROJECT ABSTRACT

At Greenwood University's Computer Science Department, Professor Sarah Chen noticed a growing problem with inconsistent grading methods across different classes. With over 400 students and multiple professors each using their own systems for calculating grades, tracking attendance, and monitoring assignments, disputes and confusion became common. To solve this challenge, she proposed developing a Python console application that would standardize the entire evaluation process. The system uses conditional statements (if, if-else, if-elif-else) to automatically calculate grades, verify attendance requirements, and track assignment completion status for all students. This automated solution not only saves professors valuable time but also ensures that every student is evaluated using the same fair criteria. What began as a departmental solution has transformed into an essential tool that provides clear, immediate feedback to both students and faculty about academic performance

2 BUSINESS REQUIREMENTS:

Screen Name	Console input screen
Problem Statement	<ol style="list-style-type: none">1. Application must determine grade based on marks using if-elif-else2. System should check attendance eligibility using if-else3. Program should evaluate assignment completion status using if statements4. Console should display appropriate messages based on conditions5. Program should calculate final status combining all conditions

3 CONSTRAINTS

3.1 INPUT REQUIREMENTS

1. Student Marks:
 - Must be stored as integer in variable **marks**
 - Must be between 0 and 100
 - Example: 75
2. Attendance Percentage:
 - Must be stored as float in variable attendance
 - Must be between 0.0 and 100.0
 - Example: 85.5
3. Assignments Completed:
 - Must be stored as integer in variable assignments_completed
 - Must be between 0 and 10
 - Example: 8

3.2 CALCULATION CONSTRAINTS

1. Grade Calculation (if-elif-else):
 - A: marks ≥ 90
 - B: marks ≥ 80
 - C: marks ≥ 70
 - D: marks ≥ 60
 - F: marks < 60
2. Attendance Check (if-else):
 - Eligible: attendance ≥ 75.0
 - Not Eligible: attendance < 75.0
3. Assignment Status (if):
 - Complete: assignments_completed $= 10$
 - Incomplete: assignments_completed < 10

3.3 OUTPUT CONSTRAINTS

1. Display Format:

- Show grade as letter (A, B, C, D, F)
- Show attendance status as "Eligible" or "Not Eligible"
- Show assignment status as "Complete" or "Incomplete"

1. Required Output Format:

- Show "Grade: {grade}"
- Show "Attendance Status: {status}"
- Show "Assignment Status: {status}"
- Show "Final Result: {result}"

4. TEMPLATE CODE STRUCTURE:

1. Conditional Functions:

- calculate_grade() [if-elif-else]
- check_attendance() [if-else]
- check_assignments() [if]
- determine_final_status() [nested if]

2. Input Section:

- Get marks (int)
- Get attendance (float)
- Get assignments completed (int)

3. Processing Section:

- Determine grade
- Check attendance eligibility
- Verify assignment completion
- Calculate final status

4. Output Section:

- Display grade
- Show attendance status
- Show assignment status
- Display final result

5. EXECUTION STEPS TO FOLLOW:

1. Run the program
2. Enter student marks
3. Enter attendance percentage
4. Enter completed assignments count
5. View complete student status report