Program detailed design document

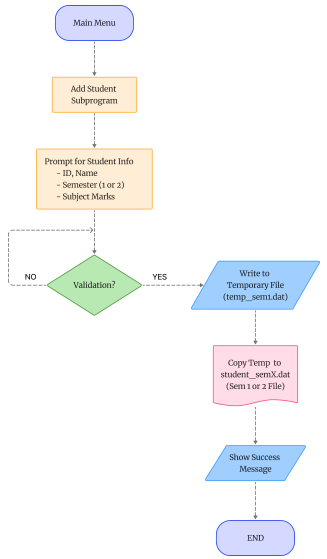
System Name	Student Management System
Screen ID	STU_IST_004
Screen Name	Student Mark Insert
Version	1.0.0
Last Updated	2025/07/24

Document Update History

Version	Date	Update Page	Item No.	Operator	Update History
1.0.0		Student Mark Insert		Nadi Linn, Khin Oo Thi Han, Khine Zin Nyut, Khine Khine Oo, Kay Zin Thant	New Creation

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Nadi Lin
	Revision Date		Revised By	
Program detailed design document	Remarks			

Functional Overview



Overview

The Add New Student Record screen allows the user to enter student information for a selected semester. Upon submission, the data is validated and then written to the appropriate semester file using a temporary file to prevent data corruption. This ensures file safety and consistency during insert operations.

Parameters

NO	Transition Source	Destination	Parameters
1	Main Menu	Add Student Record	Student ID, Name, Semester, Subject Marks

Input/Output List

NO	File ID	File Name	C	R	U	D
1	F01	student_sem1.dat		O	O	
2	F02	student_sem2.dat		O	O	
3	F03	temp_sem.dat	O	O		O

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Nadi Lin
	Revision Date		Revised By	
Console Layout Definition	Remarks			

```
**********************************
*Student Record Management System*
**********************************
1. Manage Records
2. View Reports
3. Search Record
4. Exit
Enter your choice (1-4):
1
------------------------------------------------------------
Manage Records Menu
1. Add Record
2. Edit Record
3. Delete Record
4. Back to Main Menu
Enter Manage Option (1-4):
1
Select Semester (1 or 2):
1
Enter Student ID to search:
00001
Student ID 00001      not found in Semester 1
Do you want to try again? (Y/N):
y
Enter Student ID to search:
00012
Student found: Michael Carter
Enter mark for CST11101   (0 to 100):
50
Enter mark for CST11201   (0 to 100):
```

```
50
Enter mark for CST11401   (0 to 100):
50
Enter mark for CST11501   (0 to 100):
50
Enter mark for CST11601   (0 to 100):
50
Enter mark for CST11701   (0 to 100):
50
```

```
=============STUDENT RECORD ================
ID:       00012          Name: Michael Carter

---------- MARKS ----------
CST11101        50
CST11201        50
CST11401        50
CST11501        50
CST11601        50
CST11701        50

TOTAL:    300     GRADE: C
=========================================

  CST11101-Basic Data Structures
  CST11201-Calculus I
  CST11401-Digital Fundamentals of Computer System
  CST11501-English Language Proficiency I
  CST11601-Myanmar Literature
  CST11701-Physics(Mechanics)
Marks successfully updated for student 00012
Record added successfully.
```

```
Displaying updated records...
SEMESTER I
=================================================================================================
StudentID   Name                          1101 1201 1401 1501 1601 1701  Total  Grade  Rank
=================================================================================================
00016       Quentin Harris                 98   63   90   95   59   57    462     A      1
00015       Paige Foster                   91   63   65   83   79   80    461     A      2
```

```
00024       Alena                     86   93   64   64   63   90    460     A      3
00023       David Johdan              98   94   52   98   49   68    459     A      4
00011       Laura Bennett             82   79   67   69   64   82    443     B      5
00010       Kevin Adams               69   98   53   83   64   71    438     B      6
00019       khin oo thi han           59   95   54   63   82   82    435     B      7
00018       Samuel James              95   69   64   72   73   57    430     B      8
00014       Oliver Evans              82   77   56   87   45   77    424     B      9
00021       Nadi Lin                  70   55   69   88   49   84    415     B      10
00005       Ethan Brown               96   79   70   65   47   48    405     B      11
00013       Natalie Davis             66   47   52   78   67   94    404     B      12
00006       Fiona Green               67   49   71   68   50   97    402     B      13
00022       Bay Bay                   98   57   48   54   64   80    401     B      14
00003       Charlie Lee               58   48   76   47   87   79    395     C      15
00009       Ian Gray                  50   62   58   70   81   74    395     C      16
00004       Diana Miller              71   64   53   49   78   74    389     C      17
00017       Rachel Irwin              96   55   73   59   47   55    385     C      18
00020       kay zin thant             59   61   45   80   45   89    379     C      19
00007       George White              58   61   74   52   61   68    374     C      20
00008       Hannah Black              50   56   89   50   57   63    365     C      21
00025       Nadi Lin                  56   56   56   65   65   65    363     C      22
00012       Michael Carter            50   50   50   50   50   50    300     C      23
00002       Bob Smith                 45   45   45   45   45   45    270     F      24
SUBJECT CODE EXPLANATION (SEMESTER I)
11101 - Basic Data Structures
11201 - Calculus I
11401 - Digital Fundamentals of Computer System
11501 - English Language Proficiency I
11601 - Myanmar Literature
11701 - Physics (Mechanics)
Ranked results for Semester 1 saved to ranked_results_sem1.dat
Do you want to continue? (Y/N):
```

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Khin Oo Thi Han
	Revision Date		Revised By	
Console Item Definition	Remarks			

No	Item (English)	Symbol Name	I/O	Control	Number of Digits(Number of character)		Required	Position	Data Type	Source	Default/Settings	Format	Remarks
				Control Name	Display	Input							
1	Main Menu Option	Main-Menu-Choice	I	Textbox	1	1	○	Left-justified	Numeric	User Input	-	Digit (1-4)	Choose an option:1. Manage Records,2. Reports,3. Search,4. Exit
2	Manage Records Menu	Manage-Option	I	Textbox	1	1	○	Left-justified	Numeric	User Input	-	Digit (1-4)	Choose an option: 1. Add, 2. Edit, 3. Delete, 4. Back
3	Semester Selection	WS-SEMESTER	I	Textbox	1	1	○	Left-justified	Numeric	User Input	-	Digit (1 or 2)	Semester number (1 or 2)
4	Student ID	WS-ID	I	Textbox	10	5	○	Left-justified	Alphanumeric	User Input	-	5-digit numeric	5-digit numeric ID, stored as 10 chars.
5	Display Student Name	WS-NAME	O	Textbox	30	30	-	Left-justified	Alphanumeric	Output	-	Name (Alphabets only, spaces allowed)	Maximum 15 characters
6	Marks for CST11101	WS-MARK1	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
7	Marks for CST11201	WS-MARK2	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
8	Marks for CST11401	WS-MARK3	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
9	Marks for CST11501	WS-MARK4	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
10	Marks for CST11601	WS-MARK5	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
11	Marks for CST11701	WS-MARK6	I	Textbox	3	3	○	Left-justified	Numeric	User Input	-	0–100	Integer only; no negative marks
12	Display Result Header	-	O	Display	-	-	-	Left-justified	Text	-	-	Fixed text	"=====STUDENT RECORD====="
13	Total Marks	WS-TOTAL	O	Display	5	-	-	Left-justified	Numeric (PIC 9)	System Generated	-	000–600	Displays total marks after calculation (sum of all marks).
14	Grade	WS-GRADE	O	Display	1	-	-	Left-justified	Alphanumeric	System Generated	-	A / B / C / F	Displays grade based on total marks (A, B, C,F).
15	Display Stubject Name	-	O	Display	50	-	-	Left-justified	Text	-	-	Fixed text	Explanation of subject codes
16	Continue Prompt	-	O	Display	30	-	-	Left-justified	Text	-	-	Fixed text	Prompt user to confirm action ('Y' or 'N')
17	Continue Response	WS-CHOICE	I	Textbox	1	1	○	Left-justified	Alphanumeric	User Input	-	Y/N	User inputs 'Y' or 'N' only

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Khine Khine Oo
	Revision Date		Revised By	
Program detailed design documen	Remarks			

Overview

The Insert Function is a COBOL program designed to add new student records to semester-specific data files (student_sem1.dat or student_sem2.dat).
It handles user input validation, calculates total marks and grades, and ensures data integrity during file operations.

1. Inital Prompt

GET-SEMESTER prompts for semester (1 or 2), validates input, and ensures correct file selection for data operations.

2. ID-Search

The SEARCH-STUDENT function handles student record searches by opening and reading the appropriate semester file based on the WS-SEMESTER value.

Open the files using *OPEN INPUT STUDENT-FILE-SEM1* or *OPEN INPUT STUDENT-FILE-SEM2*, and read the records in the file using *code snippets 1.7*.

3. Mark Calculations

The ACCEPT-MARKS function accepts the subject marks from the user using the *ACCEPT* keyword.
The CALCULATE-TOTAL-MARKS-AND-GRADE function adds all the subjects' marks and calculates the grade.

4. Display

Display each subject’s mark entered by the user, as well as the total mark and grade data, in the text-based console along with the subjects’ descriptions.

5. Update in files

Write the record line to the temporary file and then copy it to update the student files.

No.	Object	Event	Processing Details
		1.1 GET-SEMESTER	Prompt the user with: "Please enter semester 1 or 2." Read the input and check if it is a digit and either 1 or 2. If the input is invalid (not 1 or 2, or contains non-digit characters), display an error message and ask again.
		1.2 SEARCH-STUDENT	Open the appropriate *student_sem.dat* file and scan through all records to find the given ID. When the ID is found, display "ID found" and show the student’s name. If the ID is not found after checking all records, display "ID not found. Please try again." ***#Use code snippets 1.2***

	No.1	Screen	1.3 ACCEPT-MARKS	The system accepts marks for six subjects, requiring each input to be between 0 and 100. It validates that each mark is a non-negative numeric value. If a user enters a negative number or a value exceeding 100, the system displays an error message. The process continues sequentially, prompting for each of the six subject marks only after receiving a valid input for the previous subject. Key validation rules: Only numeric values between 0-100 are accepted Negative numbers trigger an error message Values above 100 trigger an error message The system prompts for all six subjects in sequence
			1.4 CALCULATE-MARKS-AND-GRADE	First, it initializes the total marks accumulator (*WS-TOTAL-MARKS*) to zero. Then it processes each of the six subject marks stored in the *WS-MARK-TEXT* array, converting the string values to numeric format using NUMVAL and accumulating the sum. Grading Rules: If any individual subject mark is below the passing threshold of 40, it immediately assigns a failing grade ("F"). Otherwise "A" for ≥450, "B" for ≥400, "C" for ≥300, or "F" for lower totals. The final grade is stored in the WS-GRADE variable. ***#Use code snippets 1.4***
			1.5 DISPLAY in console	The DISPLAY section presents a formatted student record with clear visual organization. It begins by outputting a header with the student's ID and name. The display then shows a detailed breakdown of all six subject marks. It dynamically displays semester-specific course names using either *SUBJECT-NAME-SEM1* or *SUBJECT-NAME-SEM2* arrays. Each course name appears alongside its corresponding mark from *WS-MARK-TEXT.* After listing individual subject results, it prominently displays the computed total marks (*WS-TOTAL-MARKS*). The final grade (*WS-GRADE*) appears in a separate section. The output concludes with a complete listing of all course names and descriptions for the relevant semester. Visual separators (==== and ----) enhance readability throughout the report. This structured format clearly distinguishes between different sections. The display ensures all academic information appears in a user-friendly manner. It effectively highlights both individual subject performance and overall results.
			1.6 FORMAT-RECORD-LINE	Transfers student ID from *WS-STUDENT-ID* to formatted variable *WS-FORMATTED-ID* Moves student name from *WS-STUDENT-NAME* to formatted variable *WS-FORMATTED-NAME* Processes each of the 6 subject marks in *WS-MARK-TEXT* array sequentially Formats each mark by replacing leading spaces with zeros using *INSPECT* Moves total marks from *WS-TOTAL-MARKS* to formatted variable *WS-FORMATTED-TOTAL* Applies zero-padding to the total marks using *INSPECT* Transfers final grade from *WS-GRADE* to formatted variable *WS-FORMATTED-GRADE* Constructs final output line by concatenating all formatted fields using STRING Combines ID, name, 6 formatted marks, formatted total, and grade into *WS-FINAL-RECORD-LINE* Creates a standardized, fixed-format record ready for file storage. ***#Use code snippets 1.5***

			1.7 UPDATE-STUDENT-RECORD	A temporary file **temp_student.dat** is used to safely perform updates. The updated student mark record is written to the temporary file. Existing records are copied to the temporary file, skipping the old record. The original file is replaced with the updated temporary file. The process applies to both semester files based on *WS-SEMESTER*. FILE-STATUS is checked after each file operation. Unchanged records are preserved exactly. An EOF flag (*WS-EOF*) controls record reading ***#Use code snippets 1.6***

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Khine Zin Nyunt
	Revision Date		Revised By	
Prompt Handling	Remarks			

Overview

1. Processing Overview

The Student Management System uses terminal prompts for all user interactions.
Each prompt follows a strict validation process before accepting input.
The system processes user responses according to field-specific rules and
provides immediate feedback for invalid inputs.

2. Prompt Processing Specifications

Main Menu Prompts

No	Prompt Text	Input Type		Validation Rules	Processing Logic
1	"Enter your choice (1-4):"	Numeric (1)		Must be 1-4	Directs to corresponding module
					1: Manage Record 2: View Reports 3: Search Record 4: Exit program

Manage Record Sub Menu

No	Prompt Text	Input Type		Validation Rules	Processing Logic

	1	"Enter Manage Option (1-4):"	Numeric (1)		Must be 1-4	Directs to corresponding module 1: Add Record 2: Edit Reports 3: Delete Record 4: Return to Main Menu
	2	"Select Semester (1 or 2): "	Numeric (1)		Must be 1 or 2	Sets WS-SEMESTER flag

Record Operations

No	Prompt Text	Input Type	Validation Rules	Processing Logic
1	"Enter Student ID to Edit:"	X(10)	Must exist in selected semester	Calls EDIT module
2	"Enter Student ID to Search: "	X(10)	Non-blank	Calls SEARCH-RECORD
3	"Enter Student ID to Delete: "	X(10)	Must exist in selected semester	Calls DELETE-RECORD

Continue Prompt

No	Prompt Text	Input Type	Validation Rules	Processing Logic
1	"Do you want to continue? (Y/N):"	X(1)	Y/y or N/n	Y: Returns to menu N: Sets exit flag (WS-NUMERIC-CHOICE=4)

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Kay Zin Thant
	Revision Date		Revised By	
Error Messages and Checking	Remarks			

Processing content

* If an error occurs in any of the following checks, the system will display an alert message and will stop further processing.
The user must correct the input before proceeding.

1. Check input format

Check that input is in the correct format.

No	What to check	Error Message	Parameters	Message ID
1	Student ID	Student ID must be exactly 5 digits		

2. Check marks input range

Ensure entered marks are within allowed range.

No	What to check	Error Message	Parameters	Message ID
1	Subject Marks	Marks must be between 0 and 100		
2	Subject Marks	Only numeric values are allowed		

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Khine Khine Oo
	Revision Date		Revised By	
Data Structures	Remarks			

Declare the files by writing ***FD STUDENT-FILE-SEM1*** and ***FD STUDENT-FILE-SEM2*** in the FILE SECTION to define the structure of the **student_sem1.dat** and **student_sem2.dat** file.

F01- student_sem1.dat

No.	Level	Variable	Data Type
1	:01	STUDENT-RECORD-SEM1	-
2	:05	STUDENT-ID1	X(10)
3	:05	STUDENT-NAME1	X(30)
4	:05	STUDENT-SUBJECTS1	-
5	:10	MARK1-CS11101	9(3)
6	:10	MARK1-CS11201	9(3)
7	:10	MARK1-CS11401	9(3)
8	:10	MARK1-CS11501	9(3)
9	:10	MARK1-CS11601	9(3)
10	:10	MARK1-CS11701	9(3)
11	:05	TOTAL-MARKS1	9(3)
12	:05	GRADE1	X(2)

F02- student_sem2.dat

No.	Level	Variable	Data Type
1	:01	STUDENT-RECORD-SEM2	-
2	:05	STUDENT-ID2	X(10)
3	:05	STUDENT-NAME2	X(30)
4	:05	STUDENT-SUBJECTS2	-
5	:10	MARK1-CS12101	9(3)
6	:10	MARK1-CS12201	9(3)
7	:10	MARK1-CS12401	9(3)
8	:10	MARK1-CS12501	9(3)
9	:10	MARK1-CS12601	9(3)

10	:10	MARK1-CS12701	9(3)
11	:05	TOTAL-MARKS2	9(3)
12	:05	GRADE2	X(2)

F03- temp_sem.dat

No.	Level	Variable	Data Type
1	:01	TEMP-STUDENT-RECORD	X(80)

Necessary Varibles for file handling, ID validation, semester control, mark processing and display

No	Level	Variable Name	Data Type	Usage
1	:01	WS-EOF	X value 'N'	End-of-file-flag
2	:88	FILE-END	Value 'Y'	Indicates file end
3	:88	FILE-NOT-END	VALUE 'N'	Indicates file not ended
4	:01	wWS-ID-VALID-FLAG	X VALUE 'N'	Flag to check if ID is valid
5	:88	ID-VALID	VALUE 'Y'	Valid ID indicator
6	:88	ID-NOT-VALID	VALUE 'N'	Invalid ID indicator
7	:01	WS-SEMESTER	9 VALUE 0	Stores selected semester(1 or 2)
8	:01	WS-STUDENT-FOUND	X VALUE 'N'	Tracks if student ID was found
9	:01	WS-STUDENt_ID	X(10)	Store Input Student ID
10	:01	WS-STUDENT-NAME	X(30)	Store Input Student Name
11	:01	WS-MARK-INPUT	9(3)	Store Input Mark
12	:01	WS-TOTAL-MARKS	9(4) VALUE 0	Holds calculateed toal marks
13	:01	WS-GRADE	X(2)	Stores computed grade
14	:01	WS-FINAL-RECORD-LINE	X(80)	Used to format final output line
15	:01	FILE-STATUS-SEM1	XX	File status for semester1
16	:01	FILE-STATUS-SEM2	XX	File status for semester2
17	:01	FILE-STATUS-TEMP	XX	File status for temporary update file
18	:01	SUBJECT-NAME-TABLE-SEM1	OCCURS 6 TIMES PIC X(10)	Holds subject codes for semester 1
19	:01	SUBJECT-NAME-TABLE-SEM2	OCCURS 6 TIMES PIC X(10)	Holds subject codes for semester 2

Student Management System	Screen ID	STU_IST_004	Screen Name	Student Mark Insert
	Event Date		Event Name	
	Create Date	2025/07/24	Author	Khine Khine Oo
	Revision Date		Revised By	
Code Snippets	Remarks			

1.1 File Control Setup

Define logical file names and link them to physical files for semester 1, semester 2, and a temporary update file.
Specify line-sequential organization and attach file status variables for error checking.

```
SELECT STUDENT-FILE-SEM1 ASSIGN TO "student_sem1.dat"
    ORGANIZATION IS LINE SEQUENTIAL
    FILE STATUS IS FILE-STATUS-SEM1.
SELECT STUDENT-FILE-SEM2 ASSIGN TO "student_sem2.dat"
    ORGANIZATION IS LINE SEQUENTIAL
    FILE STATUS IS FILE-STATUS-SEM2.
SELECT TEMP-STUDENT-FILE ASSIGN TO "temp_student.dat"
    ORGANIZATION IS LINE SEQUENTIAL
    FILE STATUS IS FILE-STATUS-TEMP.
```

1.2 Searching for Student

Open the relevant semester file (STUDENT-FILE-SEM1 or STUDENT-FILE-SEM2).
Read through the file records and check if the student ID matches WS-STUDENT-ID.
If found, set WS-STUDENT-FOUND to 'Y' and display the student's name.

```
OPEN INPUT STUDENT-FILE-SEM1 OR STUDENT-FILE-SEM2
PERFORM UNTIL FILE-END
    READ STUDENT-FILE
    IF STUDENT-ID = WS-STUDENT-ID
        MOVE 'Y' TO WS-STUDENT-FOUND
        DISPLAY STUDENT-NAME
    END-IF
END-PERFORM
CLOSE FILE
```

1.3 Mark Input and Validation

Accept marks input for each subject.
Validate that the mark is numeric and within the valid range (0-100).
Store valid marks in the WS-MARK-TEXT table.

```
PERFORM UNTIL WS-SUBJECT-INDEX > 6
   ACCEPT WS-MARK-INPUT
   IF WS-MARK-INPUT IS VALID
      MOVE WS-MARK-INPUT TO WS-MARK-TEXT(WS-SUBJECT-INDEX)
   ELSE
      DISPLAY "Invalid input"
   END-IF
END-PERFORM
```

1.4 Grade Calculation

Calculate the total of all marks.
Use EVALUATE to assign a grade based on total marks.

```
 ADD MARKS TO WS-TOTAL-MARKS
IF WS-MARK-INPUT < 40
    MOVE "F" TO WS-GRADE
ELSE
 EVALUATE TRUE
   WHEN WS-TOTAL-MARKS >= 450 MOVE "A" TO WS-GRADE
   WHEN WS-TOTAL-MARKS >= 400 MOVE "B" TO WS-GRADE
   WHEN WS-TOTAL-MARKS >= 300 MOVE "C" TO WS-GRADE
   WHEN OTHER MOVE "F" TO WS-GRADE
 END-EVALUATE
END-IF
```

1.5 Record Formatting for Outpu

Format the student record (ID, name, marks, total, grade) into a single line for output.

```
STRING STUDENT-ID DELIMITED BY SIZE
      STUDENT-NAME DELIMITED BY SIZE
      MARKS(1) TO MARKS(6) DELIMITED BY SIZE
      TOTAL DELIMITED BY SIZE
      GRADE DELIMITED BY SIZE
      INTO FINAL-RECORD-LINE
```

1.6 Updating inserted mark record

Write the updated student mark record to a temporary file and then merge it with the original file.

```
UPDATE-STUDENT-RECORD.
   OPEN OUTPUT TEMP-STUDENT-FILE
   PERFORM FORMAT-RECORD-LINE
   WRITE TEMP-STUDENT-RECORD
   CLOSE TEMP-STUDENT-FILE

   OPEN INPUT STUDENT-FILE-SEM1 OR STUDENT-FILE-SEM2
   OPEN EXTEND TEMP-STUDENT-FILE
   PERFORM UNTIL FILE-END
      READ STUDENT-FILE-SEM1 OR STUDENT-FILE-SEM2
         IF STUDENT-ID NOT = WS-STUDENT-ID
            WRITE STUDENT-RECORD
         END-IF
      END-READ
   END-PERFORM
   CLOSE FILE
   CLOSE TEMP-STUDENT-FILE
```

1.7 File Handling (Opening, Reading, Writing)

Handle file operations for reading and writing student records.

```
OPEN INPUT STUDENT-FILE-SEM1
IF FILE-STATUS-SEM1 NOT = "00"
   DISPLAY "Error opening semester 1 file. Status: " FILE-STATUS-SEM1
   STOP RUN
END-IF

PERFORM UNTIL FILE-END
   READ STUDENT-FILE-SEM1
      AT END
         SET FILE-END TO TRUE
      NOT AT END
         MOVE STUDENT-RECORD-SEM1 TO TEMP-STUDENT-RECORD
         WRITE TEMP-STUDENT-RECORD
   END-READ
END-PERFORM
CLOSE STUDENT-FILE-SEM1
```