German-Malaysian Institute

DIPLOMA PROGRAMME

LAB TASK

Academic Period : JULY 2024

Code	**SPG 0453**	**Course Name**	**Secured Object Oriented Programming**
Title	**GRADING SYSTEM**	**Examiner**	**Ainin Sofiya Hisham**
Name	**Firhan Harish Bin Mohd Faizal**	**Program/ Group**	**DSWE 3**

No	Assessment Criteria	CLO	PO	Level of Diff	Full Marks	Score	Marks	Comment
	Ability to apply required selection statement	2	4	P4	4			
	Ability to apply required looping statement	2	4	P4	4			
	Program execution: Ability to run with correct output-design logic	2	4	P4	5			
	Design of output: Scanner Class & JOptionPane components & arrangement	2	4	P4	4			
	Documentation: Comment / Description	2	4	P4	3			
	Ability to analyze problem and identify requirements (input & output)	2	4	P4	5			
	Ability to demonstrate design solution	2	4	P4	5			
				Total	**30**			

Total Marks		
Total	**PO 4**	
Signature		

LAB ASSIGNMENT: REPORT GUIDELINES

1. Introduction
2. System Pseudo-code or Flowchart to represent program algorithm
3. Briefly explain the flow of the system based on several important screenshots of the program output
4. Source Code
5. Conclusion

Grading System (10%)

Develop an interactive java program to:

1. Allow user to insert a sequence of marks based on the number of students inserted by the user earlier. The program should be able to identify the grade for each mark.
2. display the number of student who get grade A, B, C, D and F.
3. identify the number of students who failed and the number who passed
4. Display the number of pass and fail.
5. If students passed more than fail, display “Bonus to instructor” and vice versa.
6. allow user to continue or exit from the program. (Bonus Mark)

Evaluation: Report submission

Dateline: 2024 via GMi VLU

Rules : variables and object name should be unique, apply at least 3 method, comment is compulsory for each of the method, variable or object, apply looping OR selection statement

Introduction

A student grading system is a tool designed to evaluate and categorize student performance based on their academic marks or scores. Typically implemented in educational institutions, these systems automate the process of assigning grades such as A, B, C, D, or F, which correspond to various levels of achievement. By converting numerical marks into standardized grades, the system provides a clear and uniform method for assessing student outcomes.

Grading systems also help educators and institutions track student performance over time, identify trends, and make data-driven decisions regarding curriculum improvements or interventions for struggling students. In some systems, additional features might include pass/fail statistics, feedback mechanisms, and incentives for both students and educators. Overall, a student grading system simplifies the grading process, ensures fairness, and provides a comprehensive overview of academic performance.

In this lab task, I created a simple JAVA program to calculate students grade according to their marks and state whether the students passed or failed the exam. This system allows user to give input about the number of students and also being able to key in the students marks. The system then will assign grades and will state whether the students passed or failed. Lastly, the system allow user to choose to exit the system or to continue using it.

On a final note, this system is a vital role in streamlining the evaluation process within educational institutions.

Flowchart

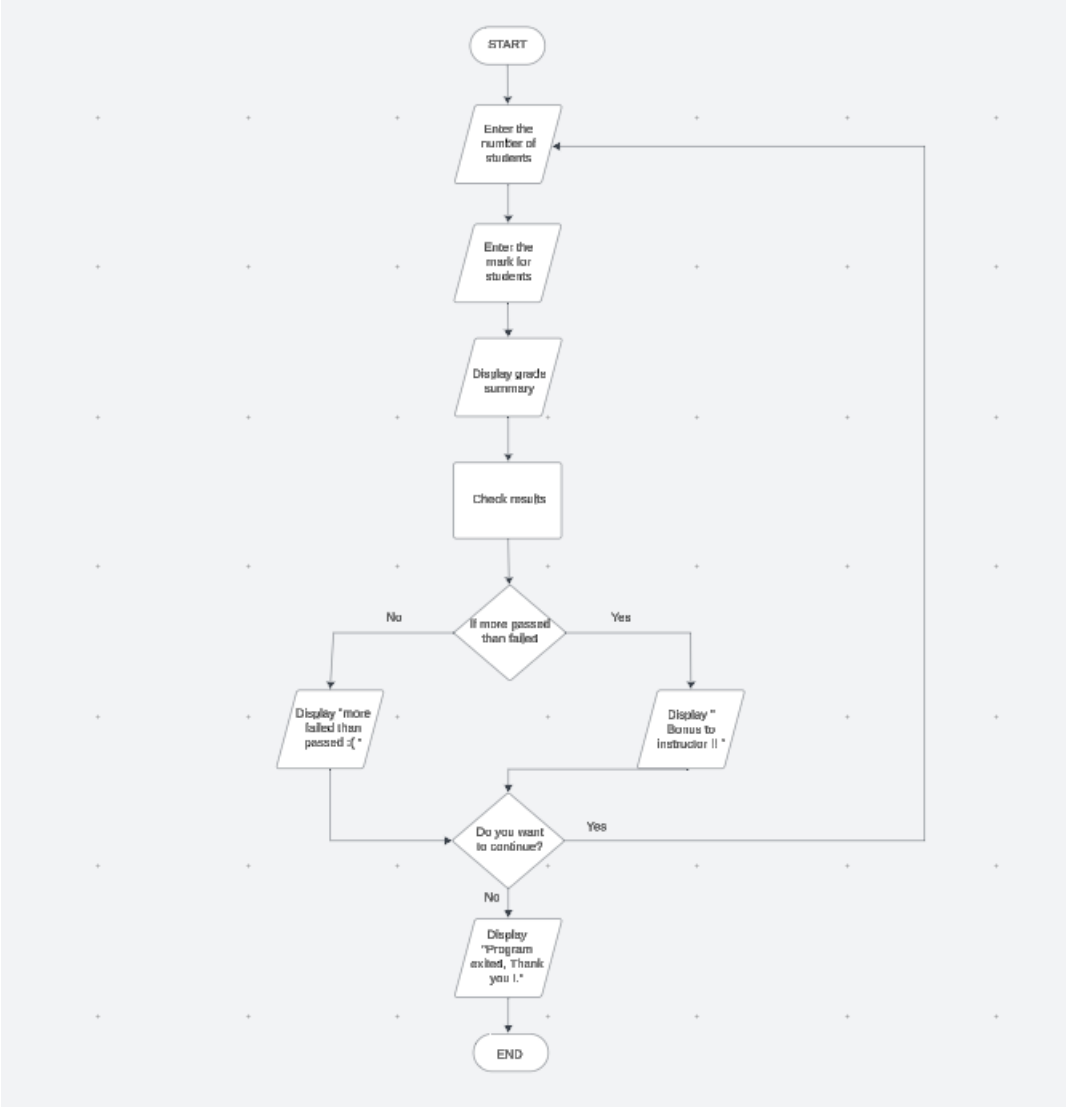

Figure 1

Program output

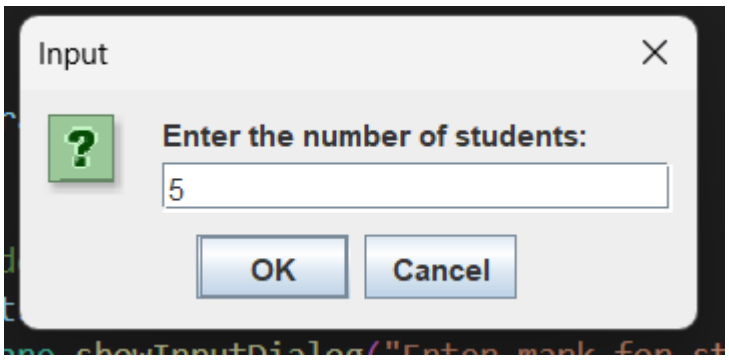

Figure 2.1

User enter the number of students which is 5.

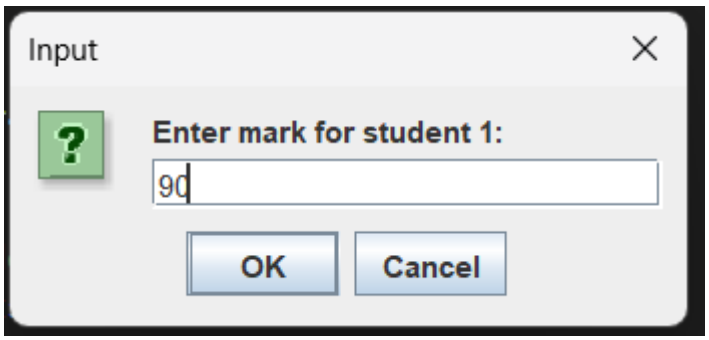

Figure 2.2

User enter the mark of student 1 which is 90.

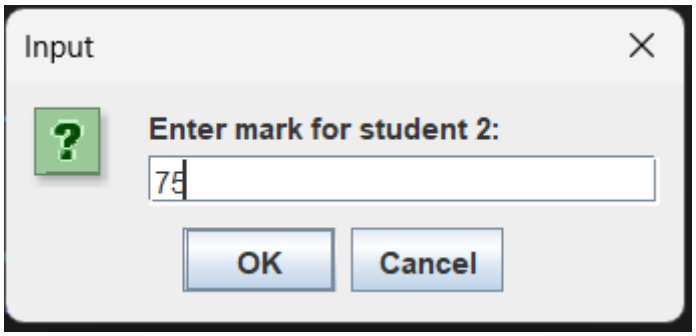

Figure 2.3

User enter the mark of student 2 which is 75.

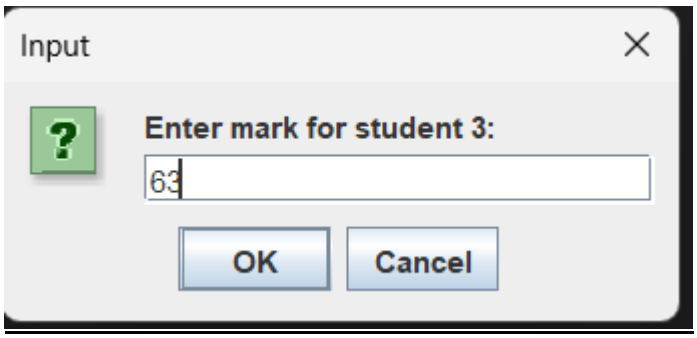

Figure 2.4

User enter the mark of student 3 which is 63.

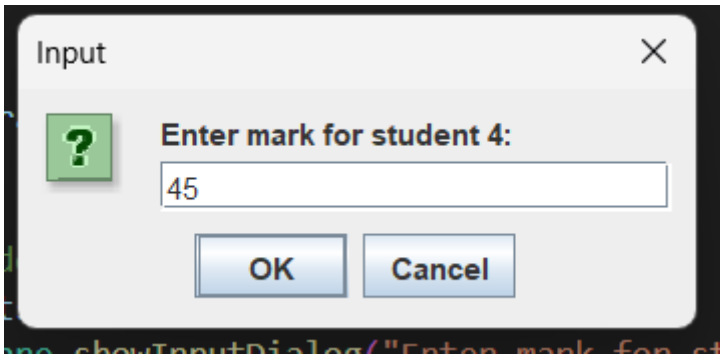

Figure 2.5

User enter the mark of student 4 which is 45.

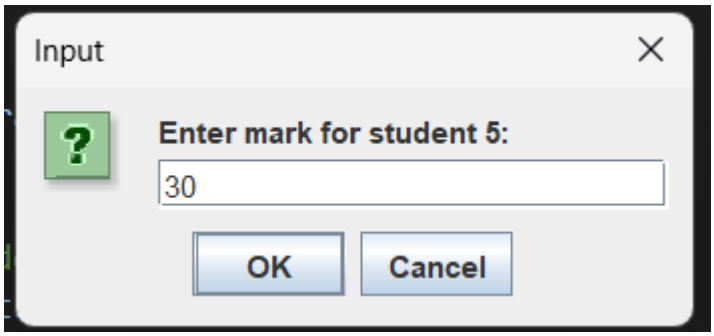

Figure 2.6

User enter the mark of student 5 which is 30.

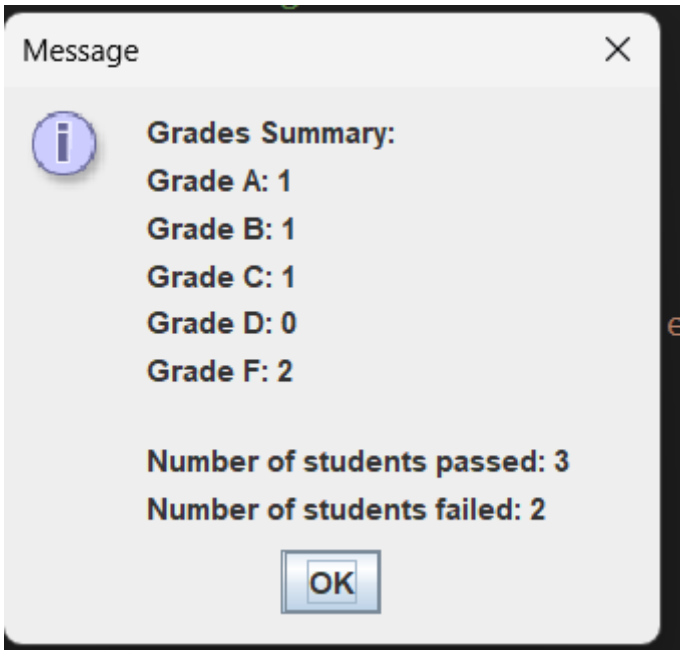

Figure 2.7

Once the user key in all the input, it will display the grades summary, number of students passed, number of students failed.

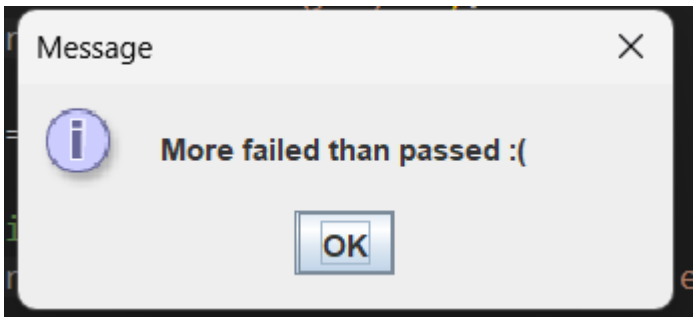

Figure 2.8

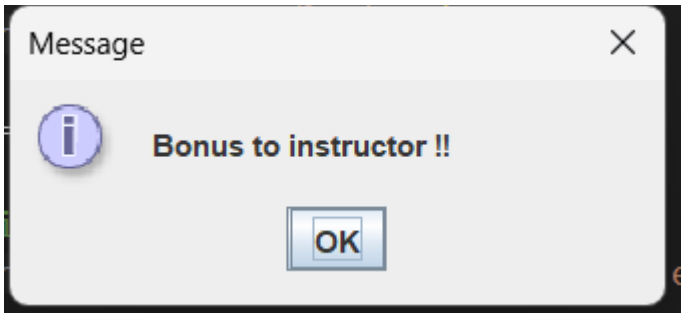

Figure 2.9

Depends whether the number of students more passed than failed, then it will print "bonus to instructor!!". Otherwise it will print "More failed than passed :("

```
Do you want to continue? (y/n): n
```

Figure 2.10

User can choose between want to continue this program or exit this program.

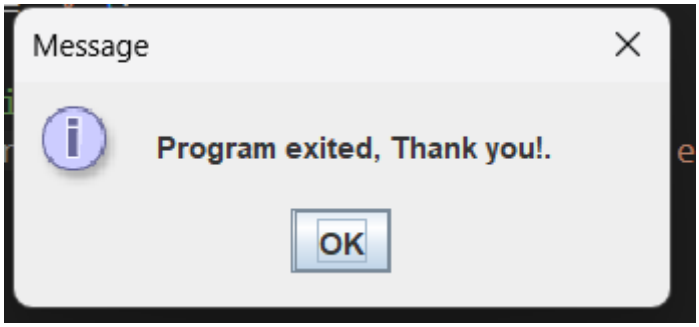

Figure 2.11

This program will display "Program exited, Thank you!." as an end to this program.
if user chose not to continue the program.

```
Do you want to continue? (y/n): y
```

Figure 2.12

If user chose to continue the program, Figure 2.1 will pop out and the system will restart .

Conclusion

Overall, the program effectively combines graphical and text-based interfaces to create a simple but functional grading system, providing clear feedback and allowing users to interact smoothly.