



CCP6114-PROGRAMMING FUNDAMENTALS

Assoc Prof Dr Junaidi Bin Abdullah

Name:

1. ZUL FADHLI BIN ZAIMAN
2. NG JUN WEI
3. SHAZARUL MUHAMMAD HAFIZ BIN SHAARI
4. WOON YU HERN

Student ID:

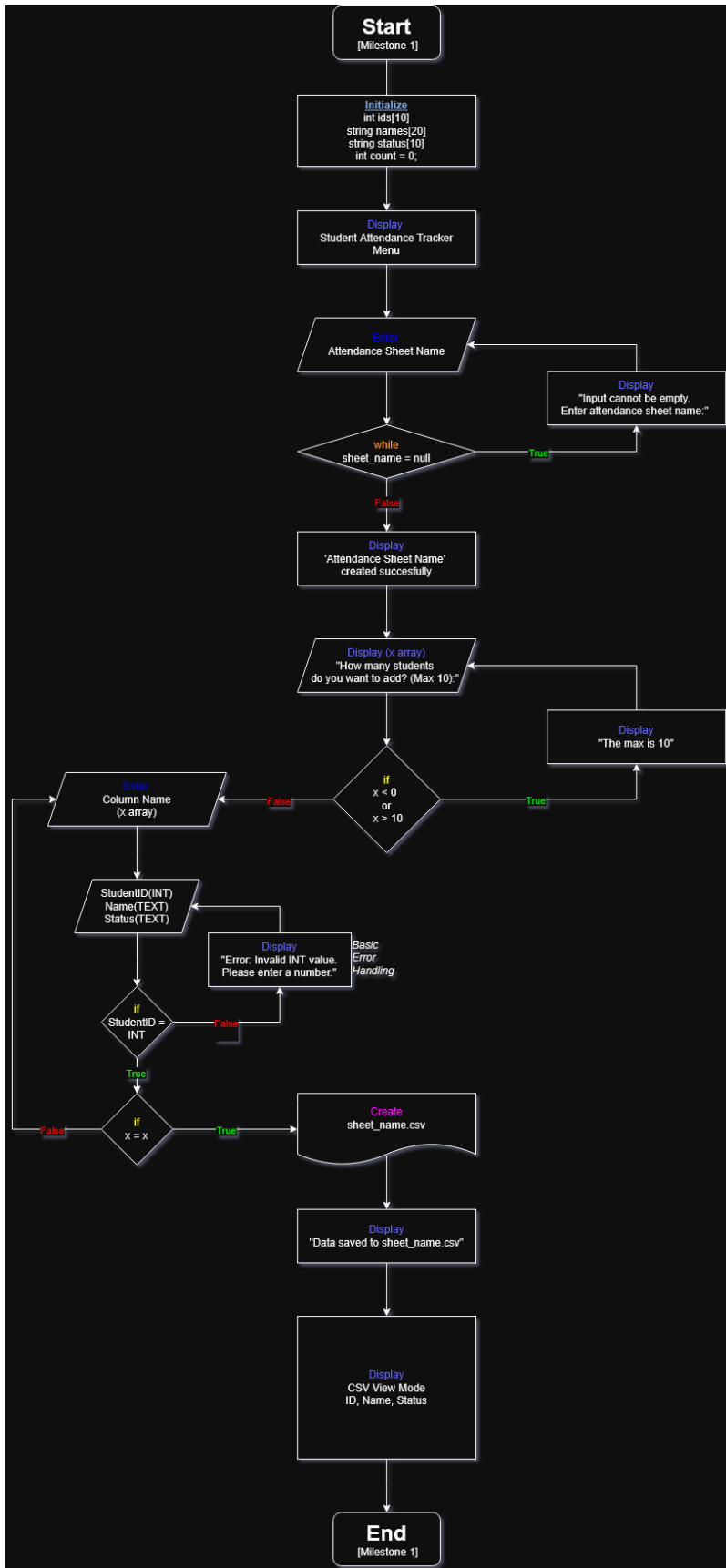
- 252UC24001
- 252UC24071
- 252UC2551C
- 252UC243K7

Table of Contents:

CCP6114-PROGRAMMING FUNDAMENTALS	1
Flowcharts	3
Pseudocode	5
Sample (Input/Output)	6
Explanation	7

Flowcharts

1.



Pseudocode

1. ``

STARTUP (main)

START

Set system encoding to UTF-8 (65001).

DISPLAY "STUDENT ATTENDANCE TRACKER - MILESTONE 1".

PROMPT user for "Attendance Sheet Name".

WHILE sheet_name is empty:

 DISPLAY "Input cannot be empty".

 RE-PROMPT for sheet_name.

SET filename = sheet_name + ".csv".

CALL writeSheet(filename).

CALL viewSheet(filename).

END

FUNCTION: writeSheet(filename)

 INITIALIZE arrays: ids (size 10), names (size 20), status (size 10).

 DO-WHILE LOOP (Get Student Count):

 PROMPT user: "How many students do you want to add? (Max 10)".

 IF input is not a number:

 Clear input error flags and ignore invalid text.

```

        Set count to 11 (to force retry).

ELSE IF count > 10:

    DISPLAY "The max is 10".

CONDITION: Repeat if count > 10 OR count <= 0.

FOR i from 0 to count - 1:

    WHILE LOOP (Get Student ID):

        PROMPT "Enter StudentID".

        IF input is a valid integer:

            EXIT while loop.

        ELSE:

            DISPLAY "Error: Invalid INT value".

            Clear and ignore invalid input.

DO-WHILE LOOP (Get Name):

    PROMPT "Enter Name".

    IF empty: DISPLAY "Input cannot be empty".

    CONDITION: Repeat if name is empty.

DO-WHILE LOOP (Get Status):

    PROMPT "Enter Status (Present/Absent)".

    IF empty: DISPLAY "Input cannot be empty".

    CONDITION: Repeat if status is empty.

```

```

CREATE and OPEN CSV file named filename.

IF file creation fails:

    DISPLAY "Error creating file".

ELSE:

    WRITE Header: "ID,Name,Status".

    FOR each student:

        WRITE ids[i], names[i], status[i] to file.

    CLOSE file.

    DISPLAY "Data saved to [filename]".

FUNCTION: viewSheet(filename)

    OPEN file named filename for reading.

    IF file open fails:

        DISPLAY "Error opening file".

    ELSE:

        DISPLAY "--- CSV VIEW MODE ---".

        READ and skip the header line.

        DISPLAY formatted table headers: "ID", "Name", "Status".

        WHILE file has data:

            READ id (until comma).

            READ name (until comma).

```

```
READ status (until newline).
```

```
IF id is not empty:
```

```
    DISPLAY data in aligned columns.
```

```
CLOSE file
```

```
END OF MILESTONE 1
```

Sample (Input/Output)

```
```text
=====
 STUDENT ATTENDANCE TRACKER - MILESTONE 1
=====
Enter attendance sheet name: Class A
How many students do you want to add? (Max 10): 2

Student 1
Enter ID (Number): 0010
Enter Name: Ali
Enter Status (Present/Absent): Present

Student 2
Enter ID (Number): 0020
Enter Name: Abu
Enter Status (Present/Absent): Absent

Data saved to Class A.csv
--- CSV VIEW MODE ---
ID,Name,Status
0010,Ali,Present
0020,Abu,Absent
```
```

Explanation

📋 Features

- * **Create Attendance Sheets:** user-defined naming for attendance files.
- * **Data Entry:** Supports input for up to **10 students** per session.
- * **Input Validation:**
 - * Prevents entering more than the maximum allowed students.
 - * Supports spaces in filenames (e.g., "Class A").
- * **Data Types:**
 - * **IDs:** Stored as strings to preserve leading zeros (e.g., ``0010`` stays ``0010``).
 - * **Names & Status:** Text-based input.
- * **File Handling:** Automatically creates and saves data to `.csv`` format.
- * **View Mode:** Reads the generated CSV file and displays the content within the console.
- * **UTF-8 Support:** optimized for Windows consoles to ensure clean text output.

🛠 Technologies Used

- * **Language:** C++
- * **Libraries:** `<iostream>`, `<fstream>`, `<string>`, `<windows.h>`

🚀 How to Run

1. **Clone the repository:**

```
```bash
git clone
[https://github.com/YourUsername/YourRepo.git] (https://github.com/YourUser
name/YourRepo.git)
```
```

2. **Compile the code:**

You can use g++ or any C++ IDE (like Visual Studio or Code::Blocks).

```
```bash
g++ TC1L_Group12_main.cpp -o attendance_tracker
```
```

```
3.  **Run the application:**
    ```bash
 ./attendance_tracker
    ```
```