GECG10069 (561085) F25: Introduction to Programming (C++)

Lab 5 : Conditional Statements

What you will learn from Lab 5

In this laboratory, you will practice using conditional statements (if, else if, and switch) correctly, and learn how to avoid common mistakes such as missing else if or break..

TASK 5-1 : IF V.S. ELSE IF

✔ Version 1 (Correct with else if)

- ➢ Conditions are mutually exclusive.
- ➢ Once one condition is true, the others are skipped.
- ➢ Clear and concise, no need to repeat conditions.

✔ Version 2 (Wrong with only if)

- ➢ Multiple conditions may be true at the same time.
- ➢ Example: input 95 prints A, B, C.
- ➢ Causes wrong or duplicated outputs.

✔ Version 3 (Correct but redundant)

- ➢ Each if repeats the full range check.
- ➢ Works correctly but is verbose and harder to maintain.

✔ Don’t forget braces!!

- ➢ In this example, each if or else if has only a single statement, so braces are optional.
- ➢ If you have multiple statements inside the block, always use braces { } to avoid logic errors and improve readability.

```
#include <iostream>
using namespace std;

int main()
{
    int ss;
    cin >> ss; //ss is int
    // Correct
    cout << "Version 1: ";
    if (ss>=90)      cout << "A" << endl; //ss≥90
    else if (ss>=80) cout << "B" << endl; //90>ss≥80
    else if (ss>=70) cout << "C" << endl; //80>ss≥70

    // Wrong
    cout << "Version 2: ";
    if (ss>=90) cout << "A" << endl; //ss≥90
    if (ss>=80) cout << "B" << endl; //90>ss≥80
    if (ss>=70) cout << "C" << endl; //80>ss≥70

    // Correct but redundant
    cout << "Version 3: ";
    if (ss>=90)          cout << "A" << endl; //ss≥90
    if (ss>=80 && ss<90) cout << "B" << endl; //90>ss≥80
    if (ss>=70 && ss<80) cout << "C" << endl; //80>ss≥70

    return 0;
}
```

TASK 5-2 : SWITCH

✔ **With break (Correct):**

- Each case ends properly, execution stops after matching case.
- Example: input 1 → only prints Select 1.

✔ **Without break (Wrong):**

- Execution “falls through” into the next cases.
- Example: input 1 → prints Select 1, Select 2, and Error.
- This is usually unintended and causes wrong outputs.

✔ **Don’t forget break!!**

- In this example, each if or else if has only a single statement, so braces are optional.
- If you have multiple statements inside the block, always use braces { } to avoid logic errors and improve readability.

```
#include <iostream>
using namespace std;

int main()
{
    int menu;
    cin >> menu;
    switch (menu) {
        // Corect
        case 1: cout << "Select 1\n"; break;
        case 2: cout << "Select 2\n"; break;
        default: cout << "Error\n";

        // Wrong
        // case 1: cout << "Select 1\n";
        // case 2: cout << "Select 2\n";
        // default: cout << "Error\n";
    }
    return 0;
}
```

EXERCISE 5-1: SORT THREE NUMBERS

Write a C++ program that reads three integers from the user, and then outputs them in non-decreasing order.

Requirements

- Your program should always print the three numbers in non-decreasing order.
- **You must not use any built-in sorting/algorithm libraries or functions** (e.g., <algorithm>, std::sort, std::stable_sort, etc.). Use only basic comparisons and swaps to sort.

Sample Input - 1
7 4 10
Sample Output - 1
4 7 10
Sample Input - 2
77 77 28
Sample Output - 2
28 77 77
Sample Input - 3
100 0 -100

Sample Output - 3
-100 0 100

EXERCISE 5-2: TAXI FARE CALCULATOR

Write a C++ program to calculate a taxi fare based on the distance.

Requirements

Input: one floating-point number (travel distance, unit = 0.1 km).

Rules:

1. **Base fare:** $70 if distance ≤ 1.5 km.
2. If distance > 1.5 km:

   For every additional 0.1 km, charge $5.

(Example: distance = 2.0 km → extra = 0.5 km →Total = $70 +$25 = $95.)

3. If distance > 20.0 km:

   After calculating the total, add a 10% surcharge (no rounding).

Output: the final fare (can be decimal).

Sample Input - 1
1.5
Sample Output - 1
70
Sample Input - 2
2.0
Sample Output - 2

95
Sample Input - 3
21.0
Sample Output - 3
1149.5