



Programming Fundamental

Lab Manual - Week 06



Introduction

Welcome to your favorite programming Lab. In this lab manual, we shall work together to learn and implement new programming concepts.

Skills to be learned:

- Write complex conditions with logical and comparison operators.
- Draw decision trees from complex real-world problems.
- Convert real-world problems with complex conditions (decision trees) into the code.

Let's do some coding.

Skill: Write complex conditions with logical and comparison operators.

Introduction

We have learned the IF Block with a single boolean expression. Now, it's time to learn how to solve complex boolean expressions that have multiple conditions. Consider the following task for better understanding.

Task 01(WP): Write a c++ program that helps the user decide if he/she can buy a certain dress or not. The conditions are that the dress must cost less than 1500 **and** it must be from MTJ brand.

IF it is an MTJ kurta and **IF** the cost of the dress is less than 1500 **Then** the user should buy that dress.

```
#include<iostream>

using namespace std;

int main()
{
    int cost;
    string brand;
    cout<< "Enter the cost of the dress: ";
    cin >> cost;
    cout<< "Enter the brand of the dress: ";
    cin >> brand;

    if (cost < 1500 && brand == "MTJ" ){
        cout<< "Buy the dress";
    }
}
```

The logical AND operator evaluates to True only if both of the conditions are True

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Notice that we have used **&& operator** in the boolean expression. This is referred to as the **Logical AND operator**. We can check multiple conditions using the && operator, however, now the body of the IF Block will be executed **only if all the conditions with && operator are true**.

Following are the different comparison operators that are used in boolean expressions.

Operator	Description	Example
>	Greater than	<pre>if (a > b)</pre>
<	Less than	<pre>if (a < b)</pre>
>=	Greater than or equal to	<pre>if (a >= b)</pre>
<=	Less than or equal to	<pre>if (a <= b)</pre>
==	if is equal to	<pre>if (a == b)</pre>

Following are the logical operators that are often used in boolean expressions.

Operator	Description	Example
AND &&	It returns True only if all conditions are True	<pre>if (a > b && a > c)</pre>
OR	It returns True if any condition is True	<pre>if (a > b a > c)</pre>
Not !	It inverts the results of the boolean expression	<pre>if (a != b)</pre>

Task 02(OP): Write a program that inputs three numbers from the user and prints the greater number. Use logical operators

Task 03(OP): Consider that Ali is a student, who recently got admission to UET. We need to write a program to congratulate him, **IF** his roll number is **501** and **IF** his name is Ali.

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Nested IF Block

An IF Condition inside an IF condition is called a Nested IF Condition.

Consider the task below:

Task 04(CL): Write a c++ program that helps the user decide if he/she can buy a certain dress or not. The conditions are that the dress must cost less than 1500 **and** it must be from MTJ brand.

```
#include<iostream>

using namespace std;

int main()
{
    int cost;
    string brand;
    cout<< "Enter the cost of the dress: ";
    cin >> cost;
    cout<< "Enter the brand of the dress: ";
    cin >> brand;

    if (cost < 1500)
    {
        if( brand == "MTJ" ){
            cout<< "Buy the dress";
        }
    }
}
```

Now the program will output only if both conditions are true

Notice that we have used the **if condition inside an if condition** in the boolean expression. This is referred to as the **Nested IF Blocks**. We can check multiple conditions using the Nested IF Blocks, however, now the body will be executed **only if all the nested IF Blocks are true**.

Task 05(CL): **IF** you have class today and **IF** your friends are going to university **Then** you will go to the university.

```
#include<iostream>

using namespace std;

int main()
{
    char going;
    char lecture;
    cout<< "Do you have class today (press Y for yes): ";
    cin >> going;
    cout<< "Are your friends going? (press Y for yes): ";
    cin >> lecture;

    if ( going == 'y' )
    {
        if( lecture == 'y' ){
            cout<< "You are also going";
        }
    }
}
```

Now the program will output only if both conditions are true

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Task 06(OP): Write a program that inputs three numbers from the user and prints the greater using the Nested IF Block

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



IF-ELSE Block

Now, the same task can be implemented with a conditional block. A conditional block consists of IF-ELSE blocks.

Task 07(CL): Consider the same task of choosing pass or fail.

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

```
int main(){
```

```
int number;
cout<< "Enter a Number: ";
cin >> number;
```

```
if(number>50)
```

```
{
cout<< "You have passed";
}
```

```
else
```

```
{
cout<< "You have failed";
}
```

```
return 0;
}
```

Only one of the blocks will be executed depending on the if condition.

If the condition in IF Block is True, else Block will not be executed.

If the condition in IF Block is False, it will be ignored and else will be executed.

Do the tasks using IF-ELSE Block

Task 08(OP): Write a program that inputs a number from the user and prints Even if it's an even number and prints Odd if it's an odd number.

Task 09(OP): Write a program that inputs two numbers from the user and prints the greater number.

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

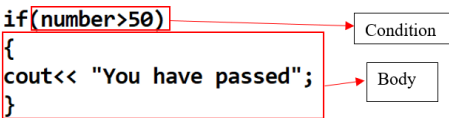
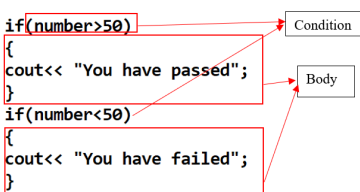
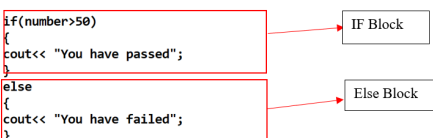
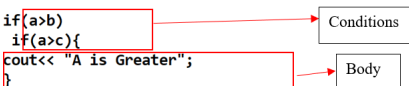
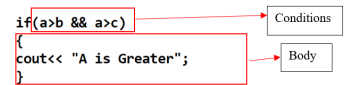
Lab Manual - Week 06



Conclusion

We can use the **comparison operators** with the **logical operators** to solve **complex boolean expressions**. In addition, there are various **conditional statements** that can be used to evaluate different conditions.

The following table highlights the difference between the different **conditional statements**.

Statement	Description
IF	<pre>if(number>50) { cout<< "You have passed"; }</pre>  <p>Executes the body only if the condition is True.</p>
Multiple IF	<pre>if(number>50) { cout<< "You have passed"; } if(number<50) { cout<< "You have failed"; }</pre>  <p>Executes only those IFs that are True.</p>
IF ELSE	<pre>if(number>50) { cout<< "You have passed"; } else { cout<< "You have failed"; }</pre>  <p>Executes IF otherwise ELSE</p>
Nested IF	<pre>if(a>b) { if(a>c){ cout<< "A is Greater"; }</pre>  <p>The body is executed only if nested conditions are True.</p>
Logical operators	<pre>if(a>b && a>c) { cout<< "A is Greater"; }</pre>  <p>The body is executed depending on the evaluation of the Logical Operator used.</p>

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



**Congratulations Students! You have just added another very important programming skill in your skillset.
Keep up the learning pace. <3**

Skill: Write complex conditions with logical and comparison operators

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Skill: Draw decision trees from complex real-world problems

Introduction

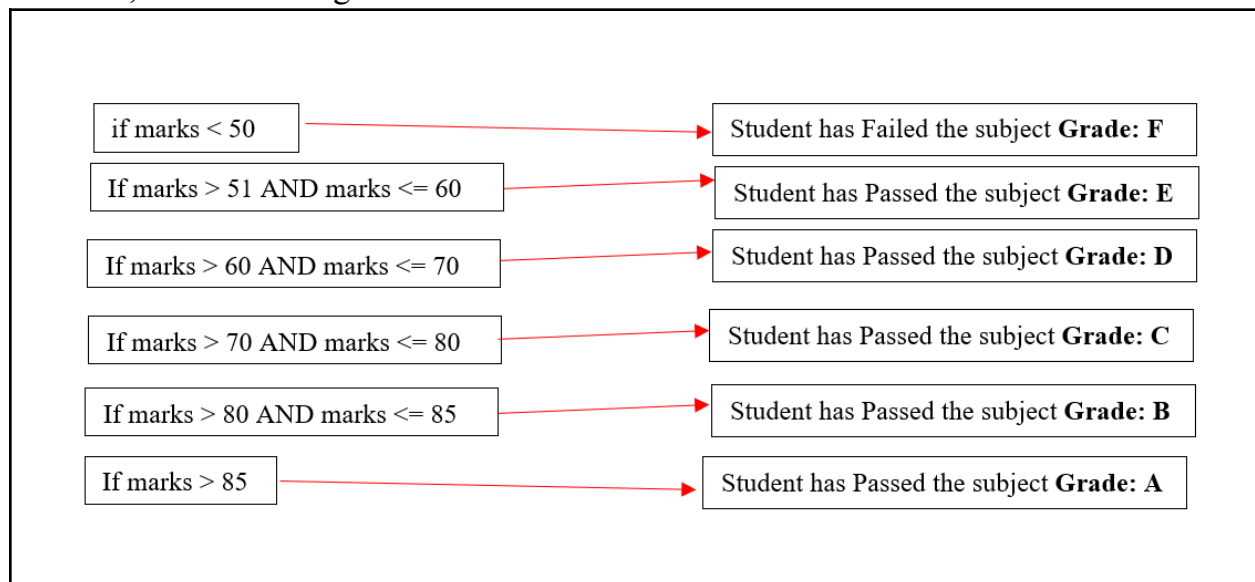
We can use the IF-ELSE Block to check **multiple conditions simultaneously**.

Consider the following task for a better understanding.

Task 01(WP): Write a c++ program that inputs the marks from the user and assigns the grade according to the following criteria.

Marks	Grade
<50	F
50-60	E
61-70	D
71-80	C
81-85	B
>85	A

Students, Based on the given information we can draw a decision tree such as follows.



Skill: Draw decision trees from complex real-world problems

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



In such a representation, we define what are the different conditions and what will happen if these conditions are satisfied respectively. However, the key takeaway is that **only one** of all these **conditional blocks** will be true and the rest will be ignored by the compiler. For example, if the user has entered 45, the first condition will be true and it should print **Grade: F** and the rest of the conditions will not be checked.

Similarly, we can draw decision trees for all the real-world problems that we face during programming. In such cases, we use **IF-ELSE IF Blocks** for solving real-world problems.

Congratulations! You have just learned how to draw decision trees for complex real-world problems.

Skill: Draw decision trees from complex real-world problems

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Skill: Convert real-world problems with complex conditions (decision trees) into the code

Lets code the above-defined decision tree using **IF-ELSE IF Blocks**

```
int main()
{
    int marks;
    cout << "Enter Obtained Marks: ";
    cin >> marks;
    cout << "Your Grade: " << gradeCalculator(marks) << endl;
}

char gradeCalculator(int marks)
{
    char grade;
    if (marks < 50)
    {
        grade = 'F';
    }
    else if (marks >= 51 && marks <= 60)
    {
        grade = 'E';
    }
    else if (marks >= 61 && marks <= 70)
    {
        grade = 'D';
    }
    else if (marks >= 71 && marks <= 80)
    {
        grade = 'C';
    }
    else if (marks >= 81 && marks <= 85)
    {
        grade = 'B';
    }
    else
    {
        grade = 'A';
    }
    return grade;
}
```

We can multiple **else if** Blocks between IF-ELSE Blocks to evaluate multiple conditions.

If none of the conditions are true then the ELSE Block is executed

Notice that we have included multiple **else if Blocks** between the **IF-ELSE Block**. Using Multiple IF-ELSE structures we can draw decision trees from complex real-world problems in an efficient manner.

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Task 01(OP):

A Store has announced to give a 10% discount on the total purchase amount every Sunday of October, and a 5% discount every other Sunday.

Write a Function that takes Day, Month, and total amount as input and returns the payable amount after the discount.

Test Cases:

Input	Output
discount("Sunday", "October", 4000);	3600
discount("Tuesday", "October", 4000);	4000
discount("Sunday", "November", 4000);	3800

Task 02(OP):

A Store has announced to give a 10% discount on the total purchase amount every Sunday of October only.

Write a Function that takes Day, Month, and total amount as input and returns the payable amount after the discount.

Test Cases:

Input	Output
discount("Sunday", "October", 4000);	3600
discount("Tuesday", "October", 4000);	4000
discount("Sunday", "March", 4000);	4000

Task 03:

A Store has announced to give a 10% discount on the total purchase amount every Sunday or Month is October.

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Write a Function that takes Day, Month, and total amount as input and returns the payable amount after the discount.

Test Cases:

Input	Output
discount("Sunday", "October", 4000);	3600

Task 04:

A Store has announced to give a 10% discount on the total purchase amount every Sunday and Month is October, March, and August.

Write a Function that takes Day, Month, and total amount as input and returns the payable amount after the discount.

Test Cases:

Input	Output
discount("Sunday", "August", 4000);	3600
discount("Tuesday", "October", 4000);	4000

Task 05:

A Store has announced to give a 10% discount on the total purchase amount every Sunday and Month is October, March, and August and a 5% discount on the total purchase amount every Monday of November and December.

Write a Function that takes Day, Month, and total amount as input and returns the payable amount after the discount.

Test Cases:

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Input	Output
discount("Sunday", "August", 4000);	3600
discount("Tuesday", "October", 4000);	4000
discount("Monday", "November", 4000);	3800

Congratulations Students! You have successfully practiced and learned conditional statements. Now, let's do some problem-solving.

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Task 01(OP):

Write a program for the following question.

Depending on age (decimal number and gender (m / f), print a personal title:

- “Mr.” – a man (gender “m”) – 16 or more years old.
- “Master” – a boy (gender “m”) under 16 years.
- “Ms.” – a woman (gender “f”) – 16 or more years old.
- “Miss” – a girl (gender “f”) under 16 years.

Input	Output	Input	Output
12 f	Miss	17 m	Mr.
Input	Output	Input	Output
25 f	Ms.	13.5 m	Master

Below is a sample function prototype that can be used to solve this problem.

string checkTitle(int, char);

Task 02(OP):

Write a program that inputs 3 numbers and prints whether they are the same (yes/no).

Input	Output
5 5 5	yes
5 4 5	no
1 2 3	no

Below is a sample function prototype that can be used to solve this problem.

bool greaterNumber(int, int, int);

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Task 03(CP):

Write a program that inputs the speed (decimal number) and prints speed information. For speed up to 10 (inclusive), print "slow". For speed over 10 and up to 50, print "average". For speed over 50 and up to 150, print "fast". For speeds over 150 and up to 1000, print "ultra-fast". For higher speed, print "extremely fast".

Input	Output
8	slow
49.5	average
126	fast
160	ultra fast
3500	extremely fast

Below is a sample function prototype that can be used to solve this problem.

string checkSpeed(float);

Task 04(CP):

In a cinema hall the chairs are ordered in a rectangle shape in r rows and c columns. There are three types of screenings with tickets of different prices:

- Premiere – a premiere screening, with a price of 12.00 EUR.
- Normal – a standard screening, with a price of 7.50 EUR.
- Discount – a screening for children and students at a reduced price – 5.00 EUR.

Write a function that enters a type of screening (string), the number of rows, and the number of columns in the hall (integer numbers) and returns the total income from tickets from a full hall.

Input	Output	Input	Output
Premiere		Normal	
10	1440.00	21	2047.50
12		13	

Below is a sample function prototype that can be used to solve this problem.

float totalIncome(string, int, int);

Task 05(CP):

A student has to travel n kilometers. He can choose between three types of transportation:

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



- Taxi. Starting fee: 0.70 EUR. Day rate: 0.79 EUR/km. Night rate: 0.90 EUR/km.
- Bus. Day / Night rate: 0.09 EUR/km. It can be used for distances of a minimum of 20 km.
- Train. Day / Night rate: 0.06 EUR/km. It can be used for distances of a minimum of 100 km.

Write a program that reads the number of kilometers n and period of the day (day or night) and calculates the price for the cheapest transport.

Input data:

number n – number of kilometers – an integer in the range of $[1 \dots 5000]$.
the word “day” or “night” – traveling during the day or during the night.

Output data:

Print the lowest price for the given number of kilometers on the console.

Input	Output	Input	Output
5 day	4.65	7 night	7
Input	Output	Input	Output
25 day	2.25	180 night	10.8

Below is a sample function prototype that can be used to solve this problem.

float lowestPrice(int, string);

Task 06(CP):

A Bulgarian entrepreneur opens small shops in a few cities with different prices for the following products:

product / city	Sofia	Plovdiv	Varna
coffee	0.50	0.40	0.45
water	0.80	0.70	0.70
beer	1.20	1.15	1.10
sweets	1.45	1.30	1.35
peanuts	1.60	1.50	1.55

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



Write a function that takes a city (string), product (string), and quantity (decimal number) and returns the cost.

Input	Output	Input	Output
coffee Varna 2	0.9	peanuts Plovdiv 1	1.5
Input	Output	Input	Output
beer Sofia 6	7.2	water Plovdiv 3	2.1

Below is a sample function prototype that can be used to solve this problem.

float checkCost(string, string, int);

Task 07(CP):

A group of football fans decided to buy tickets for the Football World Cup 2022. The tickets are sold in Qatari Rial (QR) in two price categories:

- VIP – 499.99 QR (Qatari Rial)
- Normal – 249.99 Qatari Rial (Qatari Rial)

The football fans have a shared budget, and the number of people in the group determines what percentage of the budget will be spent on transportation:

- 1 to 4 – 75% of the budget
- 5 to 9 – 60% of the budget
- 10 to 24 – 50% of the budget
- 25 to 49 – 40% of the budget
- 50 or more – 25% of the budget

Write a program that calculates whether the money left in the budget will be enough for the football fans to buy tickets in the selected category, as well as how much money they will have left or be insufficient.

Input Data

The input data is read from the console and contains exactly 3 lines:

- The first line contains the budget – real number within the range [1 000.00 ... 1 000 000.00].

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



- The second line contains the category – "VIP" or "Normal".
- The third line contains the number of people in the group – an integer within the range [1 ... 200].

Output Data

Print the following on the console as one line:

- If the budget is sufficient:
 - "Yes! You have {N} leva left." – where N is the amount of remaining money for the group.
- If the budget is NOT sufficient:
 - "Not enough money! You need {M} leva." – where M is the amount that is insufficient.

The amounts must be formatted up to the second digit after the decimal point.

Sample output:

Input	Output
1000 Normal 1	Yes! You have 0.01 QR left.
30000 VIP 49	Not enough money! You need 6499.51 QR.

Task 08(CP):

Vladimir is a student, who lives in Sofia, and goes to his hometown from time to time. He is very keen on volleyball, but is busy during **weekdays** and plays volleyball only during **weekends and on holidays**. Vladimir plays in Sofia every Saturday when **he is not working**, and **he is not traveling** to his hometown, and also during **2/3** of the holidays. He travels to his hometown **h** times a year, where he plays volleyball with his old friends on **Sunday**. Vladimir is not working **3/4** of the weekends, during which he is in Sofia. Furthermore, **during leap years Vladimir plays 15% more volleyball than usual**. We **Skill**: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore



Programming Fundamental

Lab Manual - Week 06



accept that the **year has exactly 48 weekends**, suitable for volleyball. Write a program that calculates how many times Vladimir has played volleyball throughout the year. Round the result down to the nearest whole number (e.g. 2.15 -> 2 || 9.95 -> 9).

The input data is read from the console:

- The first line contains the word “leap” (leap year) or “normal” (a normal year with 365 days).
- The second line contains the integer p – the count of holidays in the year (which are not Saturday or Sunday).
- The third line contains the integer h – the count of weekends, in which Vladimir travels to his hometown.

Input	Output	Input	Output
leap 5 2	45	normal 3 2	38

Input	Output	Input	Output
normal 11 6	44	leap 0 1	41

Good Luck and Best Wishes !!

Happy Coding ahead :)

Skill: Convert real-world problems with complex conditions (decision trees) into the code

Muhammad Irzam & Maida Shahid, Department of Computer Science, UET Lahore