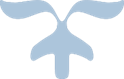
**LAB MANUAL 9**

CL1002-PROGRAMMING FUNDAMENTALS

**BSSE-FALL-2022**



**CONTROL STRUCTURES**

# If-else statement

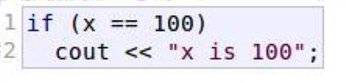
The if keyword is used to execute a statement or block, if, and only if, a condition is fulfilled.

### Syntax

|  |
| --- |
| if (boolean\_expression) {  // Executes this block if  // condition is true  }  else{  // Executes this block if  // condition is false } |

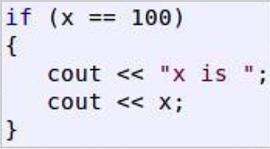
Here, the condition (boolean\_expression) is an expression that is being evaluated. If this condition is true, statement is executed. If it is false, the statement in else part is executed.

For example, the following code fragment prints the message (x is 100), only if the value stored in the x variable is indeed 100:



If x is not exactly 100, this statement is ignored, and nothing is printed.

If you want to include more than a single statement to be executed when the condition is fulfilled, these statements shall be enclosed in braces {}, forming a block:



# If-else if Statements

In C/C++ **if-else-if ladder** helps user decide from among multiple options. The C/C++ if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the else- if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. On the other hand if a sequence of ifs is used instead of elseif, the control goes to every if part even if some if-part has already been executed.

### Syntax

|  |
| --- |
| if (condition)  statement 1;  else if (condition)  statement 2;  .  .  else  statement; |

### Example:

The sample program below displays the group of person based on age:

### Program

|  |
| --- |
| int main()  {  int age;  cout<< “Enter your age: ”;  cin>> age;  if(age<13){  cout<< “You are a child”;  }  else if(age<20){  cout<< “You are a teenager”;  }  else if(age<40){  cout<< “You are still young”;  }  else {  cout<< “You are senior”;  }  return 0;  } |

# Nested If-else Statements

Sometimes, we need to use an if statement inside another if statement. This is known as nested if statement. Think of it as multiple layers of if statements. There is a first, outer if statement, and inside it is another, inner if statement.

### Syntax

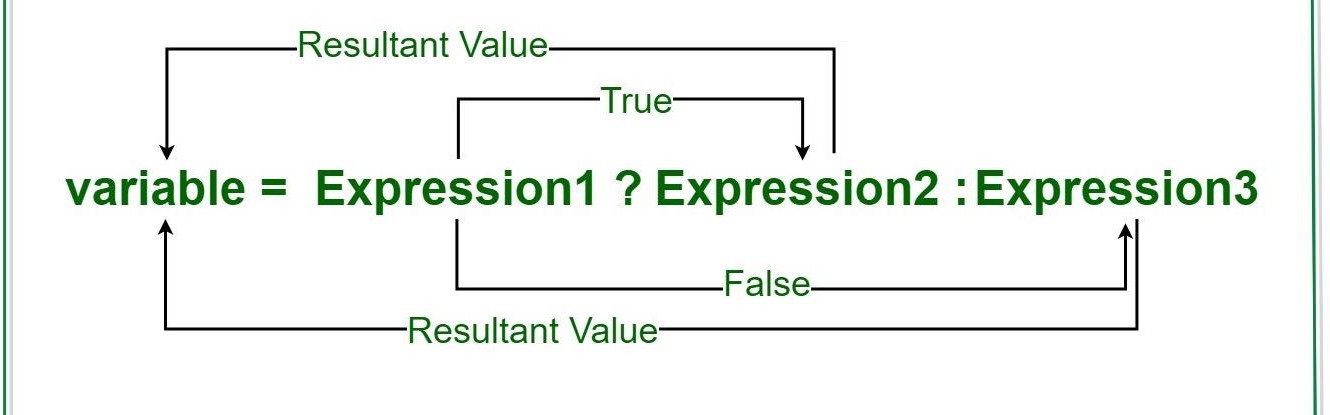
|  |
| --- |
| //Outer if statement  if (boolean\_expression){  // Executes this block if  // condition is true  //Inner if statement  if (boolean\_expression){  // Executes this block if  // condition is true  }  } |

### Program

|  |
| --- |
| int main() {  int num;  cout << "Enter an integer: ";  cin >> num;  // outer if condition  if (num != 0) {  // inner if condition  if (num > 0) {  cout << "The number is positive." << endl;}  // inner else condition  else {  cout << "The number is negative." << endl;}}  // outer else condition  else {  cout << "The number is 0 and it is neither positive nor negative." << endl;}  cout << "This line is always printed." << endl;  return 0; } |

# Conditional or Ternary Operator (?:) in C/C++

The conditional operator is kind of similar to the [if-else statement](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/) as it does follow the same algorithm as of [if-else statement](https://www.geeksforgeeks.org/decision-making-c-c-else-nested-else/) but the conditional operator takes less space and helps to write the if-else statements in the shortest way possible.



### Syntax

|  |
| --- |
| Variable = Condition ? Expression2 : Expression3 OR  Condition? Expression2 : Expression3 |

Visualization into if-else statement

|  |
| --- |
| if(Expression1)  {  variable = Expression2;  }  else  {  variable = Expression3;  } |

### Program 1

|  |
| --- |
| int n= 20;  char isEven;  isEven= n%2==0 ? ‘y’ : ‘n’;  cout<< “isEven= ”<<isEven<<endl; |

### Output:

isEven= y

### Program 2

|  |
| --- |
| int n= 20;  n%2==0 ? cout<< “n is an even number”: cout<< “n is an odd number”; |

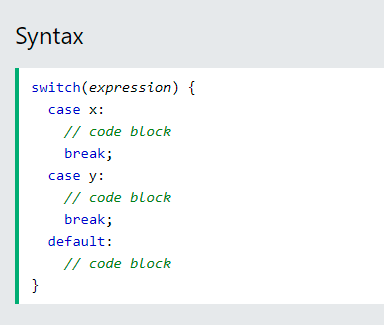
### Output:

n is an even number

# Switch Case

Switch case statements are a substitute for long if statements that compare a variable to several integral values

* The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.
* Switch is a control statement that allows a value to change control of execution.



This is how it works:

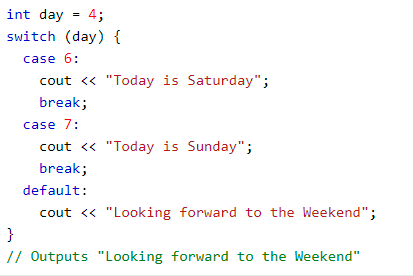
* The switch expression is evaluated once
* The value of the expression is compared with the values of each case
* If there is a match, the associated block of code is executed
* The break and default keywords are optional, and will be described later

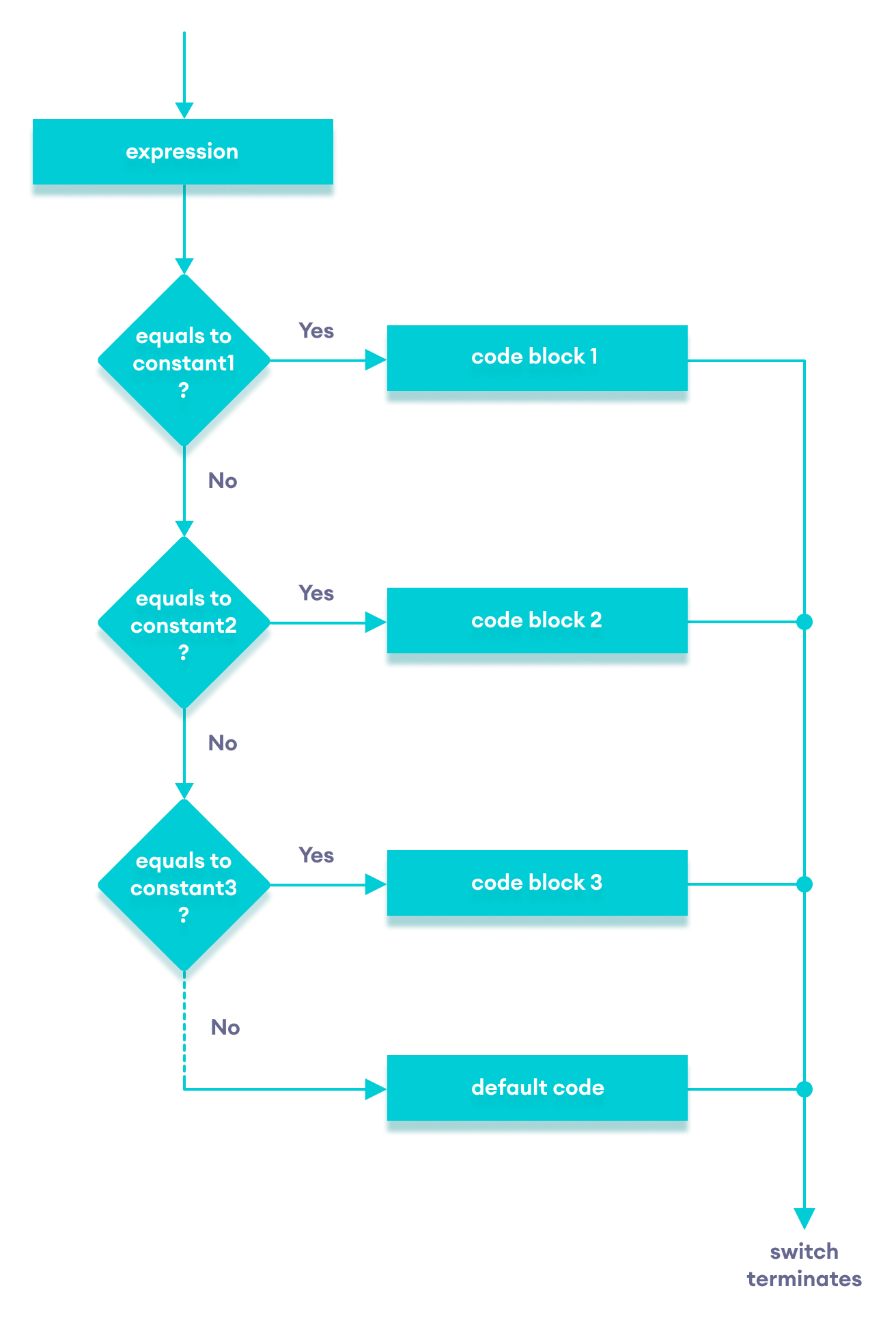
### Multiple cases can be combined if all perform similar actions

|  |
| --- |
| Switch(n){ // n can be a variable or an expression that results  // in true or false Case 1:  Case 2:  Case 3:  // code to be executed if n=1,2 or 3 break;  Case 4:  // code to be executed if n=2 break;  .  .  .  default:  // code to be executed if n doesn’t match any case above  } |

### Example 01

The example below uses the Number and will tell you if it’s a weekend or not

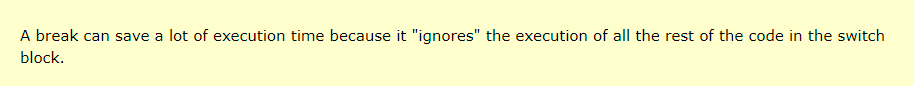




### The break Keyword

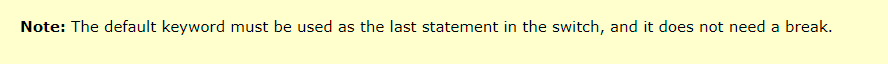
When C++ reaches a break keyword, it breaks out of the switch block. This will stop the execution of more code and case testing inside the block.

When a match is found, and the job is done, it's time for a break. There is no need for more testing.

****

### The default Keyword

The default keyword specifies some code to run if there is no case match:



**LAB TASK**

# Part A

You have been provided with a text file. You have to identify and describe errors from each part then fix that error.

# Part B

Problem solving using C++.

### Note: You have to make a Function in each task.

### Problem 1:

The date June 10, 1960 is special because when we write it in the following format, the month times the day equals the year.

**6/10/60**

Write a program that asks the user to enter a month (in numeric form), a day, and a two-digit year. The program should then determine whether the month times the day is equal to the year. If so, it should display a message saying the date is magic. Otherwise it should display a message saying the date is not magic.

### Problem 2:

A bank in your town updates its customers' accounts at the end of each month. The bank offers two types of accounts: savings and checking. Every customer must maintain a minimum balance. If a customer's balance falls below the minimum balance, there is a service charge of $10.00 for savings accounts and $25.00 for checking accounts. If the balance at the end of the month is at least the minimum balance, the account receives interest as follows:

a. Savings accounts receive 4% interest.

b. Checking accounts with balances of up to $5,000 more than the minimum balance receive 3% interest, otherwise, the interest is 5%.

Write a C++ program that prompts the user to enter account details and then display the updated balance according to above mentioned criteria. Assume that the minimum balance for the following output is $200.

### Example Input:

Enter Account Type: Savings

Enter the Account balance: $100

Output: Updated Balance is: $90

### Problem 3:

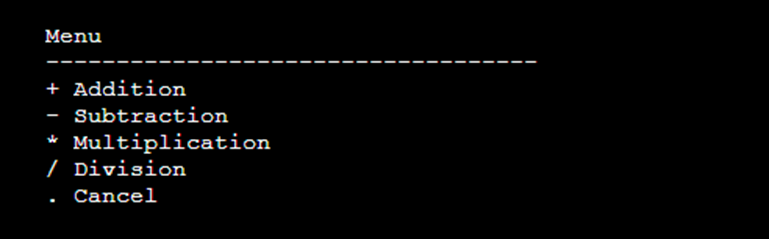
Write a program that asks the user to enter a number of seconds.

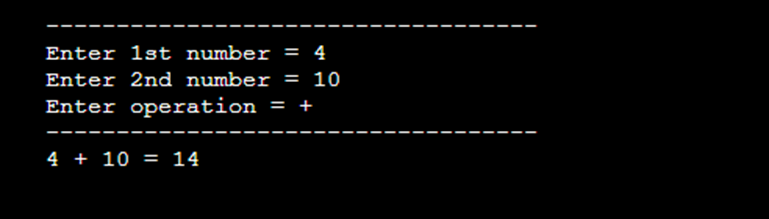
* There are 60 seconds in a minute. If the number of seconds entered by the user is greater than or equal to 60, the program should display the number of minutes in that many seconds.
* There are 3,600 seconds in an hour. If the number of seconds entered by the user is greater than or equal to 3,600, the program should display the number of hours in that many seconds.
* There are 86,400 seconds in a day. If the number of seconds entered by the user is greater than or equal to 86,400, the program should display the number of days in that many seconds.

### Problem 4:

Create a mini calculator that should perform basic arithmetic operations such as +, -, \*, / and %. The calculator should ask user to enter two numbers and the desired operation to be performed. Based on the user’s choice, perform the required operation and print the result.

Note: In case of division operation, the denominator must not be zero. Use switch case only.





|  |
| --- |
| **Submission Instructions:**   1. Save all .cpp files and screenshot with Question number e.g. Q1.cpp 2. Now create a new folder with name NAME\_ROLLNO\_LAB04 e.g. XYZ\_i21XXXX\_LAB05 3. Move all of your .cpp files to this newly created directory and compress it into .zip file. 4. Now you have to submit this zipped file on Google Classroom. |