C++ PROGRAMMING LANGUAGE

DECISION INSTRUCTIONS

Relational Operators:

- As we mentioned in the last chapter, relational operators are: >, <, >=, <=, ==, !=.
- A relational operator compares two values.
- The values can be any built-in C++ data type, such as char, int, and float
- The results of relational expressions would be of type bool (true 1 or false 0).

Example: Follow the following program and write the output.

```
#include <iostream>
using namespace std;
int main()
{
    int x, y;
    bool a, b;
    x=4<5;
    y=4>5;
    a=4<5;
    b=4>5;
    cout << x << y << a << b;
    return 0;
}
```

Output:

```
1010
Process returned 0 (0x0)   execution time : 0.060 s
Press any key to continue.
```

Example: Follow the following program and write the output.

```
#include <iostream>
using namespace std;
int main()
{
    bool a, b;
    a=true;
    b=false;
    cout << a << b;
    return 0;
}
```

Output:

```
10
Process returned 0 (0x0)   execution time : 1.639 s
Press any key to continue.
```

C++ PROGRAMMING LANGUAGE

Precedence Table

Operator type	Operators
Various	( ), postfix ++, --
Unary	!, +, –, Prefix ++, --
Arithmetic	Multiplicative *, /, %
	Additive +, –
Relational	Inequality <, >, <=, >=
	Equality ==, !=
Logical	And &&
	Or ||
Conditional	?:
Assignment	=, +=, –=, *=, /=, %=

Notes:

- **Unary operator** — Takes only one operand.
- **Binary operator** — Takes two operands.
- **Ternary operator** — Takes three operands.
- **The conditional operator** is the only ternary operator in C++.

Example: Follow the following program and write the output.

```
#include <iostream>
using namespace std;
int main()
{
    int x,y,z;
    x=3<2>1;
    y=0!=5<=0;
    z=1==1<=2;

    cout << x << y << z;
    return 0;
}
```

Output:

```
001
Process returned 0 (0x0)   execution time : 1.517 s
Press any key to continue.
```

C++ PROGRAMMING LANGUAGE

if Statement

An **if** statement consists of a boolean expression followed by one or more statements.

Syntax

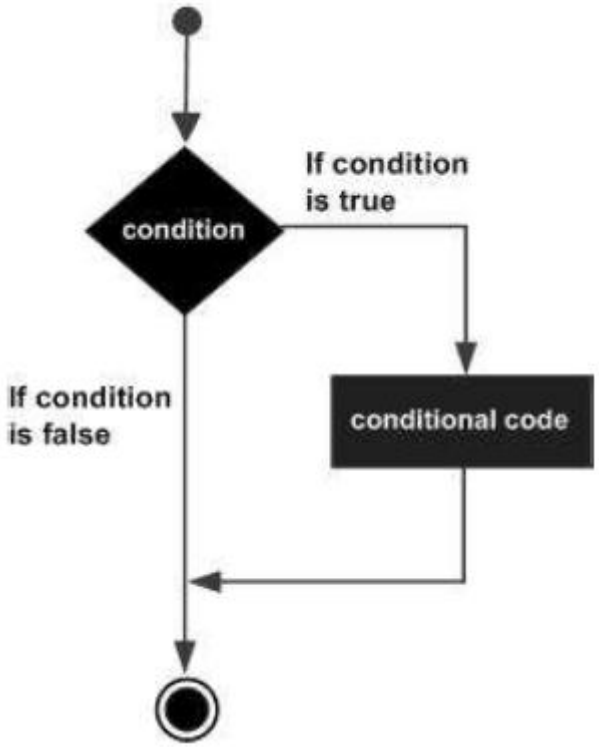
The syntax of an if statement in C++ is:

```
if(boolean_expression)
{
// statement(s) will execute if the boolean expression is true
}

Or

if(boolean_expression)
// one statement will execute if the boolean expression is true
```

If the boolean expression evaluates to **true**, then the block of code inside the if statement will be executed. If boolean expression evaluates to **false**, then the first set of code after the end of the if statement (after the closing curly brace) will be executed.

Note: There is not semicolon (;) after if statement.

Flow Diagram



C++ PROGRAMMING LANGUAGE

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100)
    {
        // if condition is true then print the following
        cout << "\nThe number is less than 100" << endl;
    }
    cout << "GO!" << endl;
    return 0;
}
```

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100)
    {
        // if condition is true then print the following
        cout << "\nThe number is less than 100" << endl;
        cout << "GO!" << endl;
    }

    return 0;
}
```

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100)
        // if condition is true then print the following
        cout << "\nThe number is less than 100" << endl;
        cout << "GO!" << endl;

    return 0;
}
```

C++ PROGRAMMING LANGUAGE

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100) // condition 1
        // if condition 1 is true then print the following
        cout << "\nThe number is less than 100" << endl;
    if(a > 100) // condition 2
         // if condition 2 is true then print the following
        cout << "\nThe number is greater than 100" << endl;

    return 0;
}
```

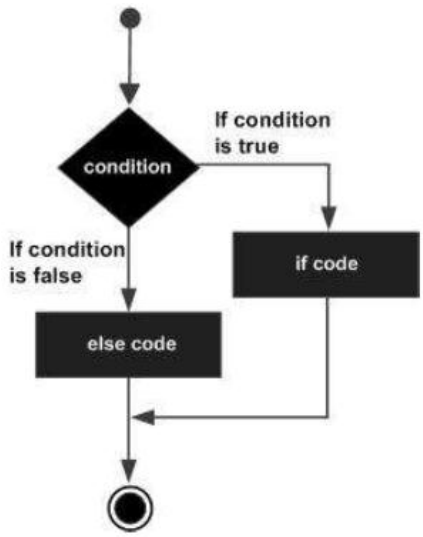
if –else Statement

Syntax

The syntax of an if-else statement in C++ is:

```
if(boolean_expression)
{
// statement(s) will execute if the boolean expression is true
}
else
{
// statement(s) will execute if the boolean expression is true
}
```

Flow Diagram



C++ PROGRAMMING LANGUAGE

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100)
        // if condition is true then print the following
        cout << "\nThe number is less than 100" << endl;
    else
        // if condition is false then print the following
        cout << "\nThe number is greater than 100" << endl;

    return 0;
}
```

if...else if...else Statement

Syntax

The syntax of an if-else if- else statement in C++ is:

```
if(boolean_expression 1)
{
// Executes when the boolean expression 1 is true
}
else if( boolean_expression 2)
{
// Executes when the boolean expression 2 is true
}
else if( boolean_expression 3)
{
// Executes when the boolean expression 3 is true
}
else
{
// executes when the none of the above condition is true.
}
```

C++ PROGRAMMING LANGUAGE

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a;
    cout<<"\nEnter a number: ";
    cin>>a;
    if(a < 100) //condition 1
        // if condition 1 is true then print the following
        cout << "\nThe number is less than 100" << endl;
    else if (a > 100) // condition 2
        // if condition 2 is true then print the following
        cout << "\nThe number is greater than 100" << endl;
    else
        // if conditions 1 & 2 are false then print the following
        cout << "\nThe number is equal to 100" << endl;

    return 0;
}
```

Nested if Statement

It is always legal to nest if-else statements, which means you can use one if or else if statement inside another if or else if tatement(s).

Syntax

The syntax for a **nested if** statement is as follows:

```
if (boolean_expression 1)
{
      // Executes when the boolean expression 1 is true
      if (boolean_expression 2)
      {
            // Executes when the boolean expression 2 is true
      }
      else
      {
            // Executes when the boolean expression 2 is false

      }
}
else
{
      // Executes when the boolean expression 1 is false

}
```

C++ PROGRAMMING LANGUAGE

Example:

```
#include <iostream>
using namespace std;
int main ()
{
    int a,b,c;
    cout<<"\nEnter 3 numbers: ";
    cin>>a>>b>>c;
```

1-

```
    if(a == b)
    {
        if(b == c)
            cout<<"a,b and c are the same!"<<endl;
        else
            cout<<"a,b and c are different!"<<endl;
    }
    else
        cout<<"a,b and c are different!"<<endl;

    return 0;
}
```

2-

```
    if(a == b)
        if(b == c)
            cout<<"a,b and c are the same!"<<endl;
        else
            cout<<"a,b and c are different!"<<endl;

    return 0;
}
```

3-

```
    if(a == b)
    {
        if(b == c)
            cout<<"a,b and c are the same!"<<endl;
    }
    else
        cout<<"a,b and c are different!"<<endl;

    return 0;
}
```

C++ PROGRAMMING LANGUAGE

Switch Statement

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case

Syntax

The syntax for a **switch** statement in C++ is as follows:

```
switch(expression)
{
      case constant-expression :
            statement(s);
            break; //optional
      case constant-expression :
            statement(s);
            break; //optional
      // you can have any number of case statements.
      Default : //Optional
            statement(s);
}
```

The following rules apply to a switch statement:

- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.

- The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.

- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.

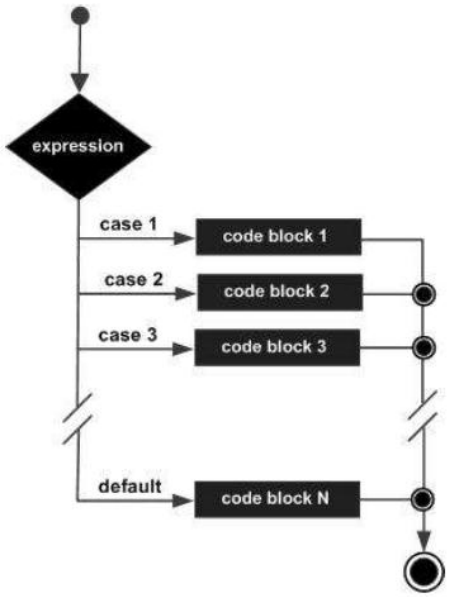
- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.

- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.

- A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

C++ PROGRAMMING LANGUAGE

Flow Diagram



Example:

```
#include <iostream>
using namespace std;
int main ()
{
    char grade;
    cout<<"\nEnter your grade:";
    cin>>grade;
    switch(grade)
    {
            case 'A' :
                cout << "Excellent!" << endl;
                break;
            case 'B' :
            case 'C' :
                cout << "Well done" << endl;
                break;
            case 'D' :
                cout << "You passed" << endl;
                break;
            case 'F' :
                cout << "Better try again" << endl;
                break;
            default :
                cout << "Invalid grade" << endl;
    }
    cout << "\nGO!" << endl;
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

Examples of the Decisions:

1- Write a program to solve a quadratic equation ($ax^2+bx+c=0$).

```
#include <iostream>
#include <cmath>    // for sqrt()
using namespace std;
int main ()
{
    float a,b,c,d,x,x1,x2;
    cout <<"enter 3 numbers: ";
    cin >> a >> b >> c;
    if (a==0)
    {
        if (b==0)
        {
            if (c==0)
                cout << "Identical case\n";
            else
                cout << "impossible case\n ";
        }
        else
        {
            x = (-c)/b;
            cout << " x = " << x;
        }
    }
    else
    {
        d = (b*b)-(4*a*c);
        if (d==0)
        {
            x1=(-b)/(2*a);
            x2=x1;
            cout << "\nx1 = x2 = " << x1;

        }
        else if (d>0)
        {
            x1=((-b)+sqrt(d)/(2*a));
            x2=((-b)-sqrt(d)/(2*a));
            cout << "\nx1 = " << x1;
            cout << "\nx2 = " << x2;
        }
        else
            cout << "\nThere is complex roots for equation\n";
    }
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

2- Write a program to calculate the age by years and months and days (assume every month = 30 days).

```
#include <iostream>
#include <cstdlib> // for exit() & system()
using namespace std;
int main ()
{
    int d1,m1,y1,d2,m2,y2,d,m,y;
    char ch1,ch2;
    cout << "Enter your birthday please (d/m/y): ";
    cin >> d1 >> ch1 >> m1 >> ch2 >> y1;
    cout << "Enter date today please (d/m/y): ";
    cin >> d2 >> ch1 >> m2 >> ch2 >> y2;
    if(d2<d1)
    {
        d2=d2+30;
        m2=m2-1;
        d=d2-d1;
    }
    else
        d=d2-d1;
    if(m2<m1)
    {
       m2=m2+12;
       y2=y2-1;
       m=m2-m1;
    }
    else
        m=m2-m1;

    if(y2<y1)
    {
        cout << "\nError!\n";
        system("pause");
        exit(1);
    }
    else
        y=y2-y1;

    cout<<"your age is: "<<y<<"years "<<m<<"months "<<d<<"days\n";
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

3- Write a program to design a simple calculator.

```
#include <iostream>
#include <cstdlib> // for exit
using namespace std;
int main ()
{
    float n1,n2,re;
    char ch;
    cout << "Enter n1 then operator then n2: ";
    cin >> n1 >> ch >> n2;
    if(ch=='+')
    {
        re=n1+n2;
    }
    else if(ch=='-')
    {
        re=n1-n2;
    }
    else if(ch=='*')
    {
        re=n1*n2;
    }
    else if(ch=='/')
    {
        re=n1/n2;
    }
    else
    {
         cout <<"\nError!! The operator (" << ch <<") is invalid operator\n";
         system("pause");
         exit(1);
    }

    cout << "= " << re << endl;
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

4- Write a program to design a simple calculator (by using switch-case).

```
#include <iostream>
#include <cstdlib> // for exit
using namespace std;
int main ()
{
    float n1,n2,re;
    char ch;
    cout << "Enter n1 then operator then n2: ";
    cin >> n1 >> ch >> n2;
    switch(ch)
    {
        case '+' : re=n1+n2;    break;
        case '-' : re=n1-n2;    break;
        case '*' : re=n1*n2;    break;
        case '/' : re=n1/n2;    break;
        default : cout <<"\nError!! The operator (" << ch <<") is invalid operator\n";
        system("pause");
        exit(1);
    }
    cout << "= " << re << endl;
    system("pause");
    return 0;
}
```

5- Write a program to calculate the sum of highest two grades of three grades.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
    int a,b,c,sum;
    cout<<"\nEnter 3 grades: ";
    cin>> a >> b >> c;
    if(a>b)
    {
        if(b>c)
            sum=a+b;
        else
            sum=a+c;
    }
    else
    {
        if(a<c)
            sum=b+c;
        else
            sum=a+b;
    }
    cout <<"sum = " << sum << endl;
    system("pause");
    return 0; }
```

C++ PROGRAMMING LANGUAGE

Logical OR Operator (||)

It is binary operator (takes two operands), if both the operands are non-zero, then condition becomes true.

A	b	a||b
0	0	0
0	1	1
1	0	1
1	1	1

Examples: Write a program to read student grade and then:

1- Print the message (DATA ERROR!!), if 0 > grade > 100.

2- Print the message (PASS), if grade ≥ 50.

3- Print the message (FAIL), if grade < 50.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
    int grade;
    cout<<"\nEnter your grade:";
    cin>>grade;
    if(grade>100||grade<0)
        cout << "\nDATA ERROR!!"<< endl;
    else if(grade>=50)
        cout << "\nPASS" << endl;
    else
        cout << "\nFAIL" << endl;
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

Logical AND Operator (&&)

It is binary operator (takes two operands), If both the operands are non-zero, then condition becomes true.

A	b	a&&b
0	0	0
0	1	0
1	0	0
1	1	1

Examples: Write a program to calculate the tax as:

1- Tax =0, when income≤400.

2- Tax =15% of income, when 400<income<800.

3- Tax =20% of income, when income≥800.

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
    float income, tax;
    cout<<"\nPlease enter your income:";
    cin>>income;
    if(income<=400)
        tax=0;
    else if(income>400 && income<800)
        tax=0.15*income;
    else
        tax=0.2*income;
    cout << "\nThe tax= " << tax <<endl;
    system("pause");
    return 0;
}
```

C++ PROGRAMMING LANGUAGE

The conditional operator

It is the only ternary operator in C++ (Takes three operands).

Syntax:

Condition? a : b	If Condition is true, then it returns value of a otherwise returns value of b.

Examples: What do the following program prints:

```
#include <iostream>
#include <cstdlib>
using namespace std;
int main ()
{
    float x,y;
    cout<<"\nEnter a number:";
    cin>>x;
    y=x>=50?1:2;
    cout << "\ny= " << y <<endl;
    system("pause");
    return 0;
}
```