Selection Structure

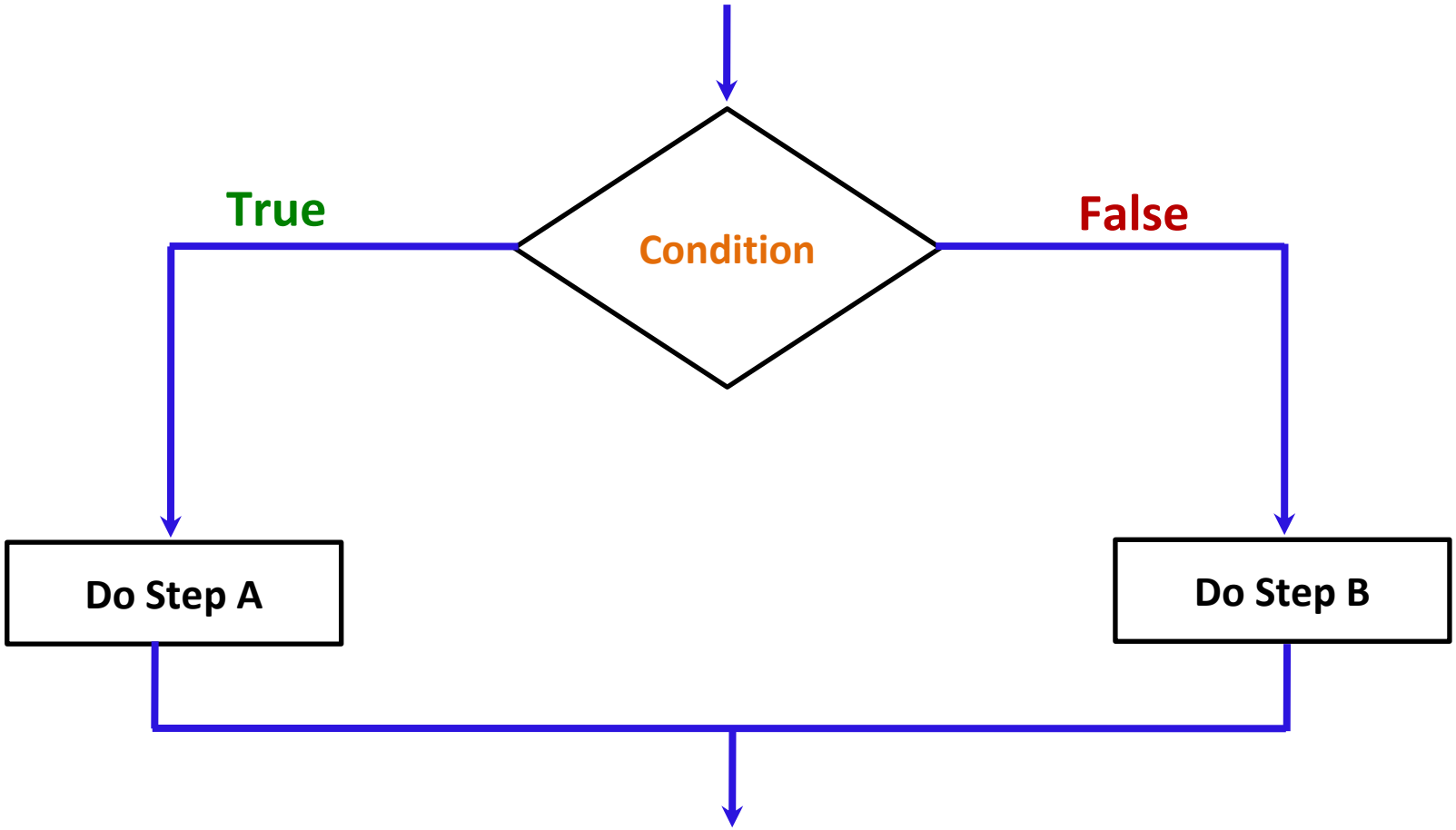
(CS 1002)

Dr. Mudassar Aslam

Cybersecurity Department

National University of Computer & Emerging Sciences,
Islamabad Campus

Selection Structures

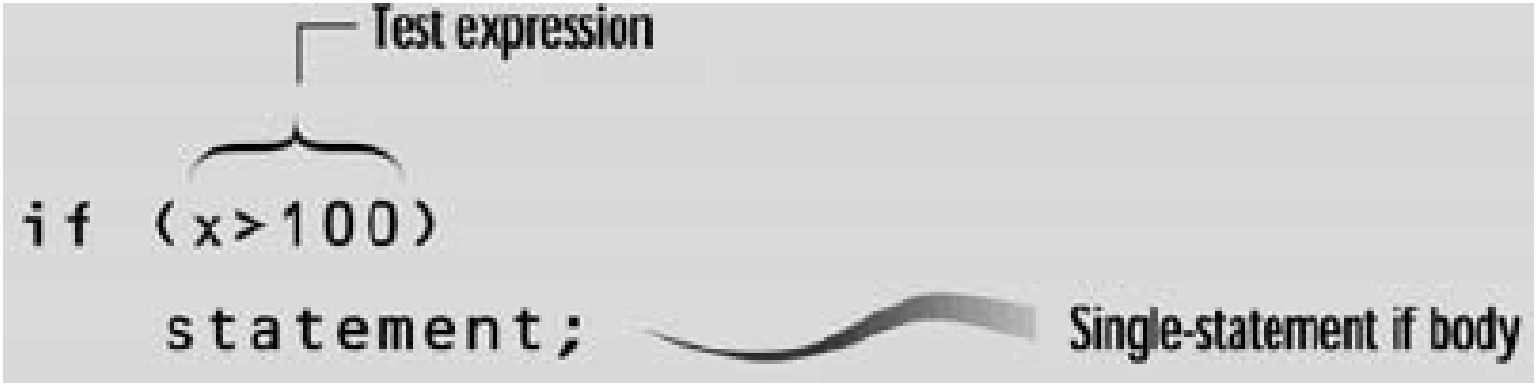


`if` Statement

- **Enables programmer** to make **decisions as to which statements** are **executed**, and which are **skipped**.

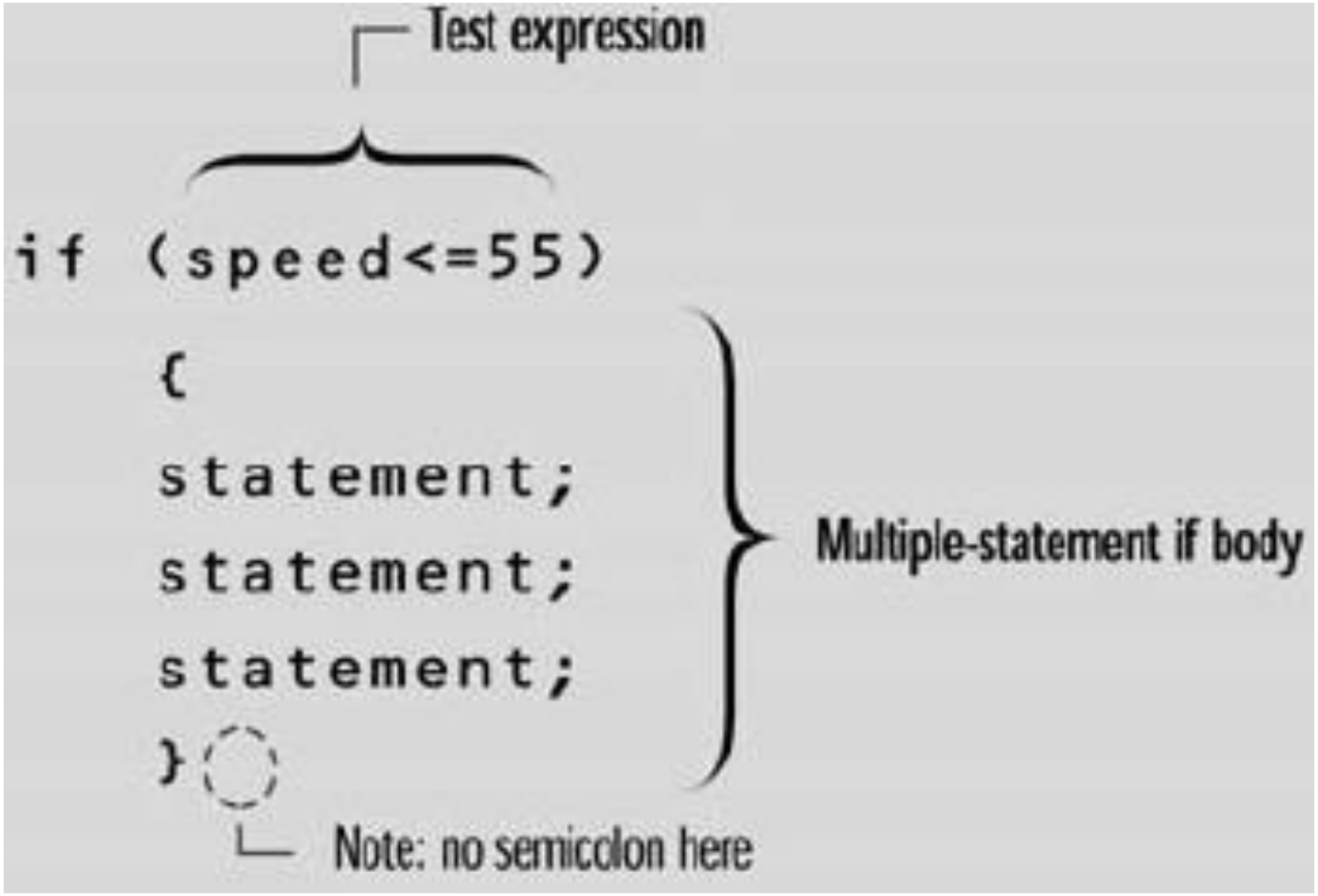
If statement (One Way)

- Syntax of if with **single-statement** body



If statement (One Way)

- Syntax of if with **multiple-statements** body



Examples…. (if, one-way)

- Write a program to calculate tax collection according to the following formula:
  - 5% Tax, if salary is above 50000
  - 3% Tax , if salary is between 30000 and 50000
  - 2% Tax, is salary is less than 30000

In the end the program should print the calculated tax.

Relational Operators

- Used to **compare values** to determine **relative order**
- **Operators:**

>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	Equal to
!=	Not equal to

Relational Expressions

- **Relational expressions** are **Boolean** (*i.e.*, evaluate to **`true`** or **`false`**)
- Examples:

  `12 > 5` is **`true`**

  `7 <= 5` is **`false`**

  if **`x`** is 10, then

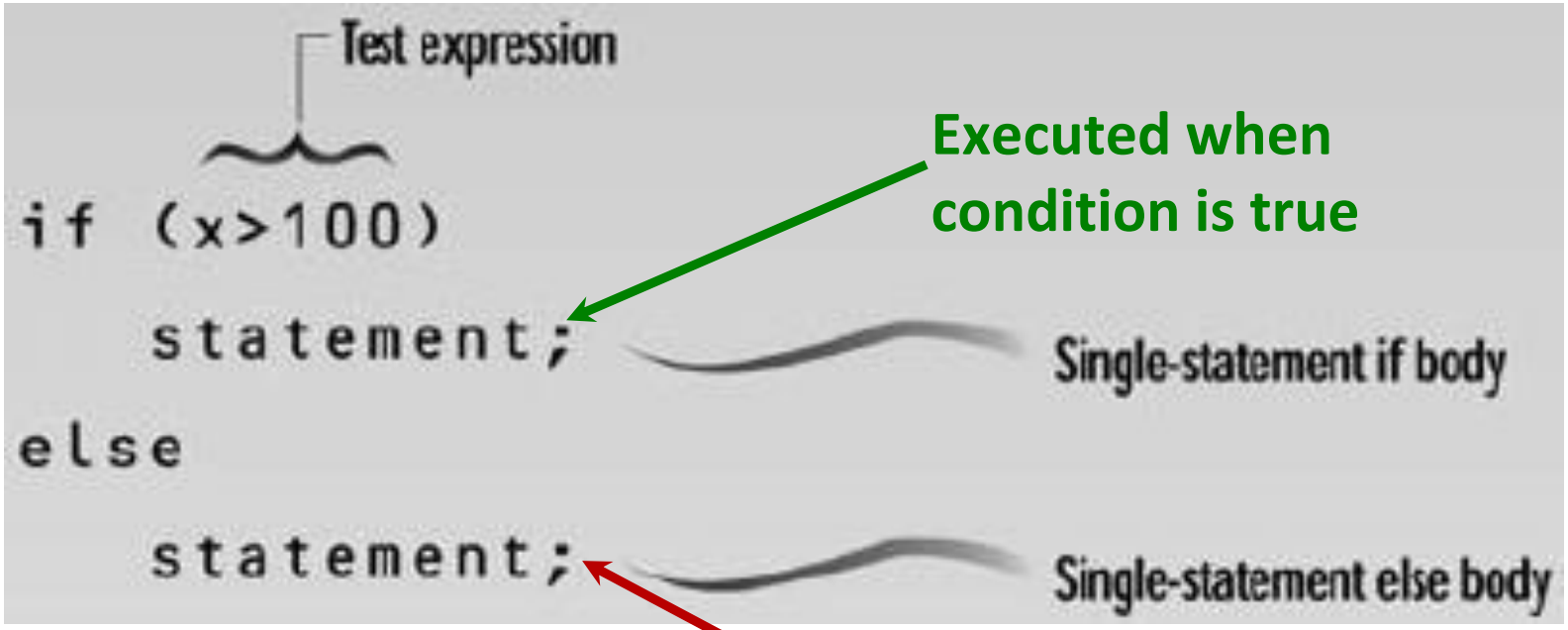
  `x == 10` is **`true`**,

  `x != 8` is **`true`**, and

  `x == 8` is **`false`**

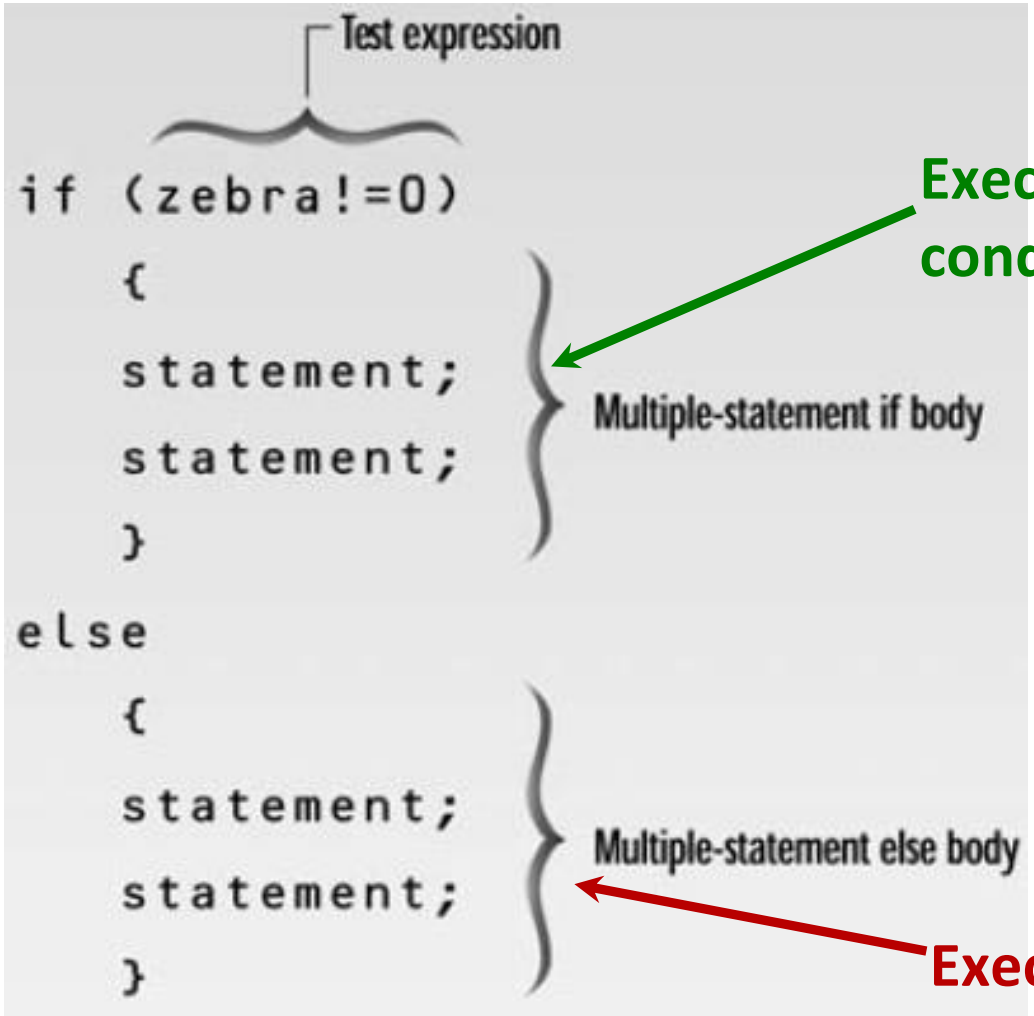
If statement (Two-Way)

- Syntax of if with **single-statement** body



If statement (Two-Way)

- Syntax of if with **single-statement** body



Examples…. (if, Two-way)

1. Write a program that squares a number (entered by user), if it is between 10 and 100. For all other numbers, an Error message is shown and program terminates.

Examples…. (if, Two-way)

2. Write a payroll program using following rules:
   - Hourly rate: 100 (rupees)
   - If an employee works 40 hours or fewer, he is paid regular pay (hourly rate * number of hours worked).
   - If employees work more than 40 hours, then he WILL receive two times his hourly rate for those hours over 40, in addition to his regular pay for the first 40.

Nested “if...else” Statements

- **Purpose**: **To test** more than one ***factors*** before we write our executable code

- By **nesting** if **structures**, we write **ONE COMPLETE** if structure inside a **SINGLE BRANCH** of a parent if structure

Nested “if...else” Statements

```
if (marks>90)
{
     cout<<"\nYou got A grade";
     if(nAvailable_scholorships>0)
     {
          cout<<"\nYou got scholorship too";
          tution_fee_due = 0;
     }
}
else
{
     if (marks>=50)
          cout<<"\nYou passed the course";
     else
          cout<<"\nYou failed the course";
}
```

Matching the "else"

```
#include <iostream>
using namespace std;
int main()
{
    int a, b, c;
    cout<<"Enter three numbers, a, b, and c:\n";
    cin >> a >> b >> c;
    if( a==b )
    {
        if( b==c )
            cout << "a, b, and c are the same\n";
    }
    else
        cout<< "a and b are different\n";
    return 0;
}
```

The else...if Construction

```
if (marks>=80)
{
        cout<<"\n You got A grade";
        cout<<"\n You won scholarship too";
}
else if (marks>=70)
        cout<<"\n You got B grade";
else if (marks>=60)
        cout<<"\n You got C grade";
else if (marks>=50)
        cout<<"\n You got D grade";
else
        cout<<"\n You are fail";
```

Compound Conditions

COMPOUND: Multiple conditions

```
if ( (Age < 0) || (Age > 120) )
```

- At least one condition must be true

```
if ( (Age >=1) && (Age <= 120) )
```

- BOTH conditions must be true

INVALID:

```
if ( Age >=1 && <= 120 ) //Need 2 relational expressions
if ( 1 <= Age <= 120 )  //Although okay in math!
```

Logical Operators

Used to **create relational expressions** from other **relational expressions**

&&	AND	New relational expression is true if both expressions are true
||	OR	New relational expression is true if either expression is true
!	NOT	Reverses the value of an expression; true expression becomes false, false expression becomes true

Logical Operator Rules

Operand(s) must be bool(true/false)

true && true true || true true || false false || true ! false	true
true && false false && true false || false ! true	false

Logical Operator Examples

```
int x = 12, y = 5, z = -4;
```

.

`(x > y) && (y > z)`	`true`
`(x > y) && (z > y)`	`false`
`(x <= z) || (y == z)`	`false`
`(x <= z) || (y != z)`	`true`
`!(x >= z)`	`false`

Logical Precedence

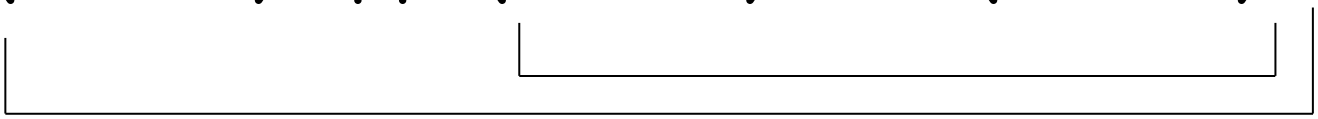
Highest `!`

`&&`

Lowest `||`

Example:

```
(2 < 3) || (5 > 6) && (7 > 8)
```



is true because AND is evaluated before OR

More on Precedence

Highest	arithmetic operators
↓	relational operators
Lowest	logical operators

Example:

8 < 2 + 7 || 5 == 6 is true

switch statement

- Provides a series of **alternatives** (selections) based on the **value of a SINGLE variable**
- **Replaces** a **series** of **chained if-else statements**
- **Makes code easier to read**

Switch - Syntax

```
switch (Variable)
{
      case <value_1> :   statement1;
                         statement2;
                         statement3;
                         break;

      case <value_2> :   statement1;
                         statement2;
                         break;

      case <value_3> :   statement1;
                         break;

      default:           statement1;
                         statement2;
}
```

Optional

switch statement (without break)

Suppose ch is 'a'

```
switch (ch) {
  case 'a': cout <<" ch contains  a";
  case 'b': cout <<" ch contains  b";
  case 'c': cout <<" ch contains  c";
}
```

switch statement (with break)

Suppose ch is ‘b’

```
switch (ch) {
  case 'a': cout <<“ ch contains  a”; break;
  case 'b': cout <<“ ch contains  b”; break;
  case 'c': cout <<“ ch contains  c”; break;
}
cout<<“\n End of program…”;
```

Switch – Example-1

```
char grade;
cin>>grade;

switch (grade)
{
      case 'A': tution_fees *= 0.20;
                break;

      case 'B': tution_fees *= 0.40;
                break;

      case 'C': tution_fees *= 0.60;
                break;

      default:  tution_fees *= 1;
}
```

Switch – Example-2

```
int  day;
cout<<"Enter day number";   cin>>day;

switch (day)
{
      case 1 :
      case 7 : cout << "This is a weekend day";
               break;
      case 2 :
      case 3 :
      case 4 :
      case 5 :
      case 6 : cout << "This is a weekday"; break;
      default : cout << "Not a legal day";
}
```

Switch – Example-3

```
int  n1, n2; char key;
cout<<"\nEnter numbers: ";
cin>>n1>>n2;
cout<<"Enter an arithmetic operator";
cin>> key;

switch (key)
{
     case '+'   :    cout<<(a+b);    break;
     case '-'   :    cout<<(a-b);    break;
     case '*'   :    cout<<(a*b);    break;
     case '/'   :    cout<<(a/b);    break;
     default:        cout<<"Error: Invalid key…";
 }
```

Nested Switch

- switch can also be nested…
- Another switch part of the case component
- Example…

Example: Nested switch

```
int x = 1, y = 2;

// Outer Switch
switch (x) {

    // If x == 1
    case 1:
            // Nested Switch
            switch (y) {

                    // If y == 2
                    case 2:
                            cout << "Choice is x=1 and y=2\n";
                            break;

                    // If y == 3
                    case 3:
                            cout << "Choice is x=1 and y=3\n";
                            break;
            }//end of nested switch
            break;

    // If x == 4
    case 4:
            cout << "Choice is x=4\n";
            break;

    // If x == 5
    case 5:
            cout << "Choice is x=5\n";
            break;

    default:
            cout << "Choice of x is other than 1, 2 3, 4, or 5";
            break;
}//end of outer switch
```

Ternary/Conditional Operator

- The **conditional operator (? :)** is a **ternary operator** (***three operands***), is used to **simplify** an **if/else statement**.

```
expression1 ? expression2 : expression3;
```

- If **expression1** is **true**, the **result** of the ***conditional expression*** is ***expression2***. Otherwise, the ***result*** is ***expression3***

Conditional Operator (Ternary operator)

```
if (x > 0)
     y = 1;
else
     y = -1;
```

is equivalent to

```
y = (x > 0) ? 1 : -1;
```

Syntax:

result = (condition) ? Expression1 : Expression2;

Conditional Operator, examples

```
cout << ((num % 2 == 0) ? "num is even" : "num is odd");
```

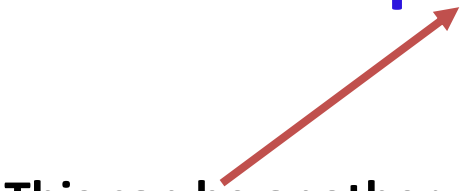
```
int min_value = (num1<num2) ? num1 : num2;
```

```
unsigned int absvalue = (n< 0) ? -n : n;
```

Nested Ternary Operator

- Just like if…, ternary operator can be nested too
- Syntax:

result = (condition) ? Expression1 : Expression2;



Examples…

```
a ? b : c;
```

?

```
if ( a )
    b;
else
    c;
```

Examples…

```
//conditions are in red
a ? b : c ? d : e ? f : g ? h : i;
```

?

```
if (a)
      b;
else if (c)
      d;
else if (e)
      f;
else if (g)
      h;
else
     i;
```

Nesting can be in any part of the operator

```
cout << "Execute expression using "
<< "ternary operator: ";

//Nesting ternary in the second part of the operator....
int a = (2 > 3) ? 2 : (3 > 4) ? 3 : 4;
cout << a << endl;

//Nesting ternary in the first part of the operator....
a = (4 > 3) ? (6 > 7)? 6 : 9 : 5;
cout << a << endl;


cout << "Execute expression using "
<< "if else statement: ";
if ( 2 > 3 )
    cout << "2";
else if ( 3 > 4 )
    cout << "3";
else
    cout << "4";
cout<<endl;
```

Exercise-1

- What will be the output of the following code?

```
int marks= 2;

switch(marks)
{
        case  1: cout<<"You are in year-1";
        case  2: cout<<"You are in year-1";
        case  3: cout<<"You are in year-2";
        case  4: cout<<"You are in year-2";
}
```

Exercise-2

- What will be the output of the following code?

```
int n = -29;
int temp = (n<0)?-n:n;
cout<<temp;
```

Exercise-3

- What will be the output of the following code?

```
int x=21, y=31, z=44;

if(x>16 && y>x || z%2==0 )
        cout<<"Hello";
else
        cout<<"World!";
```

Exercise-4

- What will be the output of the following code?

```
int a = 5, b = 30;
        if(a > b)
                if(b > 0)
                        a = a + 5;
                else
                        if(b >= 30)
                                b = b * 10;

 cout<<"a="<<a<<" "<<", b="<<b;
```

Exercise-5

- What will be the output of the following code?

```
int x = 5, y = 30;

if(y/x > 2)
        if(y % x !=2)
                x = x + 2;

cout<<x<<"\n"<<y;
```

Exercise-6

- Write a program that ask the user to enter a number in the range of 1—10.

Use a switch statement and display corresponding Roman Number against each entered decimal value. E.g.,

- 1 → I
- 2 → II
- 3 → III
- 4 → IV

Exercise-7

➢ Write a program that ask to input value in seconds. Then the program converts the number of seconds into days, hours, minutes, and seconds value. In the end, the program shows the output in the following format:

Days:2 Hours:13 Minuts:32 Seconds:11

Exercise-8

➢ Write a program that calculate the person's Body Mass Index (BMI). The BMI is often used to determine if a person is overweight, underweight for his/her height.

BMI is calculated as:

BMI = (weight * 703) / (height*height)

Here weight is in pounds and height is in inches.
The program should display the message:

- "Optimal Weight" if BMI is 18.5—25
- "Under Weight" if BMI is < 18.5
- "Over Weight" if BMI is > 25

Any Questions!