

# ELEG 1043

## Computer Applications in Engineering





# Lab Course 1

**C++** FOR ENGINEERS  
AND SCIENTISTS <sup>2</sup>

# Acknowledgement

- Some of the slides or images are from various sources. The copyright of those materials belongs to their original owners.

# Objectives

- In this chapter, you will learn about:
  - Display information, Variables and Declaration Statements
  - Assignment and Interactive Input
  - Selection criteria

# The `cout` Object

- **`cout`** object: An output object that sends data to a standard output display device



## Program 2.1

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

int main()
{
    cout << "Hello there world!";

    return 0;
}
```

# The cout Object (continued)

- **Escape sequence:** One or more characters preceded by a backslash, \



## Program 2.3

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

int main()
{
    cout << "Computers everywheren\n as far as\n\nI can see";

    return 0;
}
```

# Comments

- **Comments:** Explanatory remarks in the source code added by the programmer
- **Line comment:** Begins with `//` and continues to the end of the line

- Example: `// this program displays a message`

```
#include <iostream>
```

```
using namespace std;
```

```
int main ()
```

```
{
```

```
    cout << "Hello there world!"; //displays text
```

```
    return 0;
```

```
}
```

# Arithmetic Operations



## Program 2.6

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

int main()
{
    cout << "15.0 plus 2.0 equals "      << (15.0 + 2.0) << endl
         << "15.0 minus 2.0 equals "     << (15.0 - 2.0) << endl
         << "15.0 times 2.0 equals "      << (15.0 * 2.0) << endl
         << "15.0 divided by 2.0 equals " << (15.0 / 2.0) << endl;

    return 0;
}
```



# Variables and Declaration Statements



## Program 2.7a

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

int main()
{
    double grade1 = 85.5;
    double grade2 = 97.0;
    double total, average;

    total = grade1 + grade2;
    average = total/2.0; // divide the total by 2.0
    cout << "The average grade is " << average << endl;

    return 0;
}
```

# A Case Study: Radar Speed Trap

- Step 1: Analyze the Problem
  - Understand the desired outputs
  - Determine the required inputs
- Step 2: Develop a Solution
  - Determine the algorithms to be used
  - Use top-down approach to design
- Step 3: Code the Solution
- Step 4: Test and Correct the Program

# A Case Study: Radar Speed Trap (continued)

- Analyze the Problem
  - Output: Speed of the car
  - Inputs: Emitted frequency and received frequency
- Develop a Solution
  - Algorithm:
    - Assign values to  $f_0$  and  $f_1$
    - Calculate and display speed

# A Case Study: Radar Speed Trap (continued)

- Code the Solution



Program 2.11

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

int main()
{
    double speed, fe, fr;

    fe = 2e10;
    fr = 2.0000004e10;

    speed = 6.685e8 * (fr - fe) / (fr + fe);
    cout << "The speed is " << speed << " miles/hour " << endl;

    return 0;
}
```

# Assignment Operations



## Program 3.1

```
// This program calculates the volume of a cylinder,  
// given its radius and height  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    double radius, height, volume;  
    radius = 2.5;  
    height = 16.0;  
    volume = 3.1416 * radius * radius * height;  
    cout << "The volume of the cylinder is " << volume << endl;  
  
    return 0;  
}
```

# Assignment Operations

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

int main()
{
    int number = 2;
    int k = 0;

    k = ++number;

    cout<<"The value of k is "<<k<<endl;
    cout<<"The value of number is "<<number<<endl;

    return 0;
}
```

# Using Mathematical Library Functions (continued)



## Program 3.9

```
#include <iostream>    // this line can be placed second instead of first
#include <cmath>        // this line can be placed first instead of second
using namespace std;

int main()
{
    int height;
    double time;

    height = 800;
    time = sqrt(2 * height / 32.2);
    cout << "It will take " << time << " seconds to fall "
         << height << " feet.\n";

    return 0;
}
```

# Program Input Using **cin**



## Program 3.12

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

int main()
{
    double num1, num2, product;

    cout << "Please type in a number: ";
    cin >> num1;
    cout << "Please type in another number: ";
    cin >> num2;
    product = num1 * num2;
    cout << num1 << " times " << num2 << " is " << product << endl;

    return 0;
}
```



# Program Input Using **cin** (continued)



## Program 3.13

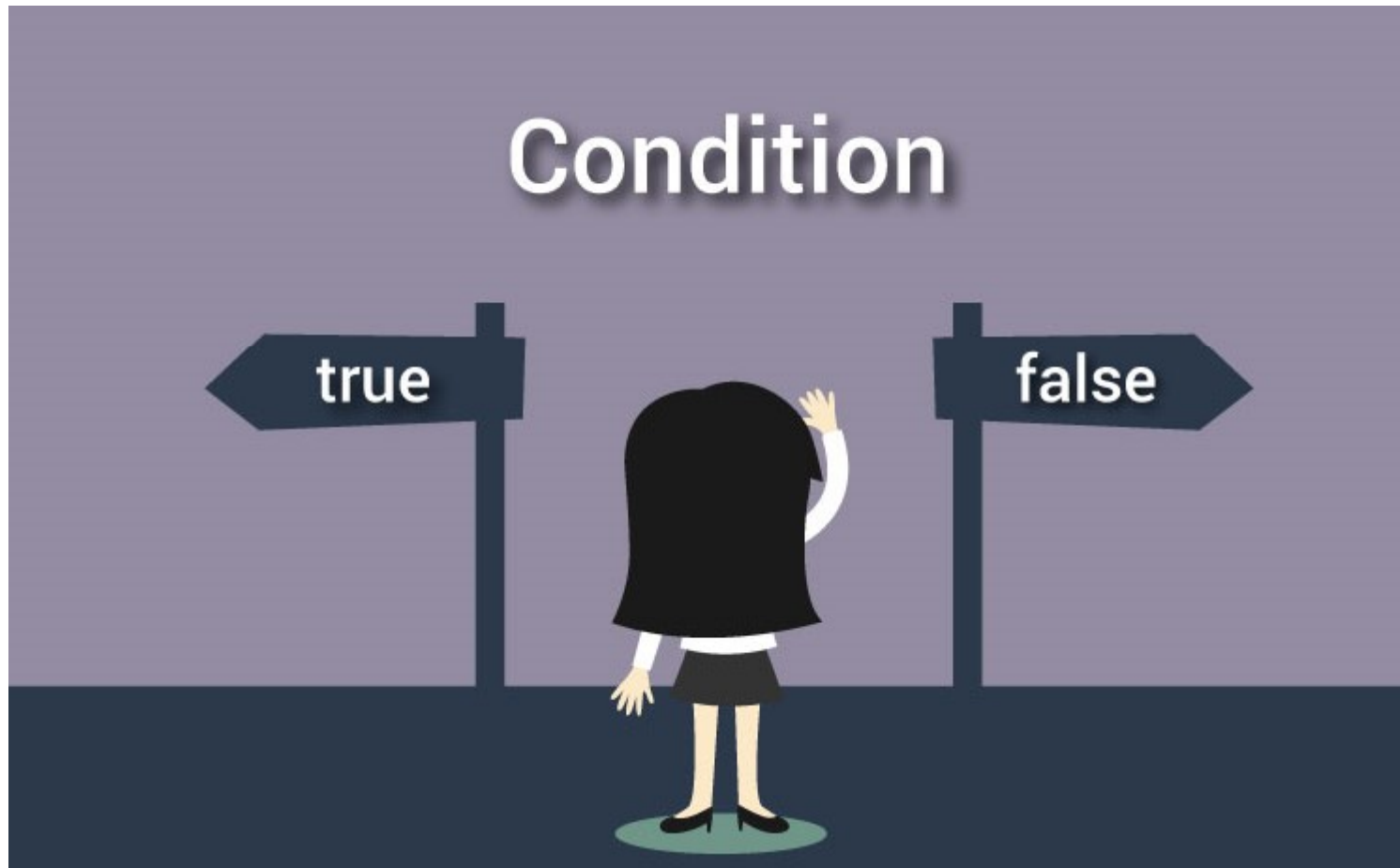
```
#include <iostream>
using namespace std;

int main()
{
    int num1, num2, num3;
    double average;

    cout << "Enter three integer numbers: ";
    cin >> num1 >> num2 >> num3;
    average = (num1 + num2 + num3) / 3.0;
    cout << "The average of the numbers is " << average << endl;

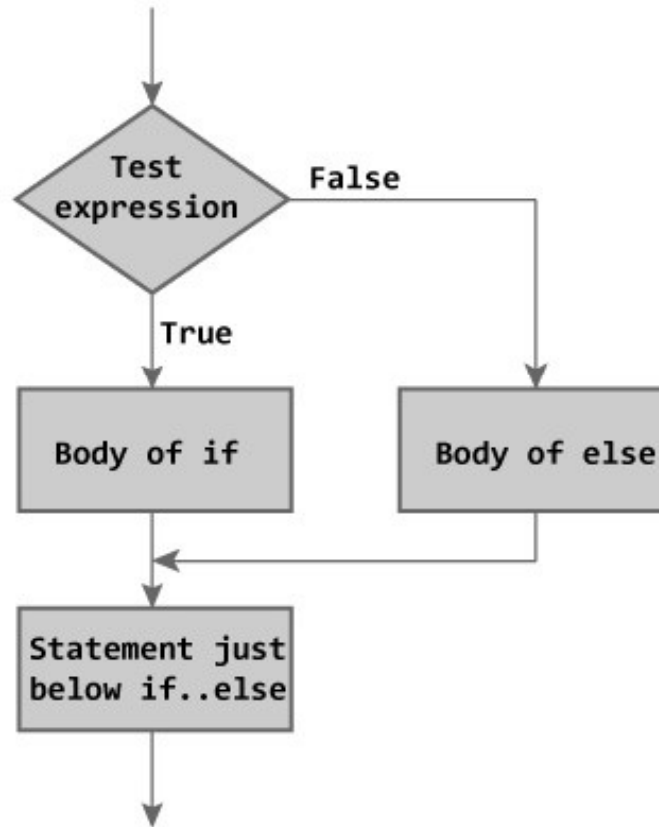
    return 0;
}
```

# Selection Criteria



<https://www.programiz.com/c-programming/c-if-else-statement>

# Flowchart of if statement



# Relational Operators

- **Relational expression:** Compares two operands or expressions using **relational operators**

Relational Operator	Meaning	Example
<	Less than	age < 30
>	Greater than	height > 6.2
<=	Less than or equal to	taxable <= 20000
>=	Greater than or equal to	temp >= 98.6
==	Equal to	grade == 100
!=	Not equal to	number != 250

**Table 4.1** C++'s Relational Operators

# Example 2

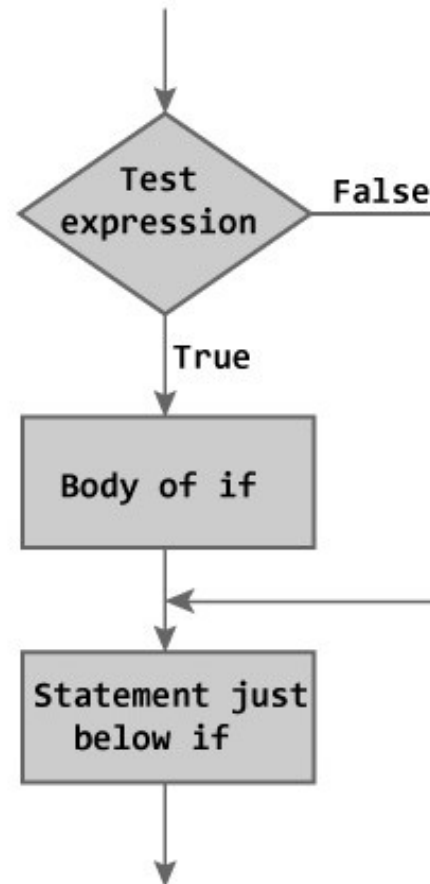
```
#include <iostream>
using namespace std;

int main()
{
    int number;
    cout<<"Enter an integer: \n";
    cin>>number;

    // True if remainder is 0
    if( number%2 == 0 )
        cout<<number<<" is an even integer.\n";
    else
        cout<<number<<" is an odd integer.\n";

    return 0;
}
```

# Flowchart of if statement



# Example 1

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

int main()
{
    int number;
    cout<<"Enter an integer: \n";
    cin>>number;

    // Test expression is true if number is less than 0
    if (number < 0)
    {
        cout<<"You entered "<<number<<"\n";
    }

    cout<<"The if statement is easy.";
    return 0;
}
```

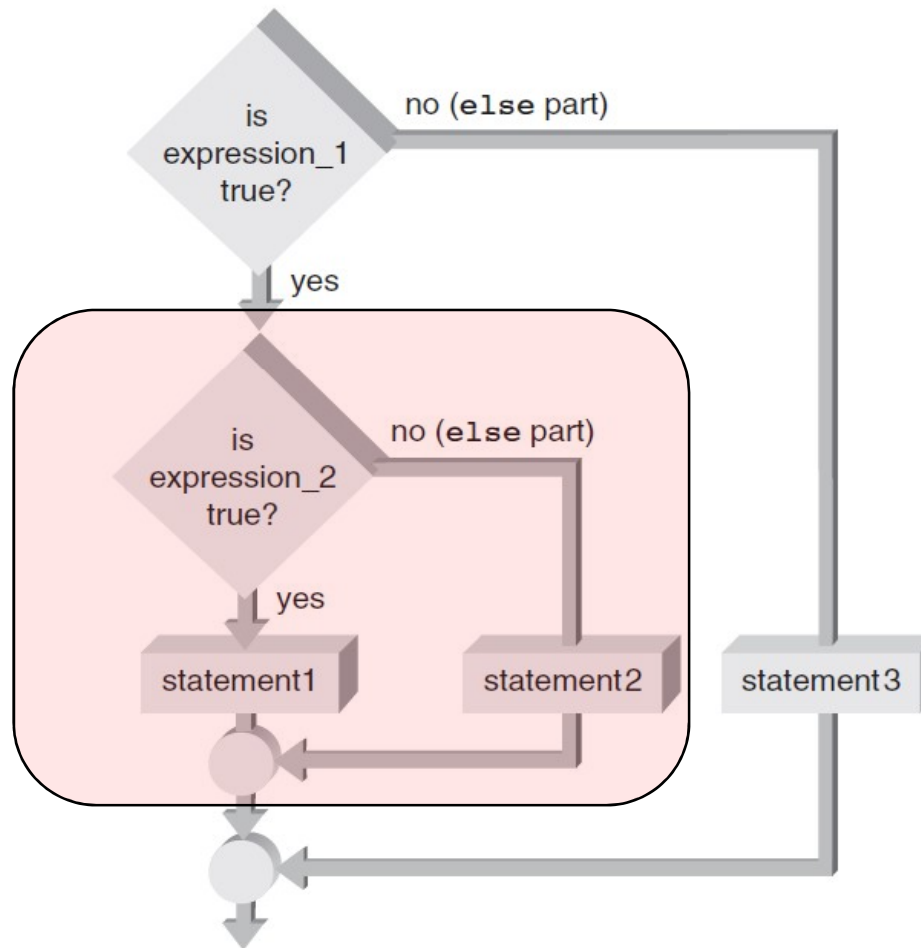
# Logical Operators

- AND (&&): Condition is true only if **both expressions are true**
- OR (||): Condition is true if **either one or both of the expressions is true**
- NOT (!): Changes an expression to its opposite state; **true becomes false, false becomes true**



# Nested `if` Statements (continued)

**Figure 4.4a**  
Nested within the  
`if` part



# Example 1

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

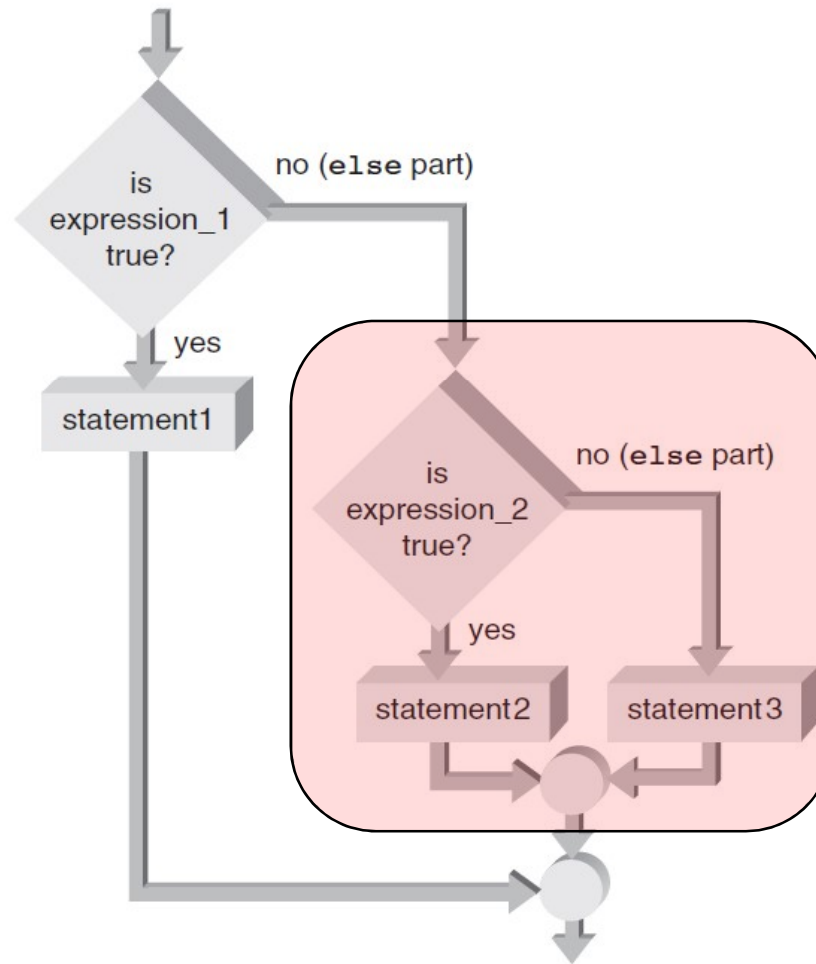
int main()
{
    int number;
    cout<<"Enter an integer: \n";
    cin>>number;

    // Test expression is true if number is less than 0
    if (number < 0)
    {
        if (number < 10)
            cout<<"You entered "<<number<<"\n";
    }

    cout<<"The if statement is easy.";
    return 0;
}
```

# Nested `else` Statements (continued)

**Figure 4.4b**  
Nested within the  
`else` part



# Example 2

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

int main()
{
    int number;
    cout<<"Enter an integer: \n";
    cin>>number;

    // True if remainder is 0
    if( number%2 == 0 )
        cout<<number<<" is an even integer.\n";
    else
        if number > 0
            cout<<number<<" is a positive odd integer.\n";

    return 0;
}
```

# The `if-else` Chain (continued)

- General form of an **`if-else`** chain

```
if (expression_1)
    statement1;
else if (expression_2)
    statement2;
else if (expression_3)
    statement3;
    .
    .
    .
else if (expression_n)
    statementn;
else
    last_statement;
```

# Example 3

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

int main()
{
    int number1, number2;
    cout<<"Enter two integers: \n";
    cin>>number1>>number2;

    if( number1 == number2)
        cout<<number1<<" is equal to "<<number2;
    else if( number1 > number2)
        cout<<number1<<" is larger than"<<number2;
    else
        cout<<number1<<" is smaller than"<<number2;

    return 0;
}
```

# The switch Statement



<https://www.programiz.com/c-programming/c-switch-case-statement>

# The `switch` Statement (continued)

```
switch(expression)
{
    case constant-expression :
        statement(s);
        break;
    case constant-expression :
        statement(s);
        break;
    ...
    default :
        statement(s); }
```



# Example 4

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

int main()
{
    int num=2;

    switch(num)
    {
        case 1:
            cout<<"Case1: Value is: "<<num; break;
        case 2:
            cout<<"Case2: Value is: "<<num; break;
        case 3:
            cout<<"Case3: Value is: "<<num; break;
        default:
            cout<<"Default: Value is: "<<num;
    }

    return 0;
}
```