

Objectives:

- To utilize control structures, functions, and built-in libraries in C++ programming.
- To perform calculations involving quadratic equations using the standard formula.
- To implement logic that checks if a triangle is valid and determines its type.
- To use string manipulation and character analysis for evaluating password strength.

Tools and Libraries Used:

- Programming Language: C++
- IDE: G++
- Libraries: `#include <iostream>`, `include <string>`, `#include <math>`

Theory:

Basics of C++ Programming

C++ is a versatile language used to build efficient programs. Beginners start by learning variables, conditional statements, and loops to solve simple problems.

Structure of a C++ Program

A basic C++ program includes header files (like `<iostream>`) and starts with `main()`, which is the entry point. The program uses `using namespace std;` to access standard features easily. The main function contains the code and ends with `return 0;` to indicate success.

Example:

```
1. #include<iostream>
2. using namespace std;
3. int main() {
4.     cout << "Hello world!";
5.     return 0;
6. }
```

Variables and Data Types

Variables store data. You can declare and assign them like this:

```
1. int age;           // Declaration
2. age = 20;          // Assignment
3. int score = 100;   // Declaration + Initialization
```

Common Data Types:

- int – whole numbers (e.g., int x = 5;)
- float – decimal numbers (e.g., float pi = 3.14;)
- double – more precise decimals (e.g., double d = 2.718;)
- char – single characters (e.g., char c = 'A';)

Variable Naming Rules

- Start with a letter or underscore
- No digits at the beginning
- No space or special characters (except _)
- Case-sensitive (Age ≠ age)

Conditional Statements

Conditional statements control program flow based on conditions.

if statement:

Used when we must check the condition.

Syntax:

```
1. if (condition) {  
2. // Code runs if condition is true  
3. }
```

if...else statement:

Used when we must check the condition and execute true and false condition separately.

Syntax:

```
1. if (condition) {  
2. // Runs if true  
3. } else {  
4. // Runs if false  
5. }
```

else...if ladder:

Used when multiple conditions are to be checked one after another.

Syntax:

```
1. if (condition1) {  
2. // code if condition1 is true  
3. } else if (condition2) {  
4. // code if condition2 is true  
5. } else if (condition3) {  
6. // code if condition3 is true  
7. } else {  
8. // code if none are true  
9. }
```

switch Statement:

Used to select one block of code from many options based on a variable's value.

Syntax:

```
1. switch (expression) {  
2.   case value1:  
3.     // code for case 1  
4.     break;  
5.   case value2:  
6.     // code for case 2  
7.     break;  
8.   ...  
9.   default:  
10.    // code if no cases match  
11. }
```

Loops in C++

for Loop

Used when the number of iterations is known.

Syntax:

```
1. for (initialization; condition; update) {  
2.   // code to repeat  
3. }
```

while Loop

Used when the condition is checked before the loop body and the number of repetitions is not fixed.

Syntax:

```
1. while (condition) {  
2.   // code to repeat  
3. }  
4.
```

do...while Loop

Runs the loop body at least once before checking the condition.

Syntax:

```
1. do {  
2.   // code to repeat  
3. } while (condition);
```