

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);
```

Lab Questions:

Q no 1: Solve $ax^2 + bx + c = 0$ and handle all discriminant cases.

Code:

```
1. #include<iostream>
2. #include<math.h>
3. using namespace std;
4. main()
5. {
6.     float d,x1,x2,a,b,c;
7.     cout<<"The given equation is ax^2+bx+c : "<<endl;
8.     cout<<"a : ";
9.     cin>>a;
10.    cout<<"b : ";
11.    cin>>b;
12.    cout<<"c : ";
13.    cin>>c;
14.    if(a==0)
15.    {
16.        cout<<"Error: The given equation is not quadratic.;"
17.    }
18.    else
19.    {
20.        d=(b*b)-4*a*c;
21.        if(d==0)
22.        {
23.            cout<<"There exists one common root. "<<endl;
24.            x1=-b/(2*a);
25.            cout<<"The root is: "<<x1;
26.        }
27.        else if(d>0)
28.        {
29.            cout<<"There exists two distinct roots. "<<endl;
30.            x1=(-b+sqrt(d))/(2*a);
31.            x2=(-b-sqrt(d))/(2*a);
32.            cout<<"The roots are: "<<x1<<" and "<<x2;
33.        }
34.        else
35.        {
```

```

36.     cout<<"There exists two complex roots. "<<endl;
37.     x1=(-b)/(2*a);
38.     x2=sqrt(-d)/(2*a);
39.     cout<<"The roots are: "<<x1<<"+"i"<<x2<<" and "<<x1<<"-i"<<x2;
40. }
41. }
42. }

```

Output:

```

The given equation is ax^2+bx+c :
a : 0
b : 4
c : 5
Error: The given equation is not quadratic.

```

```

The given equation is ax^2+bx+c :
a : 1
b : -4
c : 4
There exists one common root.
The root is: 2

```

```

The given equation is ax^2+bx+c :
a : 1
b : 4
c : 3
There exists two distinct roots.
The roots are: -1 and -3

```

```

The given equation is ax^2+bx+c :
a : 2
b : 3
c : 4
There exists two complex roots.
The roots are: -0.75+i1.19896 and -0.75-i1.19896

```

Q no 2: Check if three angles form a triangle and classify it.

Code:

```
1. #include<iostream>
2. using namespace std;
3. int main()
4. {
5.     int a,b,c;
6.     cout<<"Enter 1st angle: ";
7.     cin>>a;
8.     cout<<"Enter 2nd angle: ";
9.     cin>>b;
10.    cout<<"Enter 3rd angle: ";
11.    cin>>c;
12.    if(a+b+c==180)
13.    {
14.        if(a==90 || b==90 || c==90)
15.        {
16.            cout<<"Figure is right angled triangle. ";
17.            goto end;
18.        }
19.        else if(a>90 || b>90 || c>90)
20.        {
21.            cout<<"Figure is obtuse angled triangle. ";
22.            goto end;
23.        }
24.        else
25.        {
26.            cout<<"Figure is acute angled triangle. ";
27.            goto end;
28.        }
29.    }
30.    else
31.    {
32.        cout<<"Given angles don't form a triangle.";
33.    }
34.    end:
35. }
```


Output:

```
Enter 1st angle: 90
Enter 2nd angle: 30
Enter 3rd angle: 30
Given angles don't form a triangle.
```

```
Enter 1st angle: 90
Enter 2nd angle: 60
Enter 3rd angle: 30
Figure is right angled triangle.
```

```
Enter 1st angle: 120
Enter 2nd angle: 30
Enter 3rd angle: 30
Figure is obtuse angled triangle.
```

```
Enter 1st angle: 60
Enter 2nd angle: 60
Enter 3rd angle: 60
Figure is acute angled triangle.
```

Q no 3: Check password strength based on length and character rules.

Code:

```
1. #include <iostream>
2. #include <string>
3. using namespace std;
4. class Password {
5.     int hasUpper = 0, hasSymbol = 0, hasLower = 0, hasNumber = 0;
6. public:
7.     int isUpper(char ch) {
8.         if (ch >= 65 && ch <= 90) return 1;
9.         return 0;
10.    }
11.    int isLower(char ch) {
12.        if (ch >= 97 && ch <= 122) return 1;
13.        return 0;
14.    }
15.    int isNumber(char ch) {
16.        if (ch >= 48 && ch <= 57) return 1;
17.        return 0;
18.    }
19.    int isSymbol(char ch) {
20.        if ((ch >= 33 && ch <= 47) ||
21.            (ch >= 58 && ch <= 64) ||
22.            (ch >= 91 && ch <= 96) ||
23.            (ch >= 123 && ch <= 126)) {
24.            return 1;
25.        }
26.        return 0;
27.    }
28.    int checkPass(string password) {
29.        if (password.length() <= 8) {
30.            return 0;
31.        }
32.        for (int i = 0; i < password.length(); i++) {
33.            char ch = password[i];
34.            if (isUpper(ch)==1) hasUpper = 1;
35.            else if (isLower(ch)==1) hasLower = 1;
36.            else if (isNumber(ch)==1) hasNumber = 1;
```

```

37.         else if (isSymbol(ch)==1) hasSymbol = 1;
38.     }
39.     if (hasUpper == 1 && hasLower == 1 && hasNumber == 1 &&
hasSymbol == 1)
40.         return 1;
41.     else
42.         return 0;
43. }
44. };
45. int main() {
46.     Password pa;
47.     string pass;
48.     cout << "Enter password: ";
49.     cin >> pass;
50.     if (pa.checkPass(pass) == 1) {
51.         cout << "Password is strong." << endl;
52.     } else {
53.         cout << "Password is not strong." << endl;
54.     }
55.     return 0;
56. }

```

Output:

```

Enter password: himalaya
Password is not strong.

```

```

Enter password: himalaya123
Password is not strong.

```

```

Enter password: Himalaya123
Password is not strong.

```

```

Enter password: Himalaya@123
Password is strong.

```

Conclusion:

In this lab, we focused on applying fundamental C++ concepts by working through a range of practical problems. It emphasized the use of control structures, functions, and string manipulation techniques. This hands-on experience helps establish a solid base for advancing into more complex topics like C++ and object-oriented programming.