

HIMALAYA COLLEGE OF ENGINEERING

# Advanced C++ Programming Lab Report

### Lab 1: Quadratic Equations, Triangle Classification and Password Strength Checking

**Prepared By:** kushal Bote

**Subject:** Object Oriented Programming (OOP)

**Program:** Bachelors of Electronics and Computer Engineering

**Institution:** Himalaya College of Engineering

**Date:** May 25, 2025

# 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> using namespace std; int main() {

cout << "Hello world!"; return 0;

}

2.

3.

4.

5.

6.

## Variables and Data Types

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

1.

int age;

age = 20;

// Declaration

// Assignment

int score = 100; // Declaration + Initialization

2.

3.

#### 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:

if (condition) {

// Code runs if condition is true

}

1.

2.

3.

#### if...else statement:

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

Syntax:

if (condition) {

// Runs if true

} else {

// Runs if false

}

1.

2.

3.

4.

5.

#### else….if ladder:

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

Syntax:

1.

if (condition1) {

// code if condition1 is true

} else if (condition2) {

// code if condition2 is true

} else if (condition3) {

// code if condition3 is true

} else {

// code if none are true

}

2.

3.

4.

5.

6.

7.

8.

9.

#### switch Statement:

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

value. Syntax:

1.

switch (expression) { case value1:

// code for case 1 break;

case value2:

// code for case 2 break;

...

default:

// code if no cases match

}

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

## Loops in C++

#### for Loop

Used when the number of iterations is known.

Syntax:

1.

for (initialization; condition; update) {

// code to repeat

}

2.

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) {

// code to repeat

}

2.

3.

4.

#### do...while Loop

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

Syntax:

1.

do {

// code to repeat

} while (condition);

2.

3.

# Lab Questions:

### **Q no 1:** Solve 𝑎2 + 𝑏𝑥 + 𝑐 = 0 and handle all discriminant cases.

Code:

#include<iostream> #include<math.h> using namespace std; main()

{

float d,x1,x2,a,b,c;

cout<<"The given equation is ax^2+bx+c : "<<endl; cout<<"a : ";

cin>>a; cout<<"b : "; cin>>b; cout<<"c : "; cin>>c; if(a==0)

{

cout<<"Error: The given equation is not quadratic.;

}

else

{

d=(b\*b)-4\*a\*c; if(d==0)

{

cout<<"There exists one common root. "<<endl; x1=-b/(2\*a);

cout<<"The root is: "<<x1;

}

else if(d>0)

{

cout<<"There exists two distinct roots. "<<endl; x1=(-b+sqrt(d))/(2\*a);

x2=(-b-sqrt(d))/(2\*a);

cout<<"The roots are: "<<x1<<" and "<<x2;

}

else

{

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

cout<<"There exists two complex roots. "<<endl; x1=(-b)/(2\*a);

x2=sqrt(-d)/(2\*a);

cout<<"The roots are: "<<x1<<"+i"<<x2<<" and "<<x1<<"-i"<<x2;

}

}

}

37.

38.

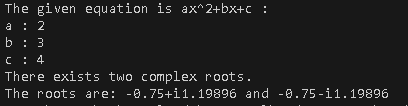
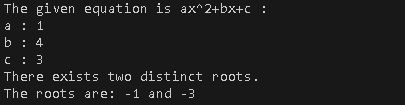
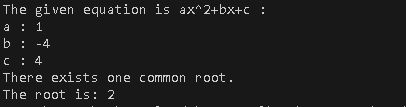
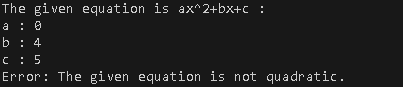
39.

40.

41.

42.

Output:



### **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. }

1. else if(a>90 || b>90 || c>90)
2. {
3. cout<<"Figure is obtuse angled triangle. ";
4. goto end;

23. }

1. else
2. {
3. cout<<"Figure is acute angled triangle. ";
4. goto end;

28. }

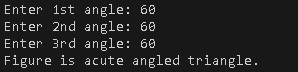
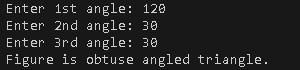
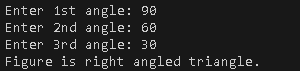
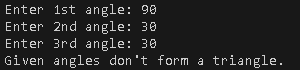
29. }

1. else
2. {
3. cout<<"Given angles don't form a triangle.";

33. }

1. end:
2. }

Output:



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

#include <iostream> #include <string> using namespace std; class Password {

int hasUpper = 0, hasSymbol = 0, hasLower = 0, hasNumber = 0; public:

int isUpper(char ch) {

if (ch >= 65 && ch <= 90) return 1; return 0;

}

int isLower(char ch) {

if (ch >= 97 && ch <= 122) return 1; return 0;

}

int isNumber(char ch) {

if (ch >= 48 && ch <= 57) return 1; return 0;

}

int isSymbol(char ch) {

if ((ch >= 33 && ch <= 47) || (ch >= 58 && ch <= 64) || (ch >= 91 && ch <= 96) || (ch >= 123 && ch <= 126)) { return 1;

}

return 0;

}

int checkPass(string password) { if (password.length() <= 8) {

return 0;

}

for (int i = 0; i < password.length(); i++) { char ch = password[i];

if (isUpper(ch)==1) hasUpper = 1;

else if (isLower(ch)==1) hasLower = 1; else if (isNumber(ch)==1) hasNumber = 1;

Code:

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.

39.

else if (isSymbol(ch)==1) hasSymbol = 1;

40.

41.

42.

43.

}

if (hasUpper == 1 && hasLower == 1 && hasNumber == 1 && hasSymbol == 1)

return 1;

else

return 0;

}

44. };

1. int main() {
2. Password pa;
3. string pass;
4. cout << "Enter password: ";
5. cin >> pass;
6. if (pa.checkPass(pass) == 1) {
7. cout << "Password is strong." << endl;
8. } else {
9. cout << "Password is not strong." << endl;

54. }

55. return 0;

56. }

Output:



# 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.