Himalaya College of Engineering

Advanced C++ Programming Lab Report

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

Prepared By: Gauri Shankar Yadav

Subject: Object-Oriented Programming (OOP)

Program: Bachelor of Electronics Engineering

Institution: Himalaya College of Engineering

Date: June 8,2025

Introduction to C++ Programming (Basics, Conditionals, Loops, and Switch-Case)

C++ is a powerful, high-performance programming language widely used for system/software development, game programming, and competitive coding. It supports both procedural and object-oriented programming. A basic C++ program consists of functions, variables, control structures (like loops and conditionals), and input/output operations.

The entry point of a C++program is the main() function. For input and output, we use the cin and cout objects from the iostream library.

```
#include <iostream>

using namespace std;

int main() {
// Your code here

return 0;
}
```

Variables and Data Types

Variables store data. Common data types include:
int for integers
float for decimals

char for single characters

bool for boolean values (true/false)

string for text (needs #include <string>)

```cpp
int age =21;
float temp =36.6;
char grade ='A';
bool passed = true;
```

Conditional Statements

Conditional statements allow the program to make decisions based on conditions.

```cpp
int marks =85;

if (marks >=90) {

    cout <<"Grade A";

} else if (marks >=75) {

    cout <<"Grade B";

} else {
    cout <<"Grade C";

}
```

Switch- Case Statement

The switch statement is an alternative to multiple if conditions, best used with discrete

v al ues.

```
int day =2;
switch (day) {
    case 1: cout <<"Monday"; break;

    case 2: cout <<"Tuesday"; break;

    def aul t: cout <<"Invalid";

}
```

Loops in C++

Loops are used to repeat a block of code.

For Loop(known iterati ons):

```
for (int i =1; i <= 5; i ++) {

    cout <<i <<" ";
}
```

While Loop(checks before):

```
int i =1;
while (i <= 5) {
```

```
cout <<i <<" ";

i ++;
}
```
Do-While Loop(runs at least once):
```
int i =1;
do{
cout <<i <<" ";

i ++;
} while (i <=5);
```

Obj ectives:

1. Understand and apply C++control structure, functions, and standard libraries.

2. Impl ement mathemati cal computations using the quadratic formula.

3. Develop logic for triangle validation and classification.
4. Apply string handling and character checking for password strength.

5. Practice working with loops and conditions for prime number generation.

Tools and Libraries Used

- Programming Language: C++
- IDE: Any C++ compiler (G++, Code::Blocks, etc.)
- Libraries: #include <iostream>, #include <cmath>, #include <string>

Qn.1 Solve ax2 +bx +c =0 and handle all discriminant cases.
Code:
Output:

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

int main () {
double a , b , c ,D,x,y,X1,X2;
cout << " Enter coefficients a:";
cin >>a;
cout << " Enter coefficients b:";
cin >>b;
cout <<" Enter coefficients c:";
cin >>c;

D= b * b - 4* a * c ;
if (D> 0) {
x= ( - b + sqrt (D) ) / (2* a ) ;
y= ( - b - sqrt (D) ) / (2* a ) ;
cout << " Real and distinct roots : "<<x<< " and "<<y;

} else if (D== 0) {
x= -b / (2* a ) ;
cout << " Real and equal roots : " <<x;
} else {
double real= -b / (2* a ) ;
double imag= sqrt (-D) / (2* a ) ;
cout<<"X1="<<real<<"+"<<imag<<"i"<<endl;
cout<<"X2="<<real<<"-"<<imag<<"i"<<endl;
}
return 0; }
```

```
Enter coefficients a:4
Enter coefficients b:3
Enter coefficients c:6
X1=-0.375+1.16592i
X2=-0.375-1.16592i

----------------------------------
```

Qn.2 Check if three angles form a triangle and classify it.
Code:
Output:

```cpp
# include<iostream >
using namespace std;
int main (){
int a1,a2 ,a3 ;
cout << " Enter 1st angles :";
cin >>a1 ;
  cout << " Enter 2nd angles :";
cin >>a2 ;

cout << " Enter 3rd angles :";
cin >>a3;

int sum = a1 + a2 + a3 ;
if ( sum == 180 && a1 > 0 && a2 > 0 && a3 > 0) {
cout << " Valid Triangle: ";
if ( a1< 90 && a2 < 90 && a3 < 90)
cout << "Acute Triangle ";
else if ( a1 == 90 || a2 == 90 || a3 == 90)
cout << " Right Triangle ";
else
cout << "Obtuse Triangle ";
} else {
cout << "Not a valid triangle .";
}
return 0;
}
```

```
Enter 1st angles :30
Enter 2nd angles :60
Enter 3rd angles :90
Valid Triangle:  Right Triangle
------------------------------------
```

Qn 3 Check password strength based on length and character rules.
Code:
Output:

```cpp
#include<iostream>
#include<cstring>
using namespace std;
int main(){
string psw;
cout<<"Enter the password:";
cin>>psw;

int hasUpper = 0;
    int hasSpecial = 0;
    int len = psw.length();
if(len>8){
    for (int i = 0; i < len; i++) {
        if (psw[i] >= 65 && psw[i] <= 90) {
            hasUpper = 1;
        }
        if (psw[i] == 35 || psw[i] == 36 || psw[i] == 37 || psw[i] == 94) {
            hasSpecial = 1;
        }
    }

    if (len >= 8 && hasUpper && hasSpecial) {
        cout << "Password is strong";
    } else {
        cout << "Password is weak";
    }
}
else{
    cout<<"Password is must contain 8 character!";
}
return 0;}
```

```
Enter the password:Arun$Sauden
Password is strong
Process returned 0 (0x0)    execution time : 7.442 s
Press any key to continue.
```

Discussion:

In the l ab, we worked on three di f f erent programs: solving quadratic equations, classifying triangles, and checking password strength. For quadratic equations, we used the formula to find the roots and learned how the discriminant decides if the roots are real or complex. In the triangle classification task, we used if-else statements to check if a triangle is equi l ateral , isosceles, or scalene based on side lengths. For password strength checking, we used conditions to check for length, digits, uppercase, lowercase, and special characters. It was a good ex erci se to appl y logic and practice usi ng conditions in real problems.

Conclusion:

Working on quadratic equations, tri angl e cl assi f i cati on, and password strength checking helped us understand the practical use of conditional statements and logical operations in C ++. These programs showed how real - worl d problems can be solved usi ng si mpl e programming concepts like `if-else`, loops, and input/output. It al so improved our problem-solving skills and logic building. Overall, this lab reinforced the importance of understanding control structures and appl yi ng them eff ectivel y in different situations.