



Himalaya College of Engineering

Advanced C++ Programming Lab Report

Lab 1: Introduction to Object Oriented Programming

Prepared By: Ankit Belbase(HCE081BEI007)

Subject: Object-Oriented Programming (OOP)

Program: Bachelor of Electronics Engineering

Institution: Himalaya College of Engineering

Date: June 8,2025

Objectives:

1. Understand and apply C++ control structure, functions, and standard libraries.
2. Implement mathematical 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: Code Blocks
- Libraries: `#include <iostream>`, `#include <cmath>`, `#include <cstring>`

Theory:

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>)

```
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.

```
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 values.

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

Loops in C++

Loops are used to repeat a block of code.

For Loop (known iterations):

```
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);
```

Ques 1: Write a program to solve a quadratic equation.

Code:

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

int main() {
    double a, b, c;
    cin >> a >> b >> c;
    double d = b * b - 4 * a * c;
    if (d > 0) {
        double r1 = (-b + sqrt(d)) / (2 * a);
        double r2 = (-b - sqrt(d)) / (2 * a);
        cout << "Roots are real and different: " << r1 << " " << r2 << endl;
    } else if (d == 0) {
        double r = -b / (2 * a);
        cout << "Roots are real and same: " << r << endl;
    } else {
        double real = -b / (2 * a);
        double imag = sqrt(-d) / (2 * a);
        cout << "Roots are complex: " << real << " + " << imag << "i, " << real << " - " << imag << "i" <<
endl;
    }
    return 0;
}
```

Output:

```
1
2
3
Roots are complex: -1 + 1.41421i, -1 - 1.41421i
```

Ques 2: Write a program to check if three angles form a triangle and classify it.

Code:

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

int main() {
    int a, b, c;
    cout << "Enter three angles of triangle: ";
    cin >> a >> b >> c;
    if (a + b + c == 180 && a > 0 && b > 0 && c > 0) {
        if (a == 90 || b == 90 || c == 90)
            cout << "Right-angled triangle\n";
        else if (a < 90 && b < 90 && c < 90)
            cout << "Acute-angled triangle\n";
        else
            cout << "Obtuse-angled triangle\n";
    } else {
        cout << "Not a valid triangle\n";
    }
    return 0;
}
```

Output:

```
Enter three angles of triangle: 50
70
90
Not a valid triangle
```

Ques 3: Write a program to check password strength based on length and character rules.

Code:

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

bool isUpper(char ch) {
    return ch >= 'A' && ch <= 'Z';
}

bool isLower(char ch) {
    return ch >= 'a' && ch <= 'z';
}

bool isDigit(char ch) {
    return ch >= '0' && ch <= '9';
}

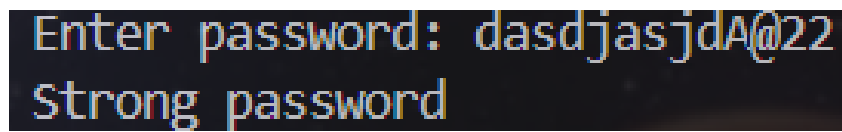
bool isStrongPassword(const string& pwd) {
    if (pwd.length() < 8) return false;
    bool hasUpper = false, hasLower = false, hasDigit = false, hasSpecial = false;
    for (char ch : pwd) {
        if (isUpper(ch)) hasUpper = true;
        else if (isLower(ch)) hasLower = true;
        else if (isDigit(ch)) hasDigit = true;
        else hasSpecial = true;
    }
    return hasUpper && hasLower && hasDigit && hasSpecial;
}

int main() {
    string password;
    cout << "Enter password: ";
    getline(cin, password);
```



```
if (isStrongPassword(password))  
    cout << "Strong password\n";  
else  
    cout << "Weak password\n";  
return 0;  
}
```

Output:

A terminal window with a dark background. The first line shows the prompt "Enter password:" followed by the input "dasdjasjdA@22". The second line shows the output "Strong password".

```
Enter password: dasdjasjdA@22  
Strong password
```

Discussion:

In the lab, we worked on three different 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 equilateral, 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 exercise to apply logic and practice using conditions in real problems.

Conclusion:

Working on quadratic equations, triangle classification, and password strength checking helped us understand the practical use of conditional statements and logical operations in C++. These programs showed how real-world problems can be solved using simple programming concepts like `if-else`, loops, and input/output. It also improved our problem-solving skills and logic building. Overall, this lab reinforced the importance of understanding control structures and applying them effectively in different situations.