
Week #3

Objective: The objective of this lab is to provide students with an opportunity to practice and reinforce their understanding of repetition and conditional structures in C++

Introduction: In this lab session, students will dive into the world of programming logic and decision-making.

Repetition and conditional structures are fundamental concepts in programming that allow them to control the flow of their code and execute certain blocks of instructions repeatedly or conditionally based on specific criteria.

Throughout this lab, the students will explore various techniques for implementing repetition using for loops, while loops, and conditional structures such as if, if-else, and if-else-if statements. These constructs are essential tools in a programmer's toolkit and are used extensively in real-world applications to automate tasks and solve complex problems.

By practicing these concepts through hands-on exercises and challenges, the students will enhance their problem-solving skills and develop a deeper understanding of how to write efficient and effective C++ code. So, let's get started and embark on this journey to master the art of programming logic!

Conditional Structures:

In OOP, conditional structures like if-else statements and switch-case statements are used to control the flow of execution based on certain conditions. These structures allow you to make decisions within your code and execute specific blocks of code based on whether certain conditions are true or false.

If-Else Statements: An if-else statement evaluates a condition and executes a block of code if the condition is true, and another block of code if the condition is false. This structure allows for branching based on a single condition.

```
if (condition) {  
    // Code to execute if condition is true  
} else {  
    // Code to execute if condition is false  
}
```

Switch-Case Statements: A switch-case statement evaluates an expression and compares it to multiple values (cases). It executes the block of code associated with the first case that matches the expression. Switch-case statements are often used when there are multiple possible conditions to check.

```
switch (expression) {  
    case value1:  
        // Code to execute if expression is equal to value1  
        break;  
    case value2:  
        // Code to execute if expression is equal to value2  
        break;  
    default:  
        // Code to execute if expression doesn't match any case  
}
```

Loops: Loops are used to execute a block of code repeatedly as long as a specified condition is true. In OOP, loops such as for, while, and do-while are commonly used to iterate over collections, perform repetitive tasks, and control program flow.

For Loops: A for loop is used to iterate over a range of values or elements in a collection. It typically consists of an initialization step, a condition for continuing the loop, and an update step to modify the loop variable.

```
for (initialization; condition; update) {  
    // Code to execute repeatedly as long as condition is true  
}
```

While Loops: A while loop executes a block of code as long as a specified condition is true. It continuously evaluates the condition before each iteration.

```
while (condition) {  
    // Code to execute repeatedly as long as condition is true  
}
```

Do-While Loops: A do-while loop is similar to a while loop, but it executes the block of code at least once before checking the condition for subsequent iterations. This ensures that the block of code is executed at least once, regardless of whether the condition is initially true or false.

```
do {  
    // Code to execute at least once, then repeatedly as long as  
    condition is true  
} while (condition);
```

In OOP, conditional structures and loops are fundamental tools for controlling the flow of execution, making decisions, and performing repetitive tasks within classes and objects. They help in writing flexible, efficient, and expressive code that can adapt to various scenarios and conditions.

Q.1) Any character is entered by the user; write a program to determine whether the character entered is a capital letter, a small case letter, a digit or a special symbol. The following table shows the range of ASCII values for various characters.

Characters	ASCII Values
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
special symbols	0 - 47, 58 - 64, 91 - 96, 123 – 127

Code:

```
#include <iostream>  
using namespace std;  
  
int startlab2(){  
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;  
    cout << "Start of Lab 02" << endl;  
    return 0;  
}  
  
int l2q1(){  
    char a;  
    cout << "Enter a character: ";  
    cin >> a;  
    int i = int(a);  
    if (i >= 65 && i <= 90){  
        cout << "Capital letter";  
    }  
}
```

```
    else if (i >= 97 && i <= 122){  
        cout << "Small letter";  
    } else if (i >= 48 && i <= 57){  
        cout << "Digit";  
    } else {  
        cout << "Special Character";  
    }  
    return 0;  
}  
  
int main()  
{  
    startlab2();  
    l2q1();  
    return 0;  
}
```

OUTPUT:

Capital:

```
Name: Saad Ali Khan(SE-23083)  
Start of Lab 02  
Enter a character: F  
Capital letter
```

Small:

```
Name: Saad Ali Khan(SE-23083)  
Start of Lab 02  
Enter a character: s  
Small letter
```

Digit:

```
Name: Saad Ali Khan(SE-23083)  
Start of Lab 02  
Enter a character: 5  
Digit
```

Special Character:

```
Name: Saad Ali Khan(SE-23083)  
Start of Lab 02  
Enter a character: ?  
Special Character
```

Q.2) Write a program to calculate the monthly telephone bills as per the following rule:

Minimum Rs. 200 for upto 100 calls.

Plus Rs. 0.60 per call for next 50 calls.

Plus Rs. 0.50 per call for next 50 calls.

Plus Rs. 0.40 per call for any call beyond 200 calls.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q2()
{
    int calls;
    cout << "Enter number of calls: ";
    cin >> calls;
    if (calls <= 100)
    {
        cout << "Charges: " << 200;
    }
    else if (calls > 100 && calls <= 150)
    {
        cout << "Charges: " << 200 + (0.60 * (calls - 100));
    }
    else if (calls > 150 && calls <= 200)
    {
        cout << "Charges: " << 200 + (0.60 * 50) + (0.50 * (calls - 150));
    }
    else
    {
        cout << "Charges: " << 200 + (0.60 * 50) + (0.50 * 50) + (0.40 * (calls -
200));
    }
    return 0;
}

int main()
{
    startlab2();
    l2q2();
    return 0;
}
```

OUTPUT:

Test Case 1:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter number of calls: 90
Charges: 200
```

Test Case 2:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter number of calls: 145
Charges: 227
```

Test Case 3:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter number of calls: 190
Charges: 250
```

Test Case 4:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter number of calls: 390
Charges: 331
```

Q3) Write a program to check the strength of a password entered by the user. The strength of the password is determined based on the following criteria:

- Minimum length of 8 characters.
- Contains at least one uppercase letter, one lowercase letter, one digit, and one special character.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q3()
{
    string password;
```

```
bool capital = false;
bool small = false;
bool digit = false;
bool special_char = false;
cout << "Enter a password: ";
cin >> password;

for (int i = 0; i <= password.length(); i++)
{
    if (int(password[i]) >= 65 && int(password[i]) <= 90)
    {
        capital = true;
    }
    else if (int(password[i]) >= 97 && int(password[i]) <= 122)
    {
        small = true;
    }
    else if (int(password[i]) >= 48 && int(password[i]) <= 57)
    {
        digit = true;
    }
    else if ((int(password[i]) >= 0 && int(password[i]) <= 47) || (int(password[i])
>= 58 && int(password[i]) <= 64) || (int(password[i]) >= 91 && int(password[i]) <= 96)
|| (int(password[i]) >= 123 && int(password[i]) <= 127))
    {
        special_char = true;
    }
}
if (capital && small && digit && special_char && password.length() > 8)
{
    cout << "Strong password";
}
else
{
    if (capital && small && digit)
    {
        cout << "Password should contain a special character";
    }
    else if (capital && small && special_char)
    {
        cout << "Password should contain a digit";
    }
    else if (capital && special_char && digit)
    {
        cout << "Password should contain a small character";
    }
    else if (special_char && small && digit)
    {
        cout << "Password should contain a capital character";
    }
}
```

```
    }  
    else  
    {  
        cout << "Weak password";  
    }  
}  
  
return 0;  
}  
  
int main()  
{  
    startlab2();  
    l2q3();  
    return 0;  
}
```

Test Case 1:

```
Name: Saad Ali Khan(SE-23083)  
Lab 02  
Enter a password: saadaliKhan041@  
Strong password
```

Test Case 2:

```
Name: Saad Ali Khan(SE-23083)  
Lab 02  
Enter a password: saadlikhan  
Weak password  
PS D:\SE\oops_labs>
```

Q4) Write a program to encrypt and decrypt a text file using a simple encryption algorithm. The encryption algorithm involves shifting each character by a fixed number of positions in the ASCII character set.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

string caesarCipher(const string &text, int shift)
{
    string result = text;
    for (char &c : result)
    {
        if (isalpha(c))
        {
            char base = islower(c) ? 'a' : 'A';
            c = static_cast<char>((c - base + shift) % 26 + base);
        }
    }
    return result;
}

int l2q4()
{
    string text;
    int shift;

    cout << "Enter the text: ";
    getline(cin, text);

    cout << "Enter the shift value: ";
    cin >> shift;

    string encrypted_text = caesarCipher(text, shift);
    string decrypted_text = caesarCipher(encrypted_text, -shift);

    cout << "\nEncrypted Text:\n"
         << encrypted_text << endl;
    cout << "\nDecrypted Text:\n"
         << decrypted_text << endl;
    return 0;
}

int main()
```



```
{  
    startlab2();  
    l2q4();  
    return 0;  
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)  
Lab 02  
Enter the text: hello  
Enter the shift value: 3  
  
Encrypted Text:  
khood  
  
Decrypted Text:  
hello  
PS D:\SE\oops_labs> |
```

Q5) Write a C++ program to generate the multiplication table of a given number.

CODE:

```
#include <iostream>  
using namespace std;  
  
int startlab2()  
{  
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;  
    cout << "Lab 02" << endl;  
    return 0;  
}  
  
int l2q5()  
{  
    int num, limit_num;  
    cout << "Enter the number to get its table: ";  
    cin >> num;  
    cout << "Enter the limit of the table: ";  
    cin >> limit_num;  
    for (int i = 1; i <= limit_num; i++)  
    {  
        cout << num << " * " << i << " = " << num * i << endl;  
    }  
  
    return 0;  
}
```

```
int main()
{
    startlab2();
    l2q5();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter the number to get its table: 5
Enter the limit of the table: 10
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

Q6) Write a C++ program to create a simple menu-driven calculator that performs basic arithmetic operations (addition, subtraction, multiplication, division).

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q6()
{
    int num1, num2, operation;
    cout << "Welcome to simple calculator" << endl;
    cout << "Enter 1 number: ";
    cin >> num1;
    cout << "Enter 2 number: ";
    cin >> num2;
    cout << "Operations" << endl;
    cout << "Enter 1 for addition" << endl;
    cout << "Enter 2 for subtraction" << endl;
    cout << "Enter 3 for multiplication" << endl;
```

```
    cout << "Enter 4 for division" << endl;
    cin >> operation;
    if (operation == 1)
    {
        cout << "The sum is: " << num1 + num2;
    }
    else if (operation == 2)
    {
        cout << "The difference is: " << num1 - num2;
    }
    else if (operation == 3)
    {
        cout << "The product is: " << num1 * num2;
    }
    else if (operation == 4)
    {
        cout << "The division is: " << num1 / num2;
    }
    else
    {
        cout << "Operation is not valid";
    }
    return 0;
}

int main()
{
    startlab2();
    l2q6();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Welcome to simple calculator
Enter 1 number: 10
Enter 2 number: 15
Operations
Enter 1 for addition
Enter 2 for subtraction
Enter 3 for multiplication
Enter 4 for division
3
The product is: 150
PS D:\SE\oops_labs> █
```

Q7) Write a C++ program to generate the Fibonacci series up to a given number of terms.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q6()
{
    int num1 = 0, num2 = 1, sum = 0, limit;
    cout << "Enter number of terms: ";
    cin >> limit;
    for (int i = 0; i <= limit; i++)
    {
        sum = num1 + num2;
        cout << num1 << " ";
        num1 = num2;
        num2 = sum;
    }

    return 0;
}

int main()
{
    startlab2();
    l2q6();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter number of terms: 7
0 1 1 2 3 5 8 13
PS D:\SE\oops_labs>
```

Object-Oriented Programming (Lab Exercise)
Spring 2024
Software Engineering Department, NED University of Engineering and Technology

Q8) Write a C++ program to implement a number guessing game where the user tries to guess a randomly generated number.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q8()
{
    int guessed_num, rand_num;
    string play;
    while (1)
    {
        cout << "Enter a number bw 1 and 100 to check if its same as randomly generated
number: ";
        cin >> guessed_num;
        rand_num = rand() % 100 + 1;
        if (guessed_num == rand_num)
        {
            cout << "Congrulation! You are right" << endl;
        }
        else
        {
            cout << "Opps! wrong number.The correct is: " << rand_num << endl;
        }
        cout << "Do you want to play again(yes/no): ";
        cin >> play;
        if (play == "yes")
        {
            continue;
        }
        else
        {
            break;
        }
    }
    return 0;
}

int main(){
    startlab2();
    l2q8();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter a number bw 1 and 100 to check if its same as randomly generated number: 42
Congrulation! You are right
Do you want to play again(yes/no): yes
Enter a number bw 1 and 100 to check if its same as randomly generated number: 67
Opps! wrong number.The correct is: 68
Do you want to play again(yes/no): no
PS D:\SE\oops_labs> 
```

Q9) Write a C++ program to implement a simple rock, paper, scissors game between the user and the computer.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q9()
{
    int user_num, comp_num;
    string play;

    cout << "Welcome to Rock, Paper, Scissors!\n";
    cout << "Enter 1 for Rock, 2 for Paper, or 3 for Scissors:\n";

    int userChoice;
    cin >> userChoice;

    int computerChoice = rand() % 3 + 1;

    cout << "You chose: ";
    switch (userChoice)
    {
        case 1:
            cout << "Rock\n";
            break;
        case 2:
            cout << "Paper\n";
            break;
        case 3:
            cout << "Scissors\n";
            break;
    }
}
```

```
        default:
            cout << "Invalid choice\n";
            return 1;
        }

        cout << "Computer chose: ";
        switch (computerChoice)
        {
            case 1:
                cout << "Rock\n";
                break;
            case 2:
                cout << "Paper\n";
                break;
            case 3:
                cout << "Scissors\n";
                break;
        }

        if (userChoice == computerChoice)
        {
            cout << "It's a tie!\n";
        }
        else if ((userChoice == 1 && computerChoice == 3) ||
                 (userChoice == 2 && computerChoice == 1) ||
                 (userChoice == 3 && computerChoice == 2))
        {
            cout << "You win!\n";
        }
        else
        {
            cout << "Computer wins!\n";
        }

        return 0;
    }

int main()
{
    startlab2();
    l2q9();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Welcome to Rock, Paper, Scissors!
Enter 1 for Rock, 2 for Paper, or 3 for Scissors:
1
You chose: Rock
Computer chose: Scissors
You win!
PS D:\SE\oops_labs> 
```

Q10) Write a C++ program to display the name of the day of the week based on the day number entered by the user.

CODE:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q10()
{
    int day_num;
    cout << "Enter the day number: ";
    cin >> day_num;
    string days[7] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
"Saturday", "Sunday"};

    for (int i = 1; i <= days->length() + 1; i++)
    {
        if (i == day_num)
        {
            cout << days[i - 1];
        }
        else
        {
            continue;
        }
    }

    return 0;
}
```



```
int main()
{
    startlab2();
    l2q10();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
Enter the day number: 5
Friday
PS D:\SE\oops_labs> 
```

Q11) Write a C++ program to print the following patterns and shapes:

```

  *
 ***
*****
*****
*****
*****
***
 *
```

```

*****
*      *
*      *
*      *
*****
```

```

  1
 212
32123
4321234
543212345
```

```

 *
**
***
****
*****
```

```

 *          *
**          **
***         ***
****        ****
*****       *****
*****       *****
****        ****
***         ***
**          **
 *
```

PATTERN 1:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q11()
{
    // Pattern 1
    for (int i = 0; i < 4; i++)
    {
```

```
        for (int j = 5; j > i; j--)
        {
            cout << " ";
        }
        for (int k = 0; k < i; k++)
        {
            cout << "* ";
        }
        for (int l = 0; l < i + 1; l++)
        {

            cout << "* ";
        }
        cout << endl;
    }
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < i + 1; j++)
        {
            cout << " ";
        }
        for (int k = 4; k > i; k--)
        {
            cout << "* ";
        }
        for (int l = 5; l > i; l--)
        {

            cout << "* ";
        }
        cout << endl;
    }

    return 0;
}

int main()
{
    startlab2();
    l2q11();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02

      *
    * * *
  * * * * *
* * * * * * *
* * * * * * * * *
  * * * * * * *
    * * * * *
      * * *
        *
```

PATTERN 2:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q11()
{
    for (int i = 0; i < 5; i++)
    {
        for (int j = 0; j < 5; j++)
        {
            if (i == 0 || i == 4 ||
                j == 0 || j == 4)
            {
                cout << "* ";
            }
            else
            {
                cout << "  ";
            }
        }
        cout << endl;
    }

    return 0;
}
```

```
}

int main()
{
    startlab2();
    l2q11();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
* * * * *
*       *
*       *
*       *
*       *
* * * * *
```

PATTERN 3:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q11()
{
    for (int i = 1; i < 6; i++)
    {
        for (int j = 5; j > i; j--)
        {
            cout << " ";
        }
        for (int k = i; k >= 1; k--)
        {
            cout << k << " ";
        }
        for (int l = 2; l <= i; l++)
```

```
        {
            cout << 1 << " ";
        }
        cout << endl;
    }

    cout << endl;
    cout << endl;
    return 0;
}

int main()
{
    startlab2();
    l2q11();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02

      1
    2 1 2
  3 2 1 2 3
4 3 2 1 2 3 4
5 4 3 2 1 2 3 4 5
```

PATTERN 4:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q11()
{
    for (int i = 0; i <= 5; i++)
    {
        for (int j = 0; j < i + 1; j++)
```

```
        {
            cout << "* ";
        }
        cout << endl;
    }
    cout << endl;
    cout << endl;
    cout << endl;

    return 0;
}

int main()
{
    startlab2();
    l2q11();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
*
* *
* * *
* * * *
* * * * *
* * * * * *
```

PATTERN 5:

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

int startlab2()
{
    cout << "Name: Saad Ali Khan(SE-23083)" << endl;
    cout << "Lab 02" << endl;
    return 0;
}

int l2q11()
{
```

```
for (int i = 0; i <= 6; i++)
{
    for (int j = 0; j < i + 1; j++)
    {
        cout << "* ";
    }
    for (int j = 5; j > i; j--)
    {
        cout << " ";
    }
    for (int j = 5; j > i; j--)
    {
        cout << " ";
    }
    for (int j = 0; j < i + 1; j++)
    {
        cout << "* ";
    }
    cout << endl;
}

for (int i = 0; i <= 5; i++)
{
    for (int j = 5; j > i; j--)
    {
        cout << "* ";
    }
    for (int j = 0; j < i + 1; j++)
    {
        cout << " ";
    }
    for (int j = 0; j < i + 1; j++)
    {
        cout << " ";
    }
    for (int j = 5; j > i; j--)
    {
        cout << "* ";
    }
    cout << endl;
}
return 0;
}

int main()
{
    startlab2();
    l2q11();
    return 0;
}
```

OUTPUT:

```
Name: Saad Ali Khan(SE-23083)
Lab 02
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * *
* * * * * *
* * * * *
* * * *
* * *
* *
*
```




NED University of Engineering & Technology
 Department of Software Engineering
 Object Oriented Concepts and Programming

COGNITIVE DOMAIN ASSESSMENT RUBRIC LEVEL C3-PLO3				
SKILL SETS	EXTENT OF ACHIEVEMENT			
CRITERIA	0-1	2-3	4-5	TOTAL
Understanding of Object-Oriented Concepts	Poor Understanding of Object-Oriented Concepts	Fair Understanding of Object-Oriented Concepts	Good Understanding of Object-Oriented Concepts	
Design of Object-Oriented Solutions	Poor Design of Object-Oriented Solutions	Fair Design of Object-Oriented Solutions	Good Design of Object-Oriented Solutions	
Implementation of Object-Oriented Solutions	Poor Implementation of Object-Oriented Solutions	Fair Implementation of Object-Oriented Solutions	Good Implementation of Object-Oriented Solutions	
Testing and Debugging	Poor Testing and Debugging	Fair Testing and Debugging	Good Testing and Debugging	
Documentation and Comments	Poor Documentation and Comments	Fair Documentation and Comments	Good Documentation and Comments	

Laboratory Session No. _____

Date: _____

Weighted CLO (Psychomotor Score)	
Remarks	
Instructor's Signature with Date:	