**LAB SESSION 1**

**Objective:** For students to get some practice of:

* Using basic C++ types and user-defined objects
  + Integer Types
  + Basic Integer Operators
* C++ Input and Output
  + Formatting Output
  + Reading from the keyboard
* Named constants
* Expressions

C++ is a powerful and versatile programming language that has played a significant role in the development of software applications across various domains. Here's a brief overview of its importance and history:

**Importance of C++:**

**Efficiency:** C++ allows for low-level manipulation of hardware resources, making it suitable for performance-critical applications such as system software, game development, and embedded systems.

**Portability:** C++ code can be compiled to run on different platforms without modification, making it a popular choice for cross-platform development.

**Flexibility:** C++ supports both procedural and object-oriented programming paradigms, providing developers with a wide range of tools and techniques for building complex software systems.

**Scalability:** C++'s modular design and support for abstraction enable developers to build large-scale projects with ease, facilitating code reuse and maintenance.

**Community and Ecosystem**: C++ has a large and active community of developers, libraries, and frameworks, providing resources and support for building a wide range of applications.

**History of C++:**

**Origin:** C++ was created by Bjarne Stroustrup at Bell Labs in the early 1980s. It evolved from the C programming language with the addition of object-oriented features such as classes, inheritance, and polymorphism.

**First Release:** The first version of C++ was released in 1985 as "C with Classes." It introduced the concept of classes and basic object-oriented programming features.

**Standardization:** C++ underwent several revisions and standardizations over the years to improve the language's clarity, efficiency, and portability. The ANSI/ISO standard for C++ was first published in 1998 (known as C++98) and has since been updated with newer versions (C++03, C++11, C++14, C++17, C++20, and ongoing).

**Adoption:** C++ gained popularity rapidly due to its efficiency, flexibility, and wide range of applications. It became the language of choice for building operating systems, compilers, database systems, game engines, and more.

**Modern Developments:** With each new standard revision, C++ continues to evolve, incorporating new features and improvements to meet the needs of modern software development. Recent additions include features like lambda expressions, smart pointers, and constexpr functions.

Overall, C++ remains a crucial tool for software developers, offering a balance between performance, flexibility, and productivity for a wide range of applications. Its rich history and ongoing development make it a cornerstone of modern programming.

C++ is one the most popular language in the programming world. In this article we will be looking towards 10 basic C++ programs for beginners in CPP. C++ is a powerful general-purpose programming language that was developed in the early 1980s as an extension of the C programming language. It is widely used for developing a wide range of applications, including system software, game development, embedded systems, high-performance applications, and more.

C++ combines both high-level and low-level programming features, offering a balance between performance and abstraction. It supports procedural, object-oriented, and generic programming paradigms, giving developer’s flexibility in designing and implementing their solutions.

**Variables and Data Types: C**++ supports various data types such as int, float, double, char, bool, etc. Variables are declared with a data type and a name:

**int age;** **float temperature;.**

You can initialize variables upon declaration,  **int count = 0;**

**Input and Output:** Input is typically taken from the user via **std::cin** and output is displayed using **std::cout.**

int number; std::cout << "Enter a number: "; std::cin >> number;

std::cout << "You entered: " << number << std::endl;

**Operators:** C++ supports various operators such as :

**arithmetic (+, -, \*, /) assignment (=) comparison (==, !=, <, >) logical (&&, ||, !), etc.**

int a = 5, b = 3;

int sum = a + b;

**Control Structures:**

C++ supports control structures like if, else, switch, while, for, etc., for decision-making and looping. if (condition) { // code block

} else {

// alternative code block

}

**Functions:**

Functions in C++ are blocks of code that perform a specific task.

Functions can have parameters (input) and return values (output).

{

|  |  |  |  |
| --- | --- | --- | --- |
| int add(int x, int y) | | | |
|  | return | x + y; |  |
|  | | |

}

**Strings:**

C++ provides a standard string class std::string to work with text.

Strings can be concatenated using the + operator.

|  |  |
| --- | --- |
| std::string greeting = | "Hello" |

;

**Programming Exercise:**

These questions cover various aspects of basic C++ programming, including arithmetic operations, input/output, and conversion between different units. They provide practice in implementing fundamental programming concepts. **Use setw (width) manipulator for formatting output where required.**

**Question 1**: Calculate Area of a Rectangle: Write a C++ program to calculate the area of a rectangle given its length and width.

**Code:**

#include <iostream>

using namespace std;

int startlab1(){

    cout<<"Name: Saad Ali Khan(SE-23083)"<<endl;

    cout<<"Start of Lab 01"<<endl;

    return 0;

}

int l1q1(){

    float length,width,area;

    cout<<"Enter length of rectangle: ";

    cin>>length;

    cout<<"Enter width of rectangle: ";

    cin>>width;

    area = length \* width;

    cout<<"Area of rectangle is: "<< area;

    return 0;

}

int main(){

    startlab1();

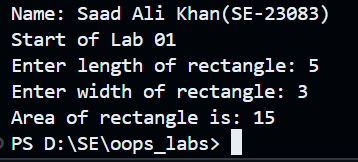
    l1q1();

    return 0;

}

**Expected Output:**

**Area of the rectangle: 15**

****

**Question 2:** Check Even or Odd: Write a C++ program to check if a given number is even or odd.

**Code:**

#include <iostream>

using namespace std;

int startlab1(){

    cout<<"Name: Saad Ali Khan(SE-23083)"<<endl;

    cout<<"Lab 01"<<endl;

    return 0;

}

int l1q2(){

   int num;

   cout<<"Enter a number to check even or odd: ";

   cin>>num;

   if(num % 2==0){

        cout<<num<<" is even";

   }else{

    cout<<num<<" is odd";

   }

   return 0;

}

int main(){

    startlab1();

    l1q2();

    return 0;

}

**Expected Output: 7 is odd**

A screenshot of a computer

Description automatically generated

**Question 3**: Generate Fibonacci Series: Write a C++ program to generate the Fibonacci series up to a given number of terms.

**Code:**

#include <iostream>

using namespace std;

int startlab1(){

    cout<<"Name: Saad Ali Khan(SE-23083)"<<endl;

    cout<<"Lab 01"<<endl;

    return 0;

}

int l1q3(){

    int num,num1=0,num2=1;

    cout<<"Enter a number to get its fibbonacci sequence: ";

    cin>>num;

    if(num<= 0){

        cout<<"Enter number greater than zero";

    }else{

        for(int i=0;i<=num;i++){

            int sum = num1+num2;

            cout<<num1<<" ";

            num1=num2;

            num2=sum;

        }

    }

   return 0;

}

int main(){

    startlab1();

    l1q3();

    return 0;

}

**Expected Output: Fibonacci Series: 0 1 1 2 3 5 8 13 21 34**

A screenshot of a computer

Description automatically generated

**Question 4:** Find Maximum Number: Write a C++ program to find the maximum number among three given numbers.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q4(){

    int num1, num2, num3;

    cout << "Enter number 1: ";

    cin >> num1;

    cout << "Enter number 2: ";

    cin >> num2;

    cout << "Enter number 3: ";

    cin >> num3;

    if (num1 > num2 && num1 > num3)

    {

        cout << "Maximum number: " << num1;

    }else if (num2 > num1 && num2 > num3)

    {

        cout << "Maximum number: " << num2;

    }else

    {

        cout << "Maximum number: " << num3;

    }

    return 0;

}

int main()

{

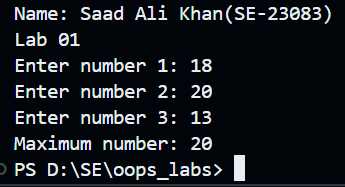
    startlab1();

    l1q4();

    return 0;

}

**Expected Output: Maximum number: 20**



**Question 5:** Reverse a String: Write a C++ program to reverse a given string.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q5()

{

    string word;

    cout << "Enter a word: ";

    cin >> word;

    int len = word.length();

    for (int i = len - 1; i >= 0; i--)

    {

        cout << word[i];

    }

    return 0;

}

int main()

{

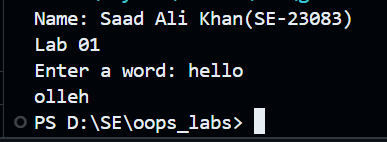
    startlab1();

    l1q5();

    return 0;

}

**Expected Output: Reversed string: olleh**



**Question 6:** Check Prime Number: Write a C++ program to check if a given number is prime or not.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q6(){

    int num;

    bool flagprime;

    cout << "Enter a number to check for prime: ";

    cin >> num;

    for (int i = 2; i <= num / 2; i++) {

        if (num % i == 0){

            flagprime = false;

        }else{

            flagprime = true;

        }

    }

    if (flagprime == true){

        cout << num << " is prime";

    }else{

        cout << num << " is not prime";

    }

    return 0;

}

int main(){

    startlab1();

    l1q6();

    return 0;

}

**Expected Output: 7 is prime**

A black background with white text

Description automatically generated

**Question 7**: Calculate Factorial: Write a C++ program to calculate the factorial of a given number.

#include <iostream>

using namespace std;

int startlab1(){

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q7(){

    int fact = 1, num;

    cout << "Enter a number to calculate its factorial: ";

    cin >> num;

    for (int i = 1; i <= num; i++)

    {

        fact \*= i;

    }

    cout << "Factorial of " << num << ": " << fact;

    return 0;

}

int main(){

    startlab1();

    l1q7();

    return 0;

}

**Expected Output: Factorial of 5: 120**

A screenshot of a computer

Description automatically generated

**Question 8:** Calculate Simple Interest: Write a C++ program to calculate the simple interest given the principal amount, rate of interest, and time period.

#include <iostream>

using namespace std;

int startlab1(){

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q8(){

    int simple\_interest, principal\_amount, time\_period;

    float rate\_of\_interest;

    cout << "Enter principal amount: ";

    cin >> principal\_amount;

    cout << "Enter rate of interest: ";

    cin >> rate\_of\_interest;

    cout << "Enter time period: ";

    cin >> time\_period;

    simple\_interest = (principal\_amount \* rate\_of\_interest \* time\_period) / 100;

    cout << "Simple interest: " << simple\_interest;

    return 0;

}

int main(){

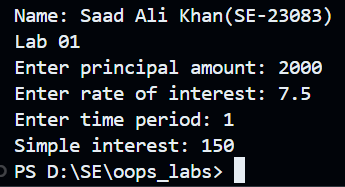
    startlab1();

    l1q8();

    return 0;

}

**Expected Output: Simple interest: 150**



**Question 9**: Check Leap Year: Write a C++ program to check if a given year is a leap year or not.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q9()

{

    int year;

    cout << "Enter year to check for leap: ";

    cin >> year;

    if (year % 4 == 0)

    {

        cout << year << " is a leap year";

    }

    else

    {

        cout << year << " is not a leap year";

    }

    return 0;

}

int main()

{

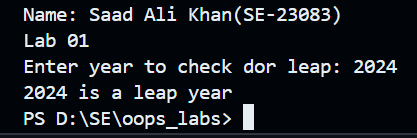
    startlab1();

    l1q9();

    return 0;

}

**Expected Output: 2024 is a leap year**



**Question 10:** Check Armstrong Number: Write a C++ program to check if a given number is an Armstrong number or not.

#include <iostream>

using namespace std;

int startlab1(){

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q10(){

    int num, originalNum, remainder, result = 0;

    cout << "Enter a three-digit integer: ";

    cin >> num;

    originalNum = num;

 while (originalNum != 0){

        remainder = originalNum % 10;

        result += remainder \* remainder \* remainder;

        originalNum /= 10;

    }if(result == num)

        cout << num << " is an Armstrong number.";

    else

        cout << num << " is not an Armstrong number.";

    return 0;

}

int main(){

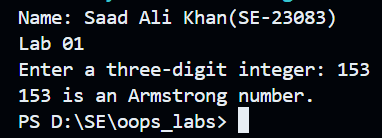
    startlab1();

    l1q10();

    return 0;

}

**Expected Output: 153 is an Armstrong number**



**Question 11:** Sum and Average Calculation: Write a C++ program that prompts the user to enter two numbers, calculates their sum, and then computes their average.

#include <iostream>

using namespace std;

int startlab1(){

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q11(){

    int num1, num2;

    cout << "Enter first number: ";

    cin >> num1;

    cout << "Enter second number: ";

    cin >> num2;

    int sum = num1 + num2;

    double average = (sum) / 2;

    cout << "Sum is " << sum << endl;

    cout << "Average is " << average << endl;

    return 0;

}

int main(){

    startlab1();

    l1q11();

    return 0;

}

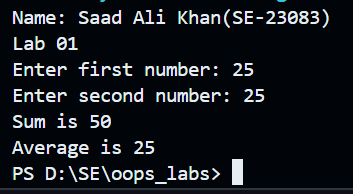
**Expected Output:**

**Enter first number: 25**

**Enter second number: 25**

**Sum is 50**

**Average is 25**

****

**Question 12:** Time Conversion: Write a C++ program that prompts the user to enter time in seconds and then converts it into hours, minutes, and seconds.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q12()

{

    int seconds;

    cout << "Enter the time in seconds: ";

    cin >> seconds;

    int hours = seconds / 3600;

    int minutes = (seconds % 3600) / 60;

    int remaining\_seconds = seconds % 60;

    cout << "Hours in time is: " << hours << endl;

    cout << "Minutes in time: " << minutes << endl;

    cout << "Seconds in time is: " << remaining\_seconds << endl;

    return 0;

}

int main()

{

    startlab1();

    l1q12();

    return 0;

}

**Expected Output:**

**Enter the time in seconds: 3713**

**Hours in time is: 1**

**Minutes in time: 1**

**Seconds in time is: 53**

A screen shot of a computer

Description automatically generated

**Question 13:** Amount in Rupees Breakdown: Write a C++ program that prompts the user to enter an amount in rupees and then breaks it down into denominations of 1000s, 500s, 100s, 50s, 10s, 5s, 2s, and 1s.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q13()

{

    int amount;

    cout << "Enter amount in rupees: ";

    cin >> amount;

    int thousands = amount / 1000;

    amount %= 1000;

    int five\_hundreds = amount / 500;

    amount %= 500;

    int hundreds = amount / 100;

    amount %= 100;

    int fifties = amount / 50;

    amount %= 50;

    int tens = amount / 10;

    amount %= 10;

    int fives = amount / 5;

    amount %= 5;

    int twos = amount / 2;

    amount %= 2;

    int ones = amount;

    cout << "1000’s in the given amount is: " << thousands << endl;

    cout << "500’s in the given amount is: " << five\_hundreds << endl;

    cout << "100’s in the given amount is: " << hundreds << endl;

    cout << "50’s in the given amount is: " << fifties << endl;

    cout << "10’s in the given amount is: " << tens << endl;

    cout << "5’s in the given amount is: " << fives << endl;

    cout << "2’s in the given amount is: " << twos << endl;

    cout << "1’s in the given amount is: " << ones << endl;

    return 0;

}

int main()

{

    startlab1();

    l1q13();

    return 0;

}

**Expected Output:**

**Enter amount in rupees: 5788**

**1000’s in the given amount is: 5**

**500’s in the given amount is: 1**

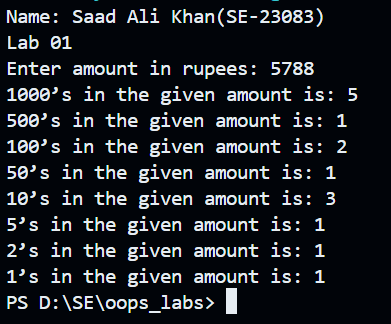
**100’s in the given amount is: 2**

**50’s in the given amount is: 1 10’s in the given amount is: 3**

**5’s in the given amount is: 1**

**2’s in the given amount is: 2**

**1’s in the given amount is: 1**



**Question 14:** Fahrenheit to Celsius Conversion: Write a C++ program that prompts the user to enter a temperature in Fahrenheit and then converts it into Celsius using the formula: Celsius = (Fahrenheit - 32) \* 5 / 9.

#include <iostream>

using namespace std;

int startlab1(){

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q14(){

    double fahrenheit;

    cout << "Enter the Fahrenheit temperature: ";

    cin >> fahrenheit;

    double celsius = (fahrenheit - 32) \* 5 / 9;

    cout << "Celsius Temperature: " << celsius << endl;

    return 0;

}

int main(){

    startlab1();

    l1q14();

    return 0;

}

**Expected Output:**

**Enter the Fahrenheit temperature: 98.6**

**Celsius Temperature: 37.0**

A screen shot of a computer

Description automatically generated

**Question 15:** Two-Digit Integer Reversal: Write a C++ program that inputs a two-digit integer value and outputs its reverse order.

#include <iostream>

using namespace std;

int startlab1()

{

    cout << "Name: Saad Ali Khan(SE-23083)" << endl;

    cout << "Lab 01" << endl;

    return 0;

}

int l1q15()

{

    int number;

    cout << "Enter a 2 digit integer value: ";

    cin >> number;

    int tens\_digit = number / 10;

    int ones\_digit = number % 10;

    int reversed\_number = ones\_digit \* 10 + tens\_digit;

    cout << "Reverse of this value is: " << reversed\_number << endl;

    return 0;

}

int main()

{

    startlab1();

    l1q15();

    return 0;

}

**Expected Output:**

**Enter a 2 digit integer value: 45**

**Reverse of this value is: 54**

A screenshot of a computer

Description automatically generated

**spring 2021**

**Software Engineering Department, NED University of Engineering and Technology**