Sonargaon University

Object-oriented programming

Lab Report

2023

Name: S. M. Shufol Islam

Id: cse2203027065

Section: 27A1

Course Code: cse122

INTRODUCTION

Object-Oriented Programming (OOP) is a way to write software that's become very popular. C++ is one of the most commonly used languages for OOP. In this lab, we will learn about OOP in C++ and how it can be used to create good software.

The main goal of this lab is to teach us the basics of OOP in C++, such as classes and methods. We'll also tried to cover more advanced topics like inheritance. Through hands-on exercises, we will learn how to design and write C++ programs using OOP principles.

We'll use the C++ programming language and an Integrated Development Environment (IDE) like Visual Studio or Code::Blocks. We'll start with simple examples and move on to more complex ones. By the end of the lab, we will have a good understanding of OOP in C++ and be able to write good software.

In summary, this lab teaches us about OOP in C++ and its features. We'll learn how to write software using OOP principles, and be prepared for further studies in computer science or software engineering.

CONTENTS

2.6: C++ Program to input marks in three subjects of a student and calculate the division according to the following conditions:

Percentage	Division
>=60	First
50-59	Second
40-49	Third
<40	Fail

2.7: C++ Program to check that a given year is a leap year or not. If not then calculate the nearest leap year.
2.8: C++ Program to input a character and check that it's a small letter, capital letter, a digit or a special symbol.
3.1:Write a C++ program that take four functions that gets two integer and returns its sum subtraction, multiplication, division.
3.2: Write a C++ program that take two functions that gets three integers and returns maximum and minimum.
3.3: Write a function that gets length and width of a rectangle and returns area.
3.4: Write a function that gets radius of a circle and returns area.
3.5: Write a function that gets three numbers and returns medium.
3.6: Write a function that gets any positive number and returns its factorial.
3.7:Write a function that gets any positive number and returns sum of its digit
3.8: Write a function that gets any positive number and returns digital root.
3.9: Write a function that gets any positive integer and returns its reverse.
3.10: C++ program to check prime number using user defined function.
4.1: C++ program to display prime numbers between two intervals using us defined function.
4.2: Write a function that has one character argument and displays that it's a small letter, capital letter, a digit or a special symbol.
4.3: Write a function that will find the GCD of two numbers. greatest comm Common divisor
4.4: Write a program that will print Fibonacci series upto n value (using user defined function).
4.5: Write a C++ program that take four functions that gets two values and returns its sum subtraction, multiplication, division using function overloading(data type different).
4.6: Write a C++ program that take four functions that get integer values and returns ts sum, subtraction, multiplication, division using function overloading(number of argument different).
5.1: Write a code for class that take an input and show that to output
5.2: Write a code for class that take three numbers as input and calculate summation of those numbers.
5.3: Write a code for class that take two float as private data and calculate summation of those numbers.
5.4: Write a code for class that calculate average of N numbers.
5.5: Create a class called Box that contains one public function (print volume) and three public variables (height and length and width). Declare the height, length and width as integer value. Use public function print volume () to calculate the volume of a box and display the volume.
5.6: Write a code for class which check a number even or odd..
5.7: Write a code for class which check a char vowel or consonant.
6.1: Create a class called Box that contains one public function (print volume) and three public variables (height and length and width). Declare the height, length and width as integer value. Use public function print_volume () to calculate the volume of a box and display the volume

6.2: Create a class called person that maintains a register. In the register, it contains name and address and blood group, date of birth and identification number. Store the name, address and blood group as string and the identification number as an integer value. Use two public functions contain () to store the name, address, date of birth, blood group and identification number. And another public function display() will display the information.

6.3: Create a class called card that maintains a library card catalog entry. In the card, it will store a book's title, author and number of copies on hand. Store the title, author as string and the number of books on hand as an integer value. Use two public functions store () to store the title, author and number of books on hand. And another public function display() will display the information.

6.4: Simple class program to find prime number using class prime. And take the input value as a private member and declare two public member function get_input() to get the input from user. And another function is calculate() to calculate the input is prime or not and display it using calculate () function.

7.1: Simple class program to find factorial number using class factorial. And take the input value as a private member and declare two public member function get_input() to get the input from user. And another function is fact () to calculate the factorial of the input

7.2: Write a Class program to find out the sum of series 1^2+2^2+...+n^2.

7.3: A constructor function will input three numbers and destructor function will print the average of those numbers.

7.4: Parent class will be par which have a string. Print the string with help of a child class named chil. 5. Base class have values of height & width of a triangle. Calculate the area of the triangle with help of a sub class.

Code Starts Here:

Lab Class 1:

1.1:**C++ Program to Print a Sentence**

Code:

```
#include<iostream>
using namespace std;
int main()
{
   cout<<"Hello World!"<<endl;
```

```
    return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\2_1.exe
Hello World!

Process returned 0 (0x0)   execution time : 0.037 s
Press any key to continue.
```

1.2: **C++ Program to Print a Integer Entered by a User**

Code:

```
#include<iostream>
using namespace std;
int main()
{
    cout<<"Please enter an integer: "<<endl;
    int a;
    cin>>a;
    cout<<"You have entered : "<<a<<endl;
    return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\2_2.exe
Please enter an integer:
45
You have entered : 45

Process returned 0 (0x0)   execution time : 2.516 s
Press any key to continue.
```

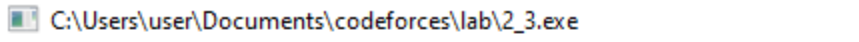
1.3**: C++ Program to Add Two Integers**

Code:

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

int main()
{
    cout<<"Please enter two integer: "<<endl;
    int a,b;
    cin>>a>>b;
    cout<<"Sum of entered integer is: "<<a+b<<endl;
    return 0;
```

```
}
```

Output:



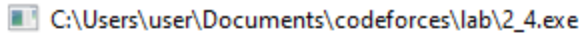

```
Please enter two integer:
45
34
Sum of entered integer is: 79

Process returned 0 (0x0)   execution time : 6.705 s
Press any key to continue.
```

1.4: **C++ Program to Multiply two Floating Point Numbers**

Code:

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

int main()
{
   cout<<"Please enter two floating point number: "<<endl;
   float a,b;
   cin>>a>>b;
   cout<<"Multiplication of entered floating point number is: "<<a*b<<endl;
   return 0;
}
```

Output:




```
Please enter two floating point number:
2.4
3.2
Multiplication of entered floating point number is: 7.68

Process returned 0 (0x0)   execution time : 6.755 s
Press any key to continue.
```

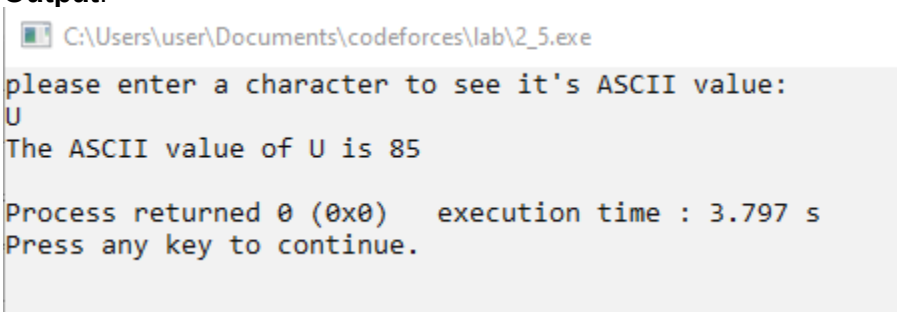
1.5: **C++ Program to Find ASCII Value of a Character**

Code:

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

int main()
{
   char c;
```

```
    cout<<"please enter a character to see it's ASCII value: "<<endl;
    cin>>c;
    cout << "The ASCII value of " << c << " is " << int(c) << endl;
    //printf("%d", c); //this is the way to do in c language
    return 0;
}
```

Output:

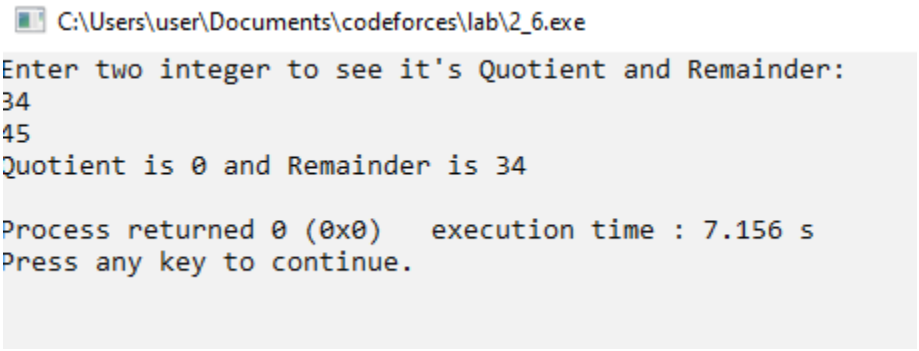


1.6: **C++ Program to Find Quotient and Remainder of Two Integers Entered by User**

Code:

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

int main()
{
    int a,b;
    cout<<"Enter two integer to see it's Quotient and Remainder: "<<endl;
    cin>>a>>b;
    cout<<"Quotient is "<< a/b <<" and Remainder is "<<a%b<<endl;
    return 0;
}
```

Output:



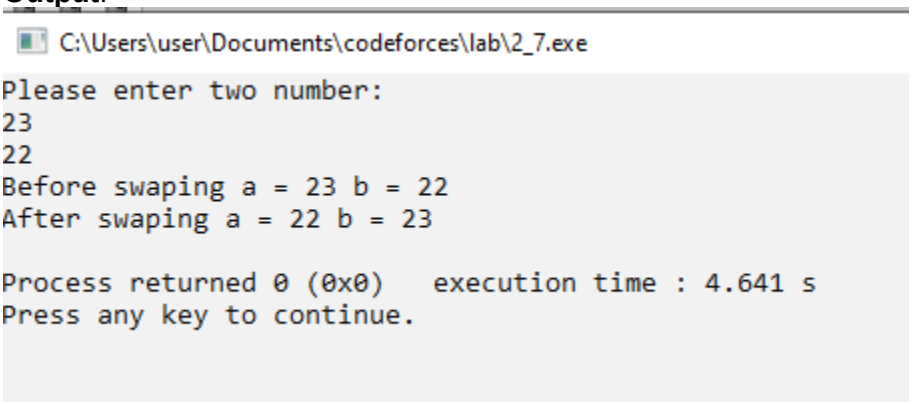
1.7: **C++ Program to Swap Two Numbers**

Code:

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

```
int main()
{
    int a,b,temp;
    cout<<"Please enter two number: "<<endl;
    cin>>a>>b;
    cout << "Before swaping a = " << a << " b = " << b <<endl;
    temp = a;
    a=b;
    b=temp;
    cout << "After swaping a = " << a << " b = " << b <<endl;
    return 0;
}
```

Output:

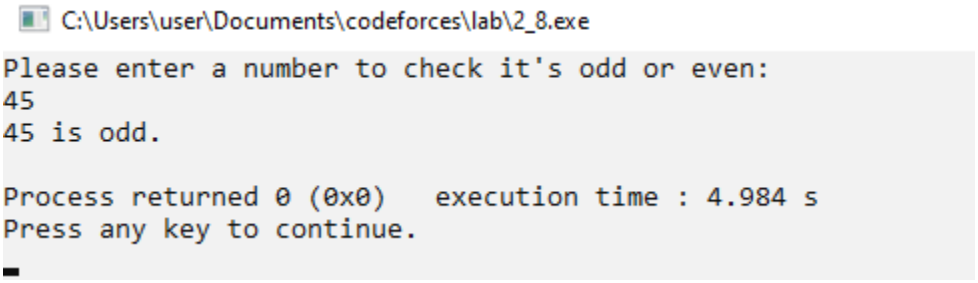


1.8: **C++ Program to Check Whether a Number is Even or Odd**

Code:

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

int main()
{
    int  a;
    cout<<"Please enter a number to check it's odd or even: "<<endl;
    cin>>a;
    if(a%2 == 0)
    {
        cout<<a<<" is even."<<endl;
    }
    else
    {
        cout<<a<<" is odd."<<endl;
    }
    return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- Basic problem solving
- Basic structure of problem's
- Basic Syntax of c++.
- How c++ code executes.

Lab Class 2:

2.1**: C++ Program to Check Vowel or Consonant**

Code:

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

int main()
{
   cout<<"Please enter only alphabet's to check it's vowel or consonent: "<<endl;
   char c;
   cin>>c;
   if(c == 'a' || c == 'e'|| c == 'i'|| c == 'o'|| c == 'u'||c == 'A' || c == 'E'|| c == 'I'|| c == 'O'|| c == 'U')
   {
      cout<<c<<" is Vowel"<<endl;
   }
   else
   {
      cout<<c<<" is consonent"<<endl;
   }
   return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\3_1.exe
Please enter only alphabet's to check it's vowel or consonent:
H
H is consonent

Process returned 0 (0x0)   execution time : 3.262 s
Press any key to continue.
```

2.2**: C++ Program to Find the Largest Number Among Three Numbers**

Code:

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

int main(){
   int a,b,c,res;
   cout<<"Enter three number to see which  one is largest: "<<endl;
   cin>>a>>b>>c;
   res = (a>b)?a:
         (a>c)?a:
            (b>c)?b:c;

   cout<<"Largest number among three number is "<<res<<endl;
   return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\3_2.exe
Enter three number to see which  one is largest:
34
56
34
Largest number among three number is 56

Process returned 0 (0x0)   execution time : 5.694 s
Press any key to continue.
```

2.3**: C++ program to Find all Roots of a Quadratic equation**

Code:

```
#include <iostream>
#include <cmath>
```

```
using namespace std;

int main() {
   double a, b, c, x1, x2, discriminant, realPart, imaginaryPart;
   cout << "Enter coefficients a, b and c: ";
   cin >> a >> b >> c;

   discriminant = b*b - 4*a*c;

   if (discriminant > 0) {
      x1 = (-b + sqrt(discriminant)) / (2*a);
      x2 = (-b - sqrt(discriminant)) / (2*a);
      cout << "Roots are real and different." << endl;
      cout << "x1 = " << x1 << endl;
      cout << "x2 = " << x2 << endl;
   }

   else if (discriminant == 0) {
      cout << "Roots are real and same." << endl;
      x1 = (-b + sqrt(discriminant)) / (2*a);
      cout << "x1 = x2 =" << x1 << endl;
   }

   else {
      realPart = -b/(2*a);
      imaginaryPart =sqrt(-discriminant)/(2*a);
      cout << "Roots are complex and different."  << endl;
      cout << "x1 = " << realPart << "+" << imaginaryPart << "i" << endl;
      cout << "x2 = " << realPart << "-" << imaginaryPart << "i" << endl;
   }

   return 0;
}
```

Output:

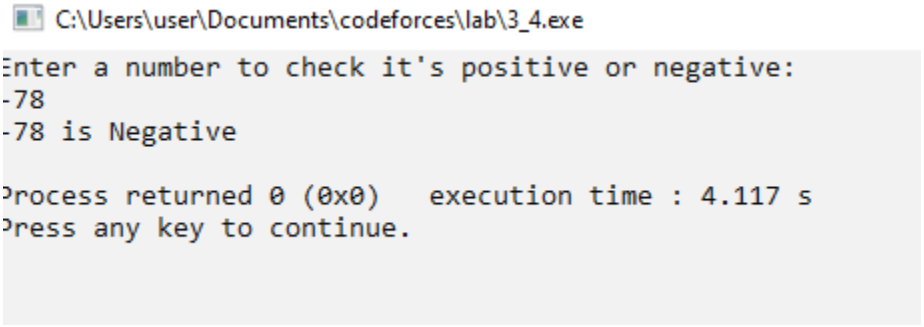
```
C:\Users\user\Documents\codeforces\lab\3_3.exe
Enter coefficients a, b and c: 2 3 6
Roots are complex and different.
x1 = -0.75+1.56125i
x2 = -0.75-1.56125i

Process returned 0 (0x0)   execution time : 4.147 s
Press any key to continue.
```

2.4**: C++ Program to Check Whether a Number is Positive or Negative**

Code:

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

int main()
{
   int a;
   cout<<"Enter a number to check it's positive or negative: "<<endl;
   cin>>a;
   if(a>0)
   {
      cout<<a<<" is Positive"<<endl;
   }
   else
   {
      cout<<a<<" is Negative"<<endl;
   }
   return 0;
}
```

Output:



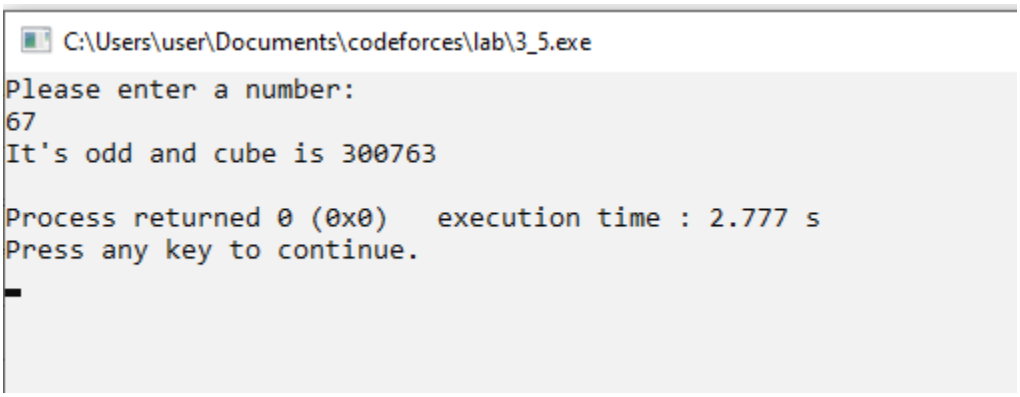
2.5: **C++ Program to input a number. If the number is even, print its square otherwise print its cube.**

Code:

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

int main()
{
   int a;
   cout<<"Please enter a number: "<<endl;
   cin>>a;
   if(a%2 == 0)
   {
      cout<<"It's even and square is "<<a*a<<endl;
   }else{
      cout<<"It's odd and cube is "<<a*a*a<<endl;
   }
```

```
    return 0;
}
```

Output:



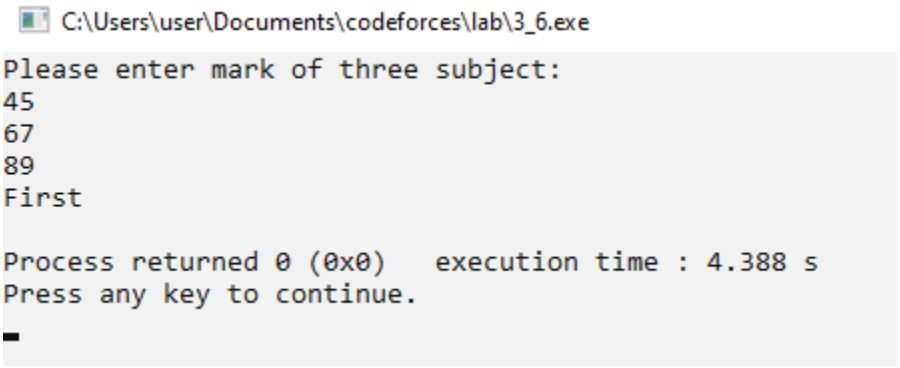
2.6: C++ Program to input marks in three subjects of a student and calculate the division according to the following conditions:

Percentage	**Division**
>=60	First
50-59	Second
40-49	Third
<40	Fail

Code:

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

int main(){
    int a,b,c,avg;
    cout<<"Please enter mark of three subject: "<<endl;
    cin>>a>>b>>c;
    avg = (a+b+c)/3;
    if(avg >= 60){
        cout<<"First"<<endl;
    }else if(avg >= 50 && avg <= 59){
        cout<<"Second"<<endl;
    }else if(avg >= 40 && avg <= 49){
        cout<<"Third"<<endl;
    }else if(avg < 40){
        cout<<"Fail"<<endl;
    }
    return 0;
}
```

Output:

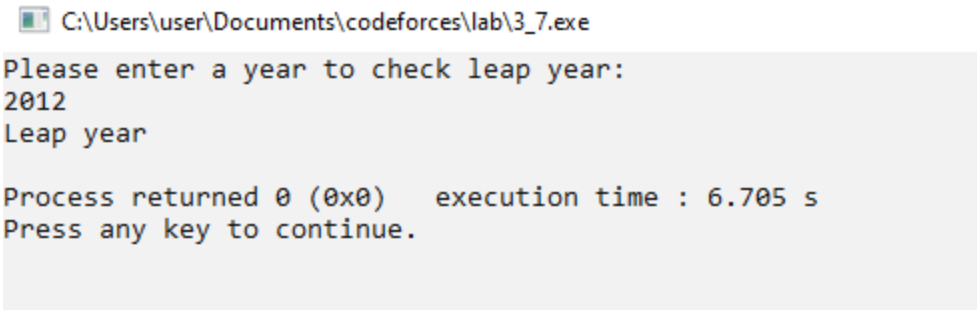


2.7**: C++ Program to check that a given year is a leap year or not.**

Code:

```
#include<iostream>
using namespace std;
int main()
{
   int num;
   cout<<"Please enter a year to check leap year: "<<endl;
   cin>>num;
   if(num%400 == 0)
   {
      cout<<"Leap year"<<endl;
   }
   else if(num%100 == 0)
   {
      cout<<"Not leap year"<<endl;
   }
   else if(num%4 == 0)
   {
      cout<<"Leap year"<<endl;
   }
   else
   {
      cout<<"Not leap year"<<endl;
   }
   return 0;
}
```

Output:

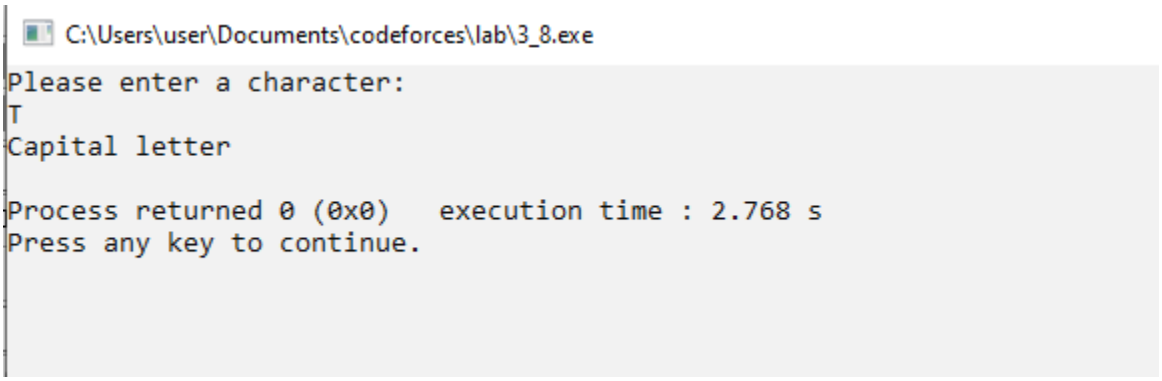


2.8: **C++ Program to input a character and check that it's a small letter, capital letter, a digit or a special symbol.**

Code:

```
#include<iostream>
using namespace std;
int main()
{
   char c;
   cout<<"Please enter a character: "<<endl;
   cin>>c;
   if(c >= 'A' && c<= 'Z')
   {
      cout<<"Capital letter"<<endl;

   }
   else if(c>= 'a' &&  c<='z')
   {
      cout<<"Small letter"<<endl;

   }
   else if(c>= '0' && c<= '9')
   {
      cout<<"Digit"<<endl;
   }
   else
   {
      cout<<"Special Symbol"<<endl;
   }
   return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- A little bit hard problem's.
- Being comfortable with the syntax.
- Solved some mid level problem .

Lab Class 3:

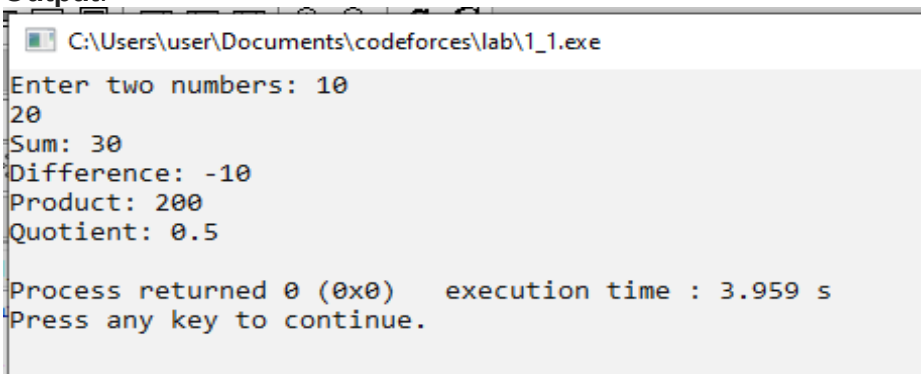
3.1:Write a C++ program that takes four functions that get two integers and returns its sum subtraction, multiplication, and division.

Code:

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

int sum(int a, int b){
   return a + b;
}

int subtract(int a, int b){
   return a - b;
}

int multiply(int a, int b){
   return a * b;
}

float divide(int a, int b){
   if(b==0){
      cout<<"Error! Division by zero is not possible.";
      return -1;
   }
   else{
      return float(a)/b;
```

```
    }
}

int main(){
    int a, b;
    cout<<"Enter two numbers: ";
    cin>>a>>b;

    cout<<"Sum: "<<sum(a,b)<<endl;
    cout<<"Difference: "<<subtract(a,b)<<endl;
    cout<<"Product: "<<multiply(a,b)<<endl;
    cout<<"Quotient: "<<divide(a,b)<<endl;

    return 0;
}
```

Output:



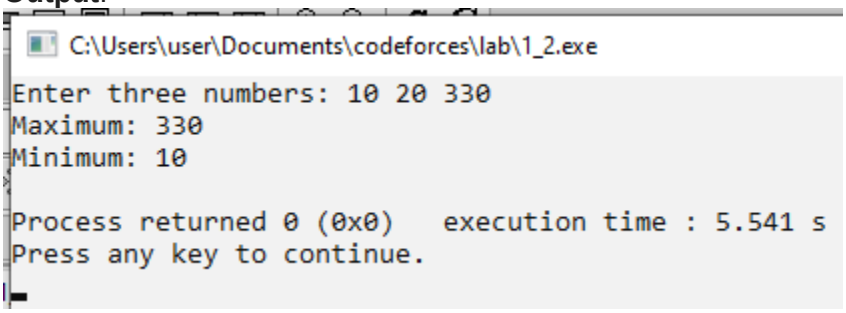
3.2: **Write a C++ program that take two functions that gets three integers and returns maximum and minimum.**

Code:

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

int max(int a, int b, int c){
    return (a>b)?((a>c)?a:c):((b>c)?b:c);
}

int min(int a, int b, int c){
    return (a<b)?((a<c)?a:c):((b<c)?b:c);
}

int main(){
    int a, b, c;
    cout<<"Enter three numbers: ";
    cin>>a>>b>>c;

    cout<<"Maximum: "<<max(a,b,c)<<endl;
```

```
    cout<<"Minimum: "<<min(a,b,c)<<endl;

    return 0;
}
```

Output:

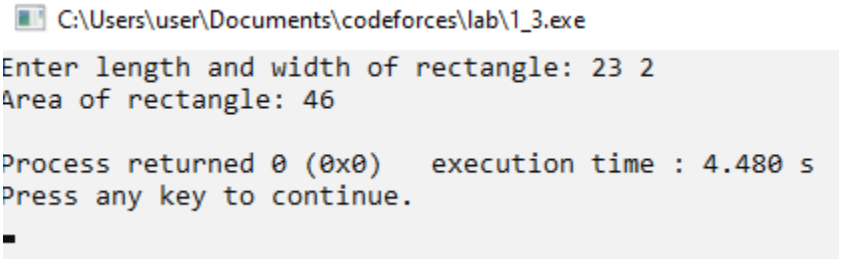


3.3: **Write a function that gets length and width of a rectangle and returns area**.

Code:

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

int area(int length, int width){
    return length*width;
}

int main(){
    int length, width;
    cout<<"Enter length and width of rectangle: ";
    cin>>length>>width;

    cout<<"Area of rectangle: "<<area(length, width)<<endl;

    return 0;
}
```

Output:

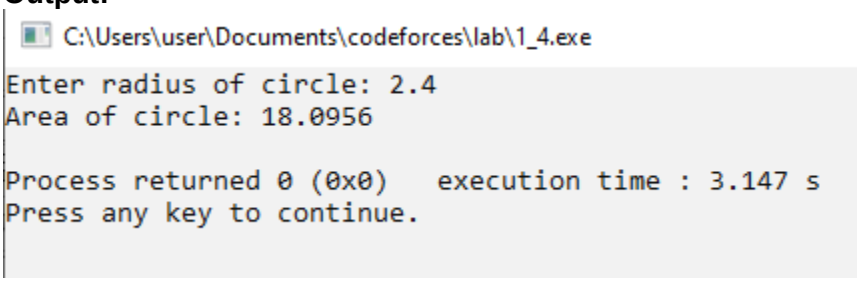


3.4: **Write a function that gets radius of a circle and returns area**.

Code:

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

float area(float radius){
   return 3.14159*radius*radius;
}

int main(){
   float radius;
   cout<<"Enter radius of circle: ";
   cin>>radius;

   cout<<"Area of circle: "<<area(radius)<<endl;

   return 0;
}
```

Output:



3.5: **Write a function that gets three numbers and returns medium.**

Code:

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

int middle(int a, int b, int c){
    if((a>b&&a<c)||(a<b&&a>c))
        return a;
    else if((b>a&&b<c)||(b<a&&b>c))
        return b;
    else
        return c;
}

int main(){
    int a, b, c;
    cout<<"Enter three numbers: ";
    cin>>a>>b>>c;

    cout<<"Middle number: "<<middle(a, b, c)<<endl;
```

```
    return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\1_5.exe
Enter three numbers: 20 45 2
Middle number: 20

Process returned 0 (0x0)   execution time : 8.058 s
Press any key to continue.
```

3.6**: Write a function that gets any positive number and returns its factorial**.

Code:

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

long long factorial(int num){
    if(num==0)
        return 1;
    else
        return num*factorial(num-1);
}

int main(){
    int num;
    cout<<"Enter a positive integer: ";
    cin>>num;

    cout<<"Factorial of "<<num<<" is "<<factorial(num)<<endl;

    return 0;
}
```

Output:

```
C:\Users\user\Documents\codeforces\lab\1_6.exe
Enter a positive integer: 7
Factorial of 7 is 5040

Process returned 0 (0x0)   execution time : 1.931 s
Press any key to continue.
```

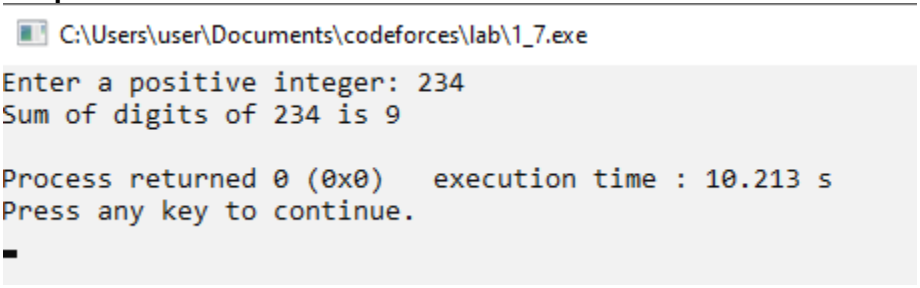
3.7:**Write a function that gets any positive number and returns sum of its digit**

Code:

```
#include<iostream>
```

```
using namespace std;

int sumOfDigits(int num){
    int sum = 0;
    while(num!=0){
        sum += num%10;
        num /= 10;
    }
    return sum;
}

int main(){
    int num;
    cout<<"Enter a positive integer: ";
    cin>>num;

    cout<<"Sum of digits of "<<num<<" is "<<sumOfDigits(num)<<endl;

    return 0;
}
```

Output:



3.8: **Write a function that gets any positive number and returns digital root.**

Code:

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

int digitalRoot(int num){
    if(num<10)
        return num;
    int sum = 0;
    while(num!=0){
        sum += num%10;
        num /= 10;
    }
    return digitalRoot(sum);
}

int main(){
    int num;
```

```
    cout<<"Enter a positive integer: ";
    cin>>num;

    cout<<"Digital root of "<<num<<" is "<<digitalRoot(num)<<endl;

    return 0;
}
```

Output:

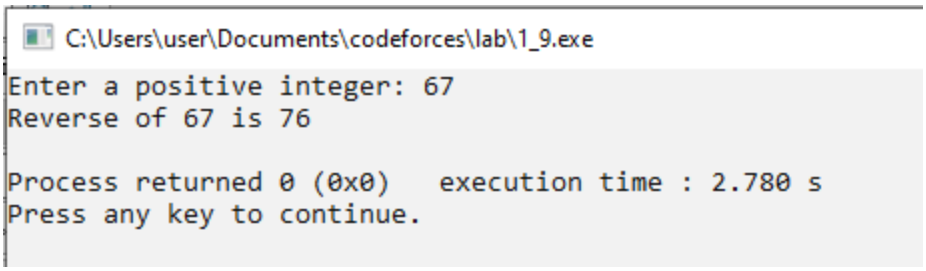
```
C:\Users\user\Documents\codeforces\lab\1_8.exe
Enter a positive integer: 56
Digital root of 56 is 2

Process returned 0 (0x0)   execution time : 3.392 s
Press any key to continue.
```

3.9: **Write a function that gets any positive integer and returns its reverse.**

Code:

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

int reverse(int num){
    int rev = 0;
    while(num!=0){
        rev = rev*10 + num%10;
        num /= 10;
    }
    return rev;
}

int main(){
    int num;
    cout<<"Enter a positive integer: ";
    cin>>num;

    cout<<"Reverse of "<<num<<" is "<<reverse(num)<<endl;

    return 0;
}
```

Output:

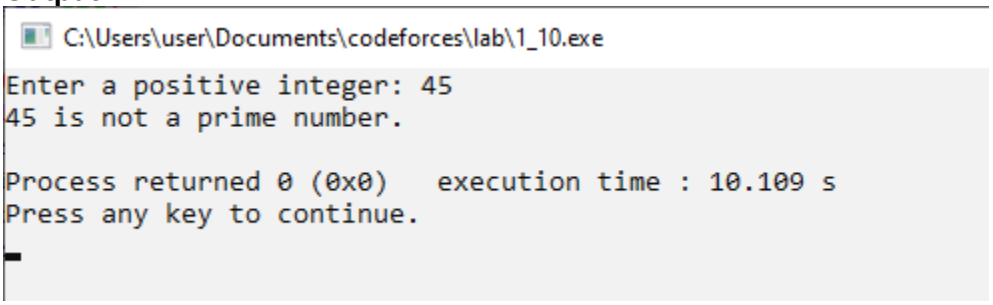


3.10: **C++ program to check prime number using user defined function.**

Code:

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

bool isPrime(int num){
    if(num<=1)
        return false;
    for(int i=2;i<=sqrt(num);i++){
        if(num%i==0)
            return false;
    }
    return true;
}

int main(){
    int num;
    cout<<"Enter a positive integer: ";
    cin>>num;

    if(isPrime(num))
        cout<<num<<" is a prime number."<<endl;
    else
        cout<<num<<" is not a prime number."<<endl;

    return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- How to use user defined function.
- Solved some basic problems using user defined function.
- How to return something from an user defined function.
- How to give parameter in an user defined function.

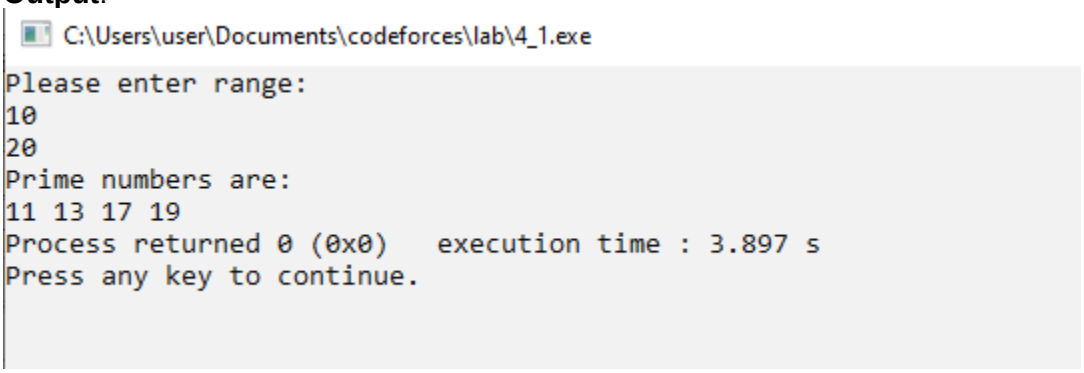
Lab Class 4:

4.1: C++ program to display prime numbers between two intervals using us defined function.

Code:

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

int primeCheck(int x){
    int flag  = 0;
    for(int i=2; i<=sqrt(x); i++){
        if(x%i == 0){
            return false;
        }
    }

    return true;

}
int primeInRange(){
    int a,b;
    cout<<"Please enter range: "<<endl;
    cin>>a>>b;
    cout<<"Prime numbers are: "<<endl;
    for(int i=a; i<=b; i++){
        if(primeCheck(i)){
            cout<<i<<" ";
        }
    }
}
int main(){
    primeInRange();
    return 0;
}
```

Output:

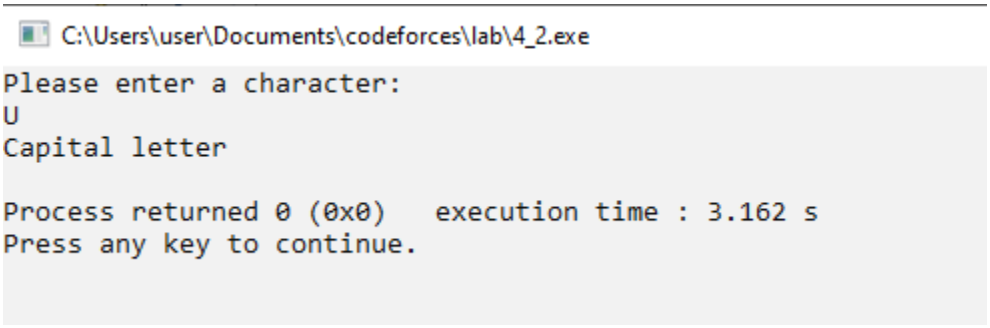


4.2**: Write a function that has one character argument and displays that it's a small letter, capital letter, a digit or a special symbol.**

Code:

```
#include<iostream>
using namespace std;
void checker(char c)
{
   if(c >= 'A' && c<= 'Z')
   {
      cout<<"Capital letter"<<endl;

   }
   else if(c>= 'a' &&  c<='z')
   {
      cout<<"Small letter"<<endl;

   }
   else if(c>= '0' && c<= '9')
   {
      cout<<"Digit"<<endl;
   }
   else
   {
      cout<<"Special Symbol"<<endl;
   }
}

int main()
{
   char ch;
   cout<<"Please enter a character: "<<endl;
   cin>>ch;
   checker(ch);
   return 0;
}
```

Output:



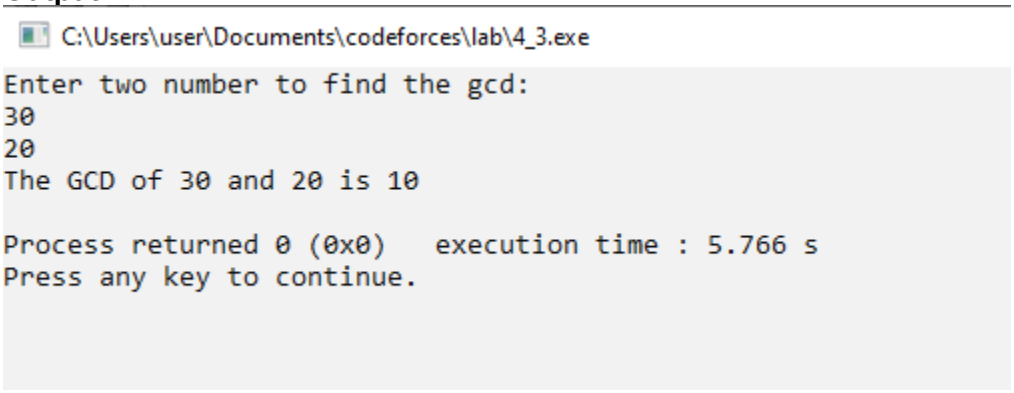
4.3**: Write a function that will find the GCD of two numbers. greatest comm Common divisor**

Code:

```
//Write a function that will find the GCD of two numbers. greatest Common divisor
#include<iostream>
using namespace std;
int findGcd(int x, int y)
{
   int smNum;
   if(x>y)
   {
      smNum = y;
   }
   else
   {
      smNum = x;
   }
   int gcdCandidate = 1;
   for(int i=1; i<=smNum; i++)
   {
      if(x%i == 0 && y%i==0)
      {
         gcdCandidate = i;
      }

   }
   return gcdCandidate;

}

int main()
{
   int a,b;
   cout<<"Enter two number to find the gcd: "<<endl;
   cin>>a>>b;
   int res = findGcd(a,b);
```

```
    cout<<"The GCD of " << a << " and " << b << " is "<<res<<endl;
    return 0;
}
```

Output:



4.4: **Write a program that will print Fibonacci series upto n value (using user defined function).**

Code:

```
//Write a program that will print Fibonacci series upto n value (using user defined function).
#include<iostream>
using namespace std;

int fiboSeries(int x)
{
    int a=0, b=1, nextNum = 0;
    while(nextNum<=x)
    {
        cout << nextNum << " ";
        a=b;
        b=nextNum;
        nextNum = a+b;
    }
}
int main()
{
    int n;
    cout<<"Please enter a number: "<<endl;
    cin>>n;
    cout<<fiboSeries(n)<<endl;
    return 0;
}
```

Output:

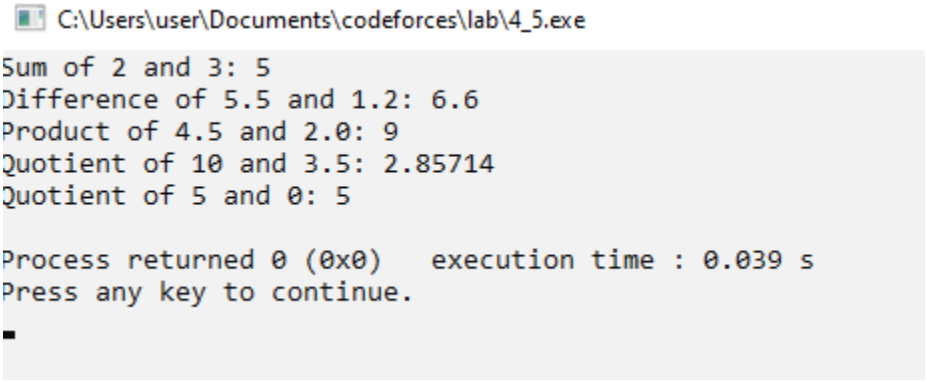
```
C:\Users\user\Documents\codeforces\lab\4_4.exe
Please enter a number:
45
0 1 1 2 3 5 8 13 21 34 55

Process returned 0 (0x0)   execution time : 1.832 s
Press any key to continue.
```

4.5: **Write a C++ program that take four functions that gets two values and returns its sum subtraction, multiplication, division using function overloading(data type different).**

Code:

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

int calc(int a, int b) {
    return a + b;
}

float calc(float a, float b) {
    return a - b;
}

double calc(double a, double b) {
    return a * b;
}

double calc(int a, double b) {
    if (b == 0) {
        cout << "Division by zero error!" << endl;
        return -1;
    }
    return a / b;
}

int main() {
    cout << "Sum of 2 and 3: " << calc(2, 3) << endl;
    cout << "Difference of 5.5 and 1.2: " << calc(5.5, 1.2) << endl;
    cout << "Product of 4.5 and 2.0: " << calc(4.5, 2.0) << endl;
    cout << "Quotient of 10 and 3.5: " << calc(10, 3.5) << endl;
    cout << "Quotient of 5 and 0: " << calc(5, 0) << endl;
    return 0;
}
```

Output:

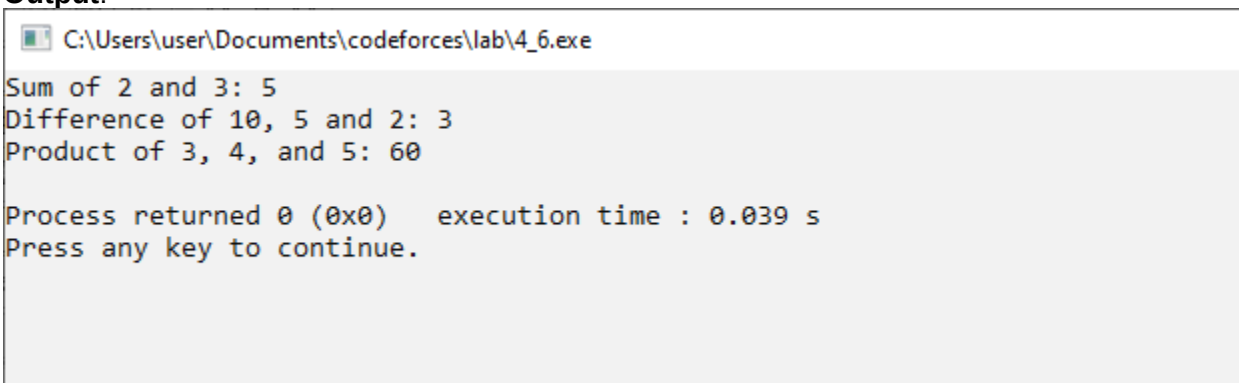


4.6: **Write a C++ program that take four functions that get integer values and returns ts sum, subtraction, multiplication, division using function overloading(number of argument different).**

Code:

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

int calc(int a, int b) {
    return a + b;
}

int calc(int a, int b, int c) {
    return a - b - c;
}

int calc(int a, int b, int c, int d) {
    return a * b * c * d;
}

float calc(int a, int b, float c) {
    if (c == 0) {
        cout << "Division by zero error!" << endl;
        return -1;
    }
    return (float)(a + b) / c;
}

int main() {
    cout << "Sum of 2 and 3: " << calc(2, 3) << endl;
    cout << "Difference of 10, 5 and 2: " << calc(10, 5, 2) << endl;
    cout << "Product of 3, 4, and 5: " << calc(3, 4, 5, 1) << endl;
    cout << "Quotient of 10 and 5, divided by 2: " << calc(10, 5, 2.0) << endl;
    cout << "Quotient of 10 and 5, divided by 0: " << calc(10, 5, 0.0) << endl;
    return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- The basic of function overloading.
- Some tricky problem solved like GCD, Fibonacchi.
- Used user defined function in several program

Lab Class 5:

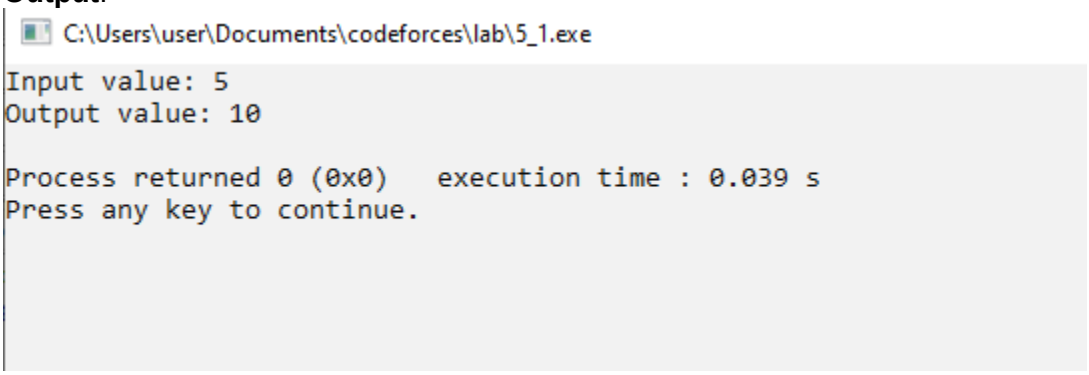
5.1**: Write a code for class that take an input and show that to output**

Code:

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

class Display {
public:
    void showInput(int input) {
        cout << "Input value: " << input << endl;
    }

    void showOutput(int output) {
        cout << "Output value: " << output << endl;
    }
};

int main() {
    Display displayObj;
    int input = 5, output = 10;

    displayObj.showInput(input);
    displayObj.showOutput(output);

    return 0;
```

}

Output:

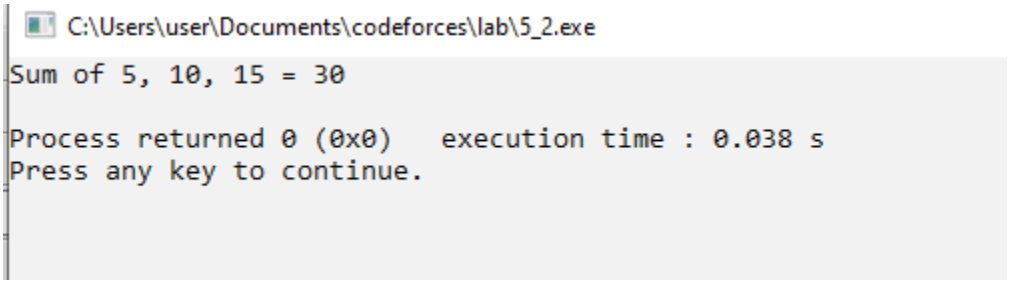


5.2: **Write a code for class that take three numbers as input and calculate summation of those numbers.**

Code:

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

class Sum {
public:
    int calculate(int a, int b, int c) {
        return a + b + c;
    }
};

int main() {
    Sum sumObj;
    int a = 5, b = 10, c = 15;

    int result = sumObj.calculate(a, b, c);
    cout << "Sum of " << a << ", " << b << ", " << c << " = " << result << endl;

    return 0;
}
```

Output:

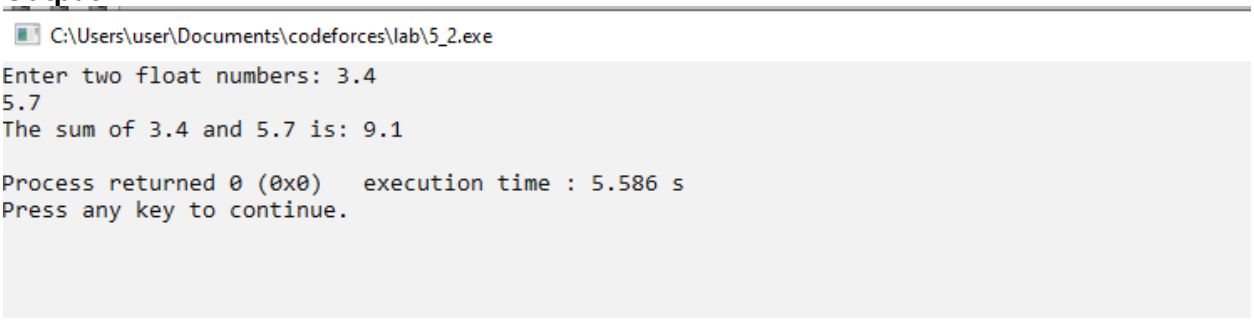


5.3: **Write a code for class that take two float as private data and calculate summation of those numbers**.

Code:

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

class Summation {
  private:
    float num1, num2;

  public:
    void setNumbers(float n1, float n2) {
      num1 = n1;
      num2 = n2;
    }

    float calculateSum() {
      return num1 + num2;
    }
};

int main() {
  Summation s;
  float num1, num2;

  cout << "Enter two float numbers: ";
  cin >> num1 >> num2;

  s.setNumbers(num1, num2);

  cout << "The sum of " << num1 << " and " << num2 << " is: " << s.calculateSum() << endl;

  return 0;
}
```

Output:



5.4: **Write a code for class that calculate average of N numbers.**

Code:

```
#include <iostream>
```

```
using namespace std;

class Average {
   private:
      float sum, average;
      int count;

   public:
      Average() {
         sum = 0;
         count = 0;
      }

      void addNumber(float num) {
         sum += num;
         count++;
      }

      float calculateAverage() {
         if(count != 0)
            average = sum / count;
         else
            average = 0;
         return average;
      }
};

int main() {
   Average a;
   int n;
   float num;

   cout << "Enter the value of N: ";
   cin >> n;

   for(int i=0; i<n; i++) {
      cout << "Enter number " << i+1 << ": ";
      cin >> num;
      a.addNumber(num);
   }

   cout << "The average of the " << n << " numbers is: " << a.calculateAverage() << endl;

   return 0;
}
```

Output:

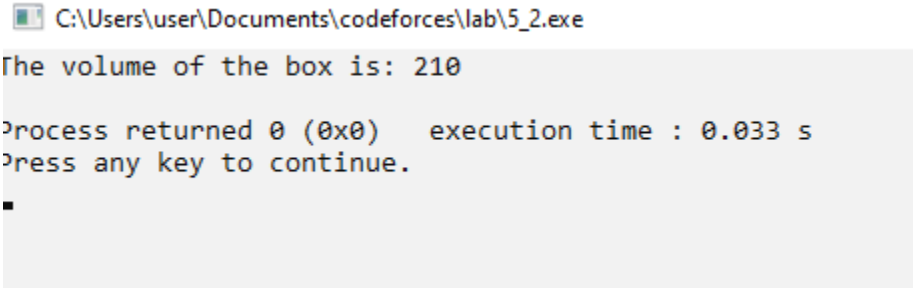
```
C:\Users\user\Documents\codeforces\lab\5_2.exe
Enter the value of N: 8
Enter number 1: 23
Enter number 2: 435
Enter number 3: 34
Enter number 4: 23
Enter number 5: 45
Enter number 6: 56
Enter number 7: 23
Enter number 8: 2
The average of the 8 numbers is: 80.125

Process returned 0 (0x0)   execution time : 11.681 s
Press any key to continue.
```

5.5: **Create a class called Box that contains one public function (print volume) and three public variables (height and length and width). Declare the height, length and width as integer value. Use public function print volume () to calculate the volume of a box and display the volume.**

Code:

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

class Box {
  public:
    int height;
    int length;
    int width;

    void printVolume() {
      int volume = height * length * width;
      cout << "The volume of the box is: " << volume << endl;
    }
};

int main() {
  Box b;

  b.height = 5;
  b.length = 6;
  b.width = 7;

  b.printVolume();

  return 0;
}
```

Output:

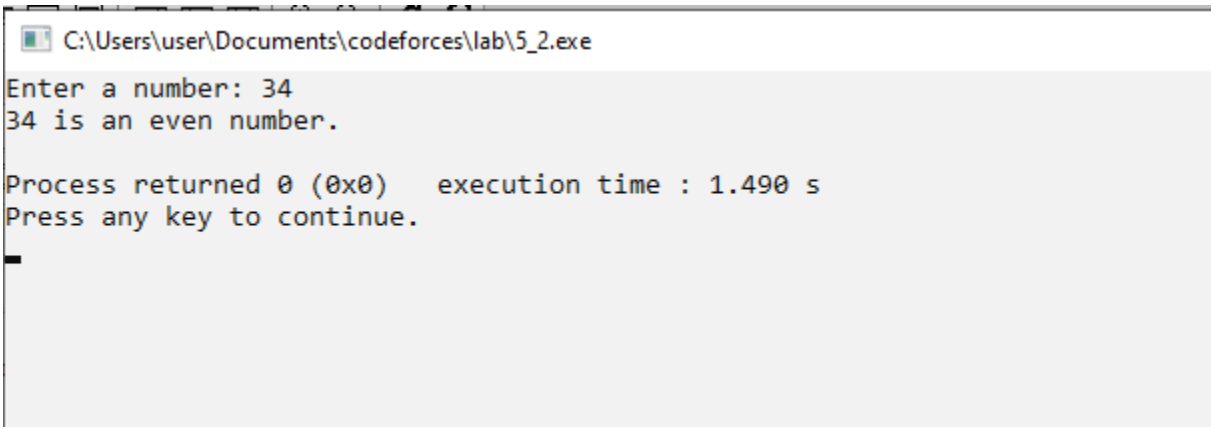


5.6: **Write a code for class which check a number even or odd..**

Code:

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

class EvenOrOdd {
   private:
      int num;

   public:
      void setNum(int n) {
         num = n;
      }

      void checkEvenOrOdd() {
         if(num % 2 == 0)
            cout << num << " is an even number." << endl;
         else
            cout << num << " is an odd number." << endl;
      }
};

int main() {
   EvenOrOdd eo;
   int n;

   cout << "Enter a number: ";
   cin >> n;

   eo.setNum(n);
   eo.checkEvenOrOdd();

   return 0;
}
```

Output:

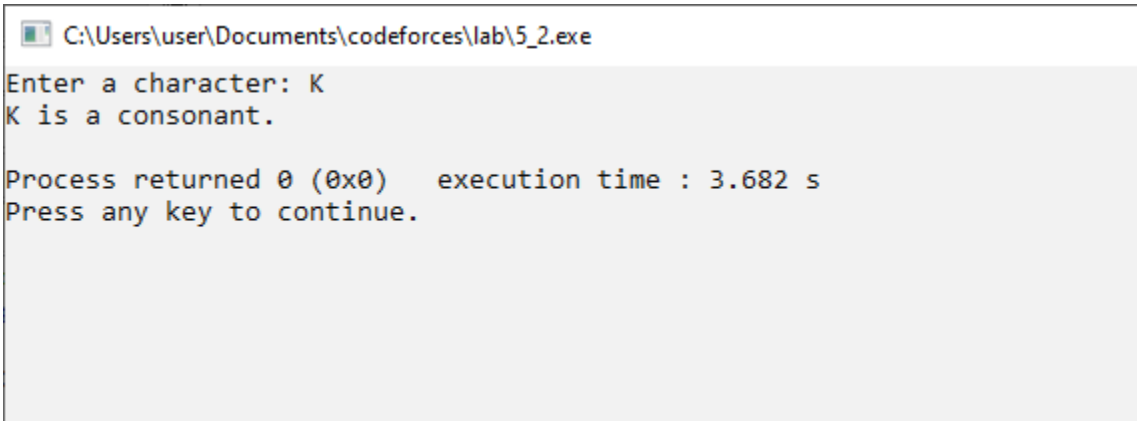


5.7: **Write a code for class which check a char vowel or consonant.**

Code:

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

class VowelOrConsonant {
   private:
      char ch;

   public:
      void setChar(char c) {
         ch = c;
      }

      void checkVowelOrConsonant() {
         if(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||
            ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U')
            cout << ch << " is a vowel." << endl;
         else
            cout << ch << " is a consonant." << endl;
      }
};

int main() {
   VowelOrConsonant voc;
   char c;

   cout << "Enter a character: ";
   cin >> c;

   voc.setChar(c);
   voc.checkVowelOrConsonant();

   return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- First time we knew about Class in C++
- Basic Syntax of class in c++
- Knew about pivate, protected and public.
- Solved some basic problem using class.

Lab Class 6:

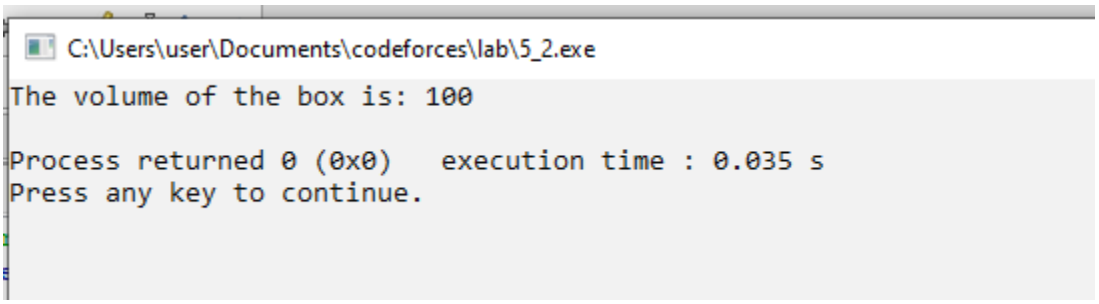
6.1: **Create a class called Box that contains one public function (print volume) and three public variables (height and length and width). Declare the height, length and width as integer value. Use public function print_volume () to calculate the volume of a box and display the volume**

Code:

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

class Box {
  public:
    int height;
    int length;
    int width;

    void print_volume() {
      int volume = height * length * width;
      cout << "The volume of the box is: " << volume << endl;
    }
```

```
};

int main() {
  Box b;

  b.height = 5;
  b.length = 10;
  b.width = 2;

  b.print_volume();

  return 0;
}
```

Output:



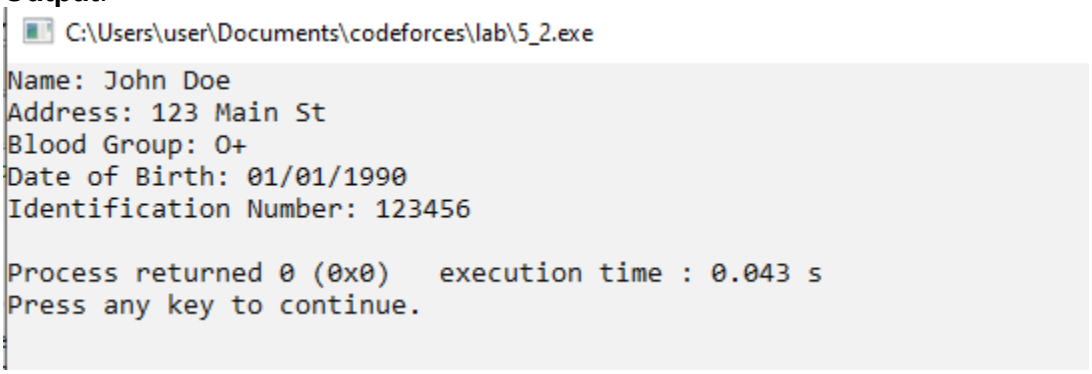
6.2: Create a class called person that maintains a register. In the register, it contains name and address and blood group, date of birth and identification number. Store the name, address and blood group as string and the identification number as an integer value. Use two public functions contain () to store the name, address, date of birth, blood group and identification number. And another public function display() will display the information.

Code:

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

class Person {
  private:
    string name;
    string address;
    string bloodGroup;
    string dateOfBirth;
    int idNumber;

  public:
    void contain(string n, string a, string b, string d, int i) {
      name = n;
      address = a;
```

```
        bloodGroup = b;
        dateOfBirth = d;
        idNumber = i;
    }

    void display() {
        cout << "Name: " << name << endl;
        cout << "Address: " << address << endl;
        cout << "Blood Group: " << bloodGroup << endl;
        cout << "Date of Birth: " << dateOfBirth << endl;
        cout << "Identification Number: " << idNumber << endl;
    }
};

int main() {
    Person p;

    p.contain("John Doe", "123 Main St", "O+", "01/01/1990", 123456);
    p.display();

    return 0;
}
```

Output:



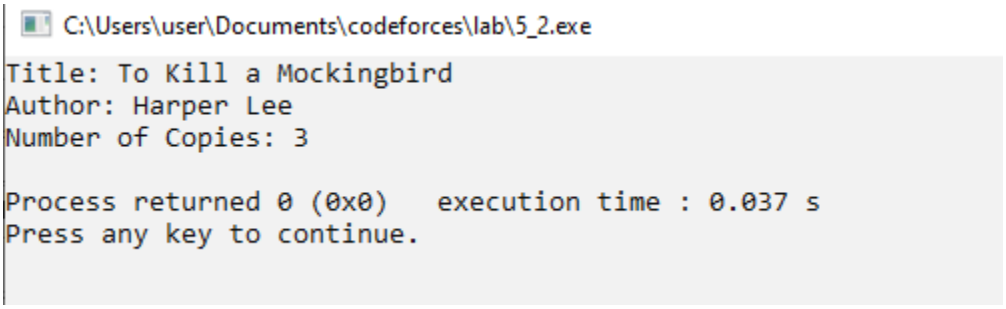
6.3: **Create a class called card that maintains a library card catalog entry. In the card, it will store a book's title, author and number of copies on hand. Store the title, author as string and the number of books on hand as an integer value. Use two public functions store () to store the title, author and number of books on hand. And another public function display() will display the information.**

Code:

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

class Card {
    private:
```

```
    string title;
    string author;
    int numCopies;

  public:
    void store(string t, string a, int n) {
      title = t;
      author = a;
      numCopies = n;
    }

    void display() {
      cout << "Title: " << title << endl;
      cout << "Author: " << author << endl;
      cout << "Number of Copies: " << numCopies << endl;
    }
};

int main() {
  Card c;

  c.store("To Kill a Mockingbird", "Harper Lee", 3);
  c.display();

  return 0;
}
```

Output:



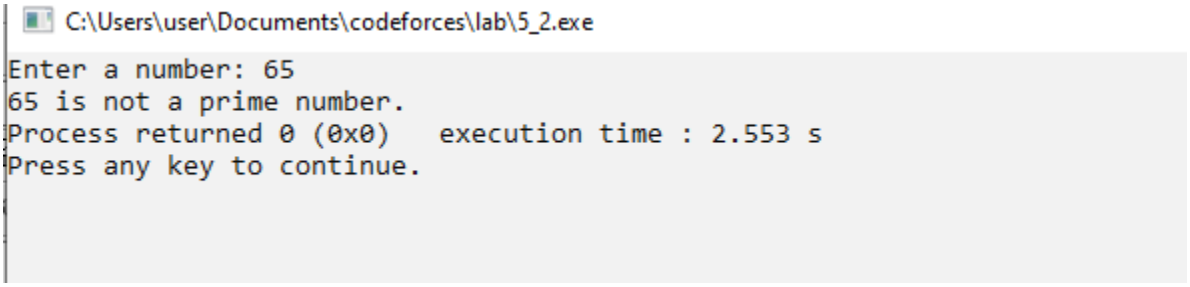
6.4: **Simple class program to find prime number using class prime. And take the input value as a private member and declare two public member function get_input() to get the input from user. And another function is calculate() to calculate the input is prime or not and display it using calculate () function.**

Code:

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

class Prime {
```

```
  private:
    int num;

  public:
    void get_input() {
      cout << "Enter a number: ";
      cin >> num;
    }

    void calculate() {
      bool isPrime = true;

      if (num <= 1) {
        isPrime = false;
      } else {
        for (int i = 2; i <= num / 2; ++i) {
          if (num % i == 0) {
            isPrime = false;
            break;
          }
        }
      }

      if (isPrime)
        cout << num << " is a prime number.";
      else
        cout << num << " is not a prime number.";
    }
};

int main() {
  Prime p;
  p.get_input();
  p.calculate();

  return 0;
}
```

Output:



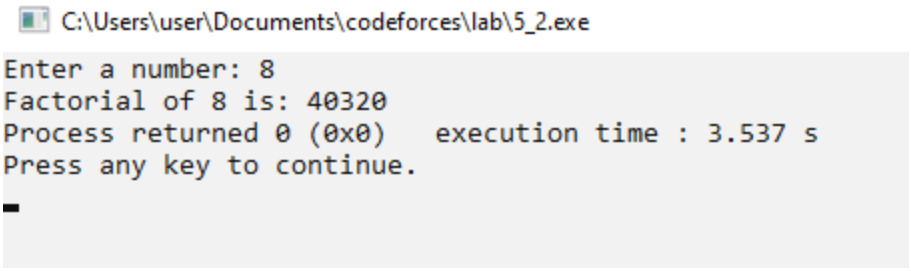
Lab Class 7:

7.1**: Simple class program to find factorial number using class factorial. And take the input value as a private member and declare two public member function get_input() to get the input from user. And another function is fact () to calculate the factorial of the input**

Code:

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

class Factorial {
  private:
    int num;

  public:
    void get_input() {
      cout << "Enter a number: ";
      cin >> num;
    }

    void fact() {
      int result = 1;

      for (int i = 1; i <= num; ++i) {
        result *= i;
      }

      cout << "Factorial of " << num << " is: " << result;
    }
};

int main() {
  Factorial f;
  f.get_input();
  f.fact();

  return 0;
}
```

Output:

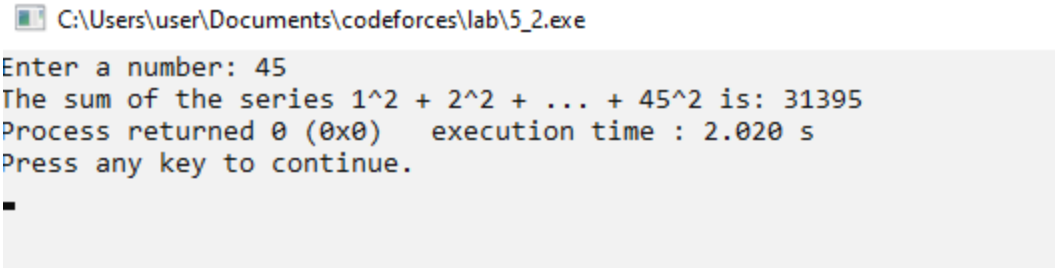


7.2: **Write a Class program to find out the sum of series 1^2+2^2+...+n^2.**

Code:

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

class SumOfSeries {
  private:
    int n;

  public:
    void get_input() {
      cout << "Enter a number: ";
      cin >> n;
    }

    void sum() {
      int result = 0;

      for (int i = 1; i <= n; ++i) {
        result += i * i;
      }

      cout << "The sum of the series 1^2 + 2^2 + ... + " << n << "^2 is: " << result;
    }
};

int main() {
  SumOfSeries s;
  s.get_input();
  s.sum();

  return 0;
}
```

Output:

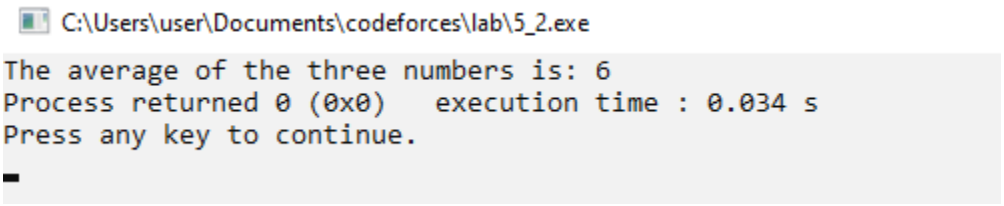


7.3: **A constructor function will input three numbers and destructor function will print the average of those numbers.**

Code:

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

class AverageCalculator {
  private:
    double num1, num2, num3;

  public:
    AverageCalculator(double n1, double n2, double n3) {
      num1 = n1;
      num2 = n2;
      num3 = n3;
    }

    ~AverageCalculator() {
      double average = (num1 + num2 + num3) / 3;
      cout << "The average of the three numbers is: " << average;
    }
};

int main() {
  AverageCalculator a(3, 6, 9);
  return 0;
}
```

Output:

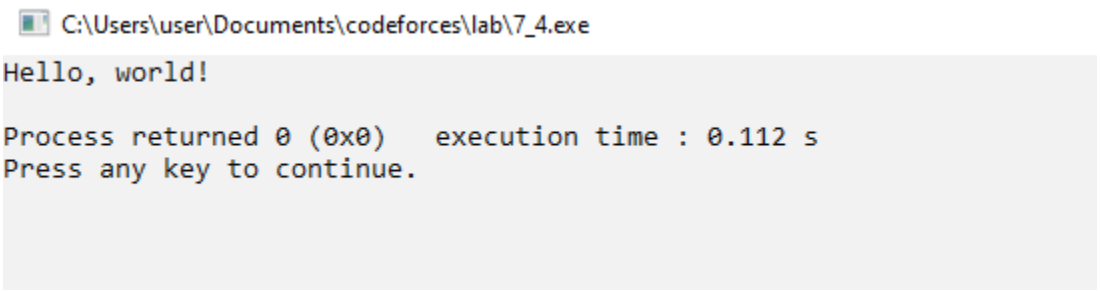


7.4: **Parent class will be par which have a string. Print the string with help of a child class named chil. 5. Base class have values of height & width of a triangle. Calculate the area of the triangle with help of a sub class.**

Code:

```
#include <iostream>
#include <string>

using namespace std;

class Par {
    protected:
        string str;

    public:
        Par(string s) {
            str = s;
        }
};

class Chil : public Par {
    public:
        Chil(string s) : Par(s) {}

        void print_string() {
            cout << str << endl;
        }
};

int main() {
    Chil chil_obj("Hello, world!");
    chil_obj.print_string();

    return 0;
}
```

Output:



Learnings of This Lab:

In this lab class we knew about:

- Implemented some complex solve using constructor and destructor.

CONCLUSION

In this whole course, we were lucky enough to learn new things which will impact in future. Specially, the OOP concept with C++ will need rest of our life and we learned very well about this concept for future.

In this lab, we have explored the basics of OOP in C++. We have learned how to define classes and objects, and how to use inheritance and polymorphism to create more complex class hierarchies. We have also seen how to use access modifiers to control the visibility of class members, and how to use constructors and destructors to manage object initialization and cleanup.

Overall, OOP provides a powerful and flexible approach to software development, enabling developers to create complex and maintainable software systems. With its rich set of language features, C++ is an excellent language for implementing OOP principles, and can be used to create high-performance, scalable, and robust software applications.