Institutional Training

June-July, 2021

**Name : Shubharthak Sangharsha**

**Class : Aiml-4-C**

**UID : 20BCS6872**

**Department of AIT-CSE**

**Daily Worksheet - Day-3**

**Functions and Operator Overloading Worksheet**

**Que: 1** Students should be able to write functions on their own to perform some specific task like

To read a matrix.

To find the power of a given number.

To find the largest of three numbers.

Also, Design and execute the main program to test the above functions

**Code:**

#include<iostream>

#include<math.h>

using namespace std;

class Function\_Overloading{

public:

void read(int m, int n){

int a[5][5];

cout<<"Enter elements of matrix: "<<endl;

for(int i = 0; i<m ; i++)

for(int j = 0;j<n; j++)

cin>>a[i][j] ;

for (int i =0; i < m ; i++){

for(int j = 0 ; j < n ; j++){

cout<<a[i][j]<< " " ;

}

}

}

void read(double x, double y) {

cout<<endl<<"Please enter two number, Second number will be power of first number: ";

cin >> x>> y;

double power = pow(x,y);

cout<< endl<<"Power: " <<power<<endl;

}

void read(int a, int b, int c){

if(a > b || a > c) {

cout<< "Largest is : "<<a<<endl;

}

else if(b > a || b> c){

cout<< "Largest is : "<<b<<endl;

}

else {

cout<<endl<< "Largest is : "<<c<<endl;

}

}

};

int main(){

Function\_Overloading obj1;

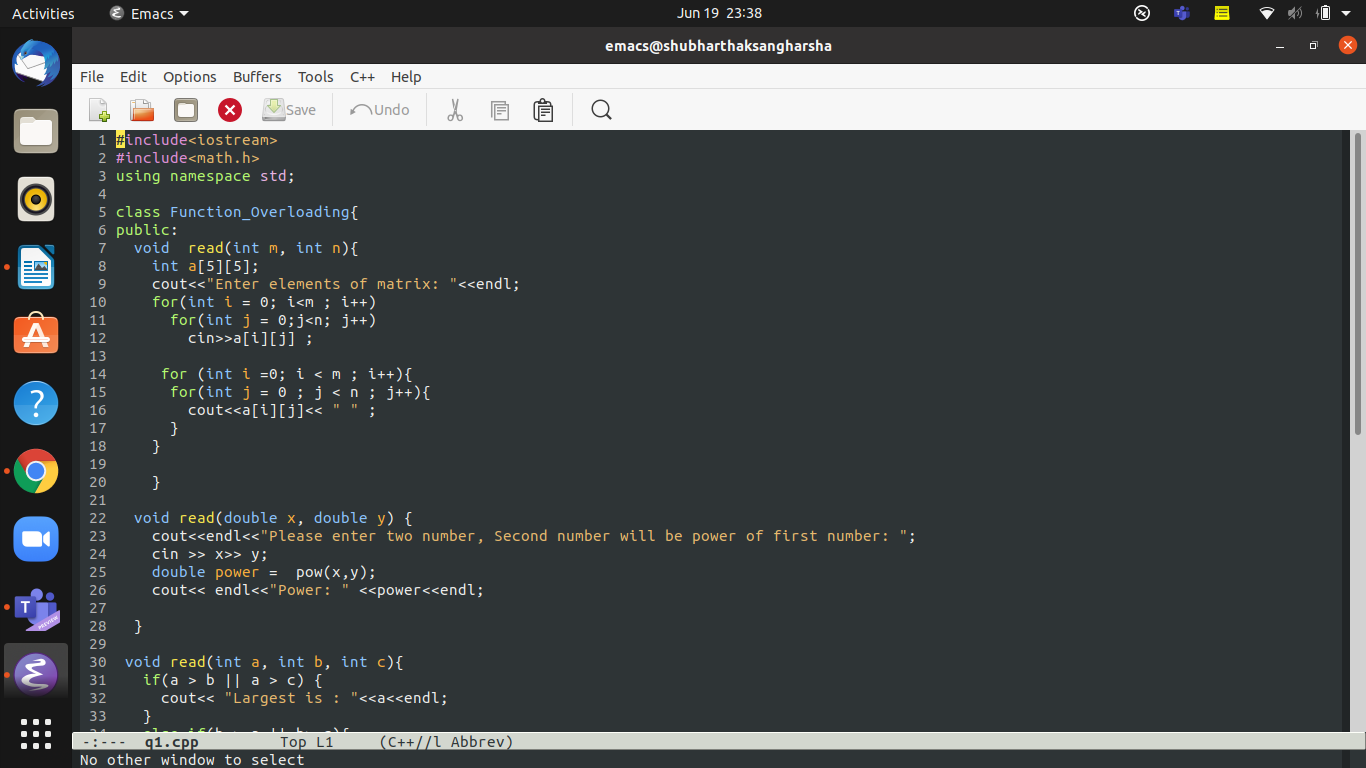
obj1.read(2,2);

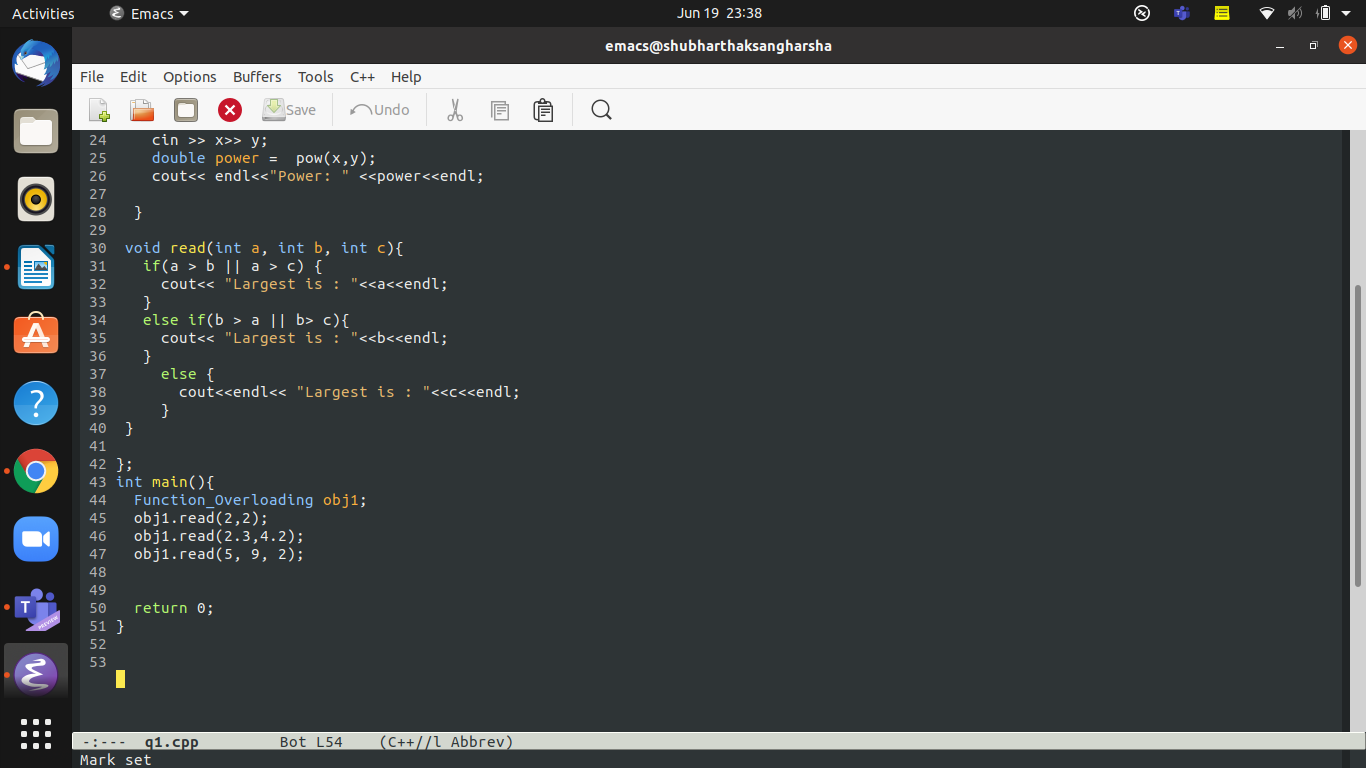
obj1.read(2.3,4.2);

obj1.read(5, 9, 2);

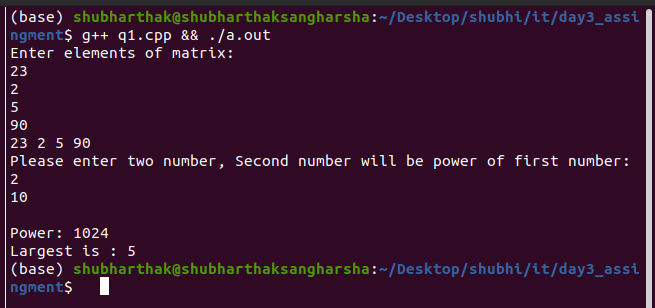
return 0;

}





**OUTPUT:-**

****

**Que: 2** Design and execute a program that substitutes an overloaded += operator for the overloaded + operator in the STRPLUS program in this chapter. This operator should allow statements like s1 += s2; where s2 is added (concatenated) to s1 and the result is left in s1. The operator should also permit the results of the operation to be used in other calculations, as in s3 = s1 += s2;

//By shubharthak

#include<iostream>

#include<string>

#include<cstring>

using namespace std;

class String //Class declaration and defination

{

private:

char str[110]; //class member

public:

void accept\_string() // member function for taking input

{

cout<<"\n Enter String: ";

cin>>str;

}

void display\_string()

{

cout<<str;

}

String operator+(String x) // overloading + operator using member function

{

String s; //stores the concatinated string

strcat(str,x.str); // concatinate the obj1 string and obj2 string using strcat

strcpy(s.str,str); // copy the string concatinated value into s

return s; // return the concatinated value back to the main function

}

};

//Driver Code

int main()

{

String str1, str2, str3;

str1.accept\_string(); //taking input string from obj1

str2.accept\_string(); //taking input string from obj2

cout<<"\n ----------------------------------------------";

cout<<"\n\n First String is: ";

str1.display\_string(); //Displaying First String

cout<<"\n\n Second String is: ";

str2.display\_string(); //Displaying Second String

cout<<"\n ----------------------------------------------";

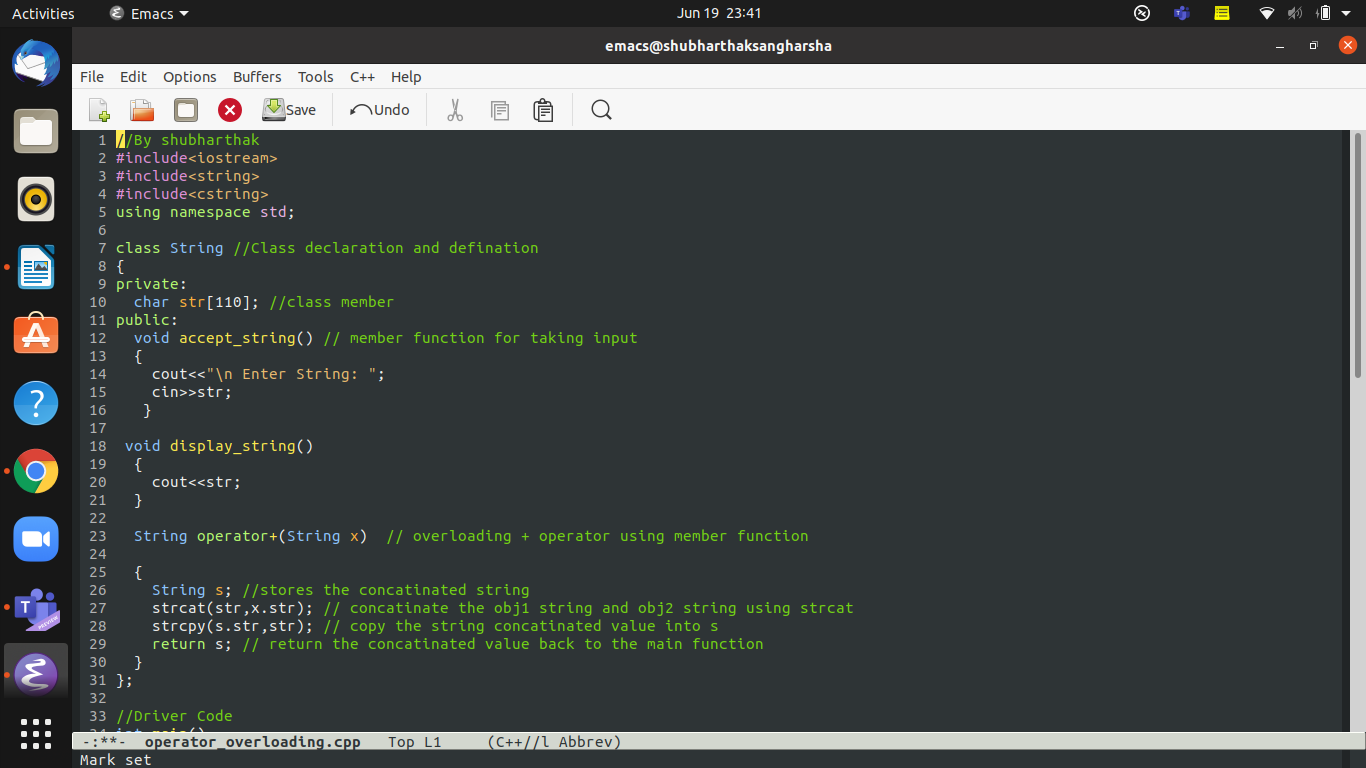
str3=str1+str2; //String is concatenated. Overloaded '+' operator

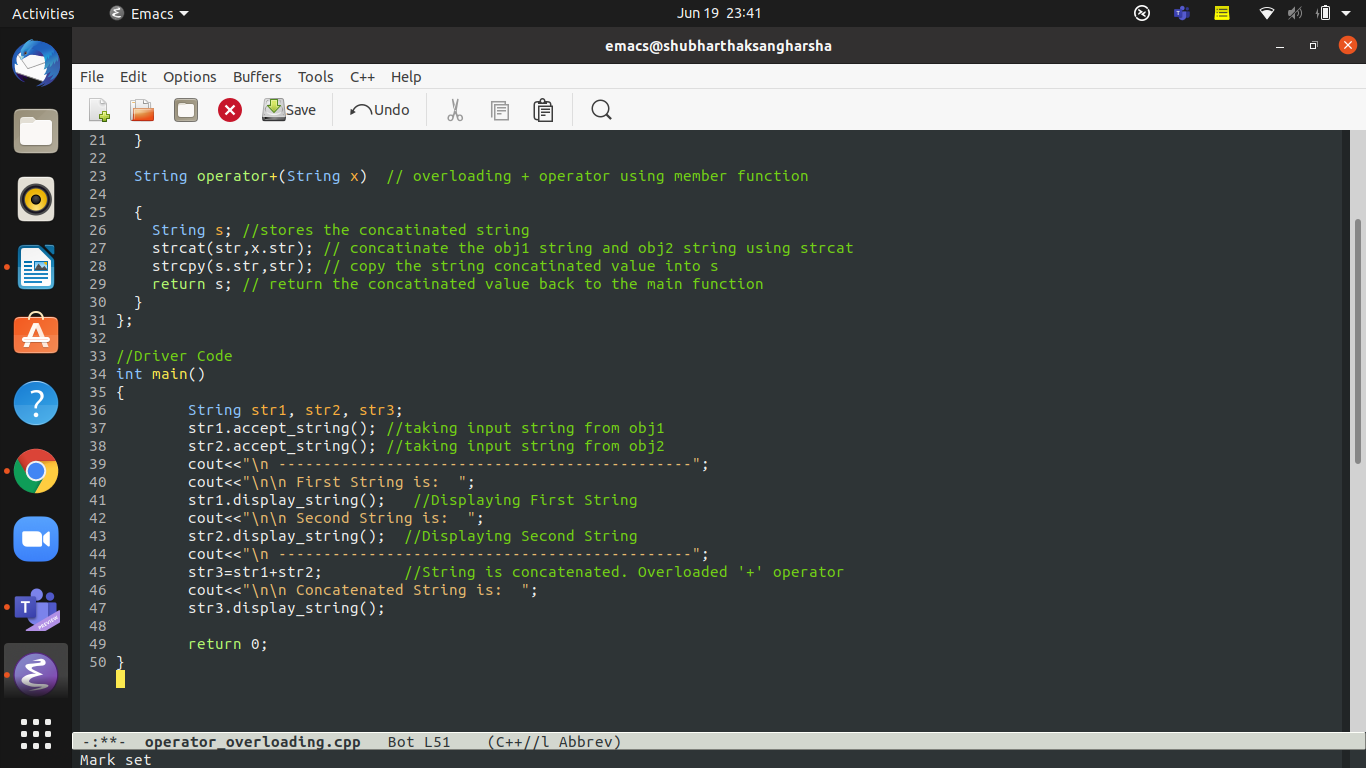
cout<<"\n\n Concatenated String is: ";

str3.display\_string();

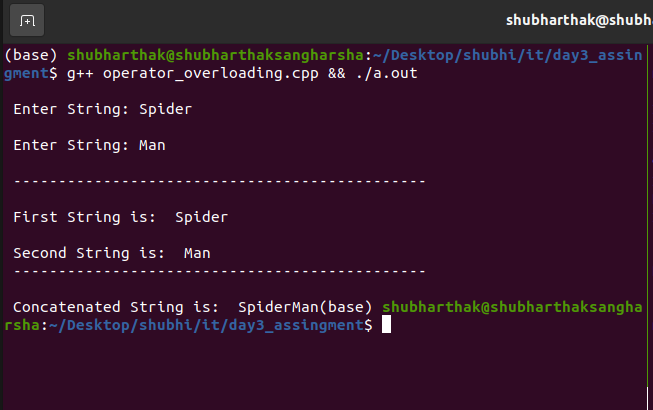
return 0;

}



****

**OUTPUT:-**

****

**Que: 3** Write a program that calculates the total point earned by a football team over a series of games and calculate the average of team. It allow the user to enter the series of game points, then -1 to signal the end of the list.

**CODE:-**

//By Shubharthak

#include<iostream>

#include<math.h>

using namespace std;

class Football{

public:

void calculate(){

int sum=0;

int count=0;

while(1){

cout<<"Count before : " << count <<endl;

cout<<"enter the score in "<<++count<<" match"<<endl;

cout<<"Count before : " << count <<endl;

int score;

cin>>score;

if(score==-1){

break;

}

sum+=score;

}

cout<<"Total points in the match is "<<sum<<endl;

cout<<"average points of series is "<<sum/(count-1)<<endl;

}

};

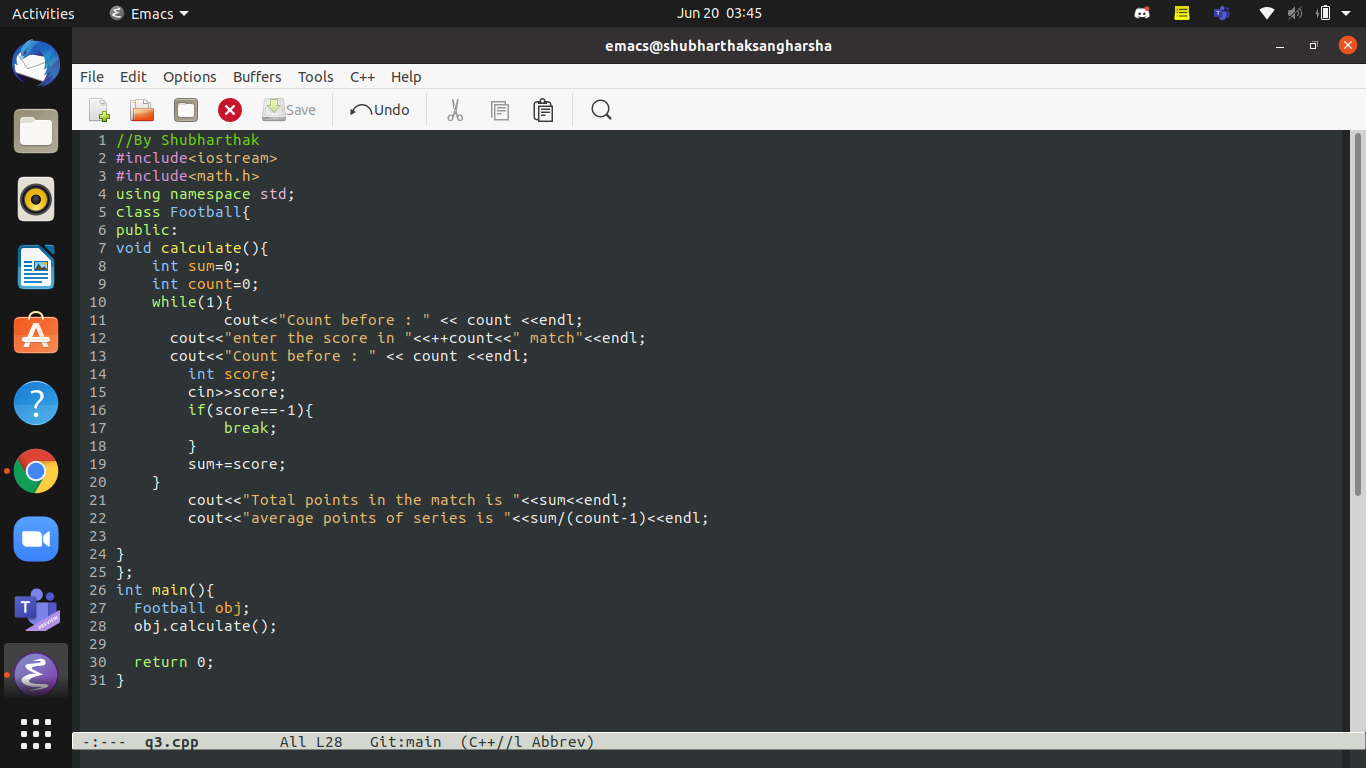
int main(){

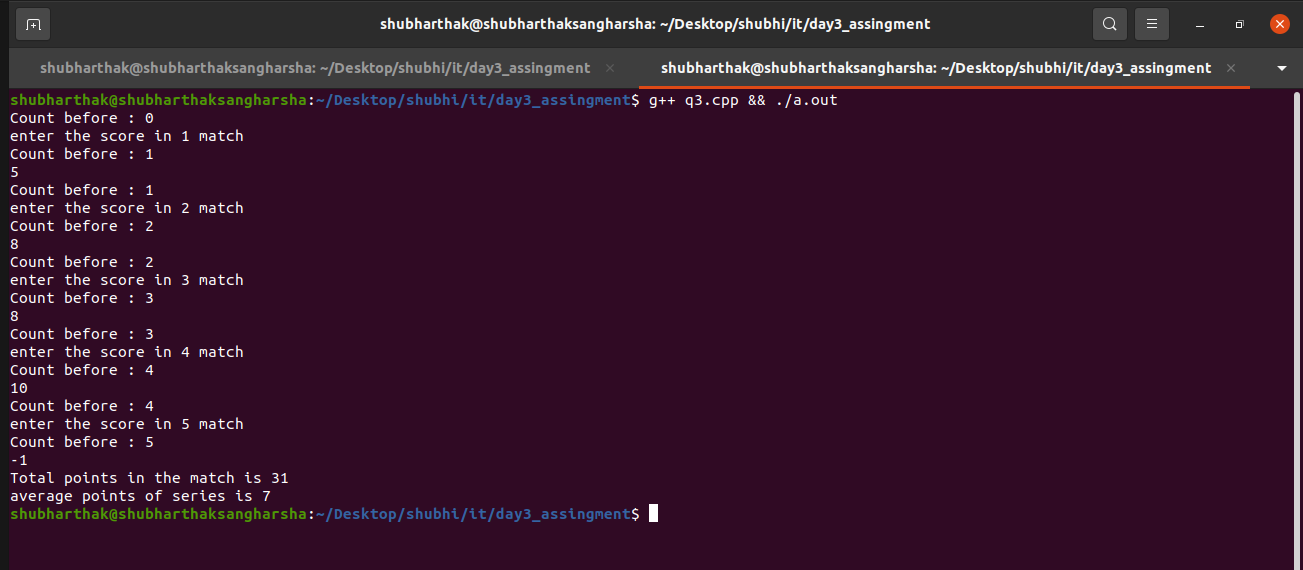
Football obj;

obj.calculate();

return 0;

}

**OUTPUT:-**



**Que: 4** Write a function named "g\_c\_d" that takes two positive integer arguments and returns as its value the greatest common divisor of those two integers. If the function is passed an argument that is not positive (i.e., greater than zero), then the function should return the value 0 as a sentinel value to indicate that an error occurred.

**CODE:-**

#include <iostream>

using namespace std;

class Gcd{

public:

int g\_c\_d (int a, int b)

{

if (a <= 0 || b <= 0) // a parameter is not positive

return 0; // exit and return the error sentinel

int trial\_divisor;

trial\_divisor = ( a <= b ? a : b ); // set it to the smaller

while (a % trial\_divisor != 0 || b % trial\_divisor != 0)

--trial\_divisor;

return trial\_divisor;

}

};

int main(){

Gcd obj;

int n1, n2 ;

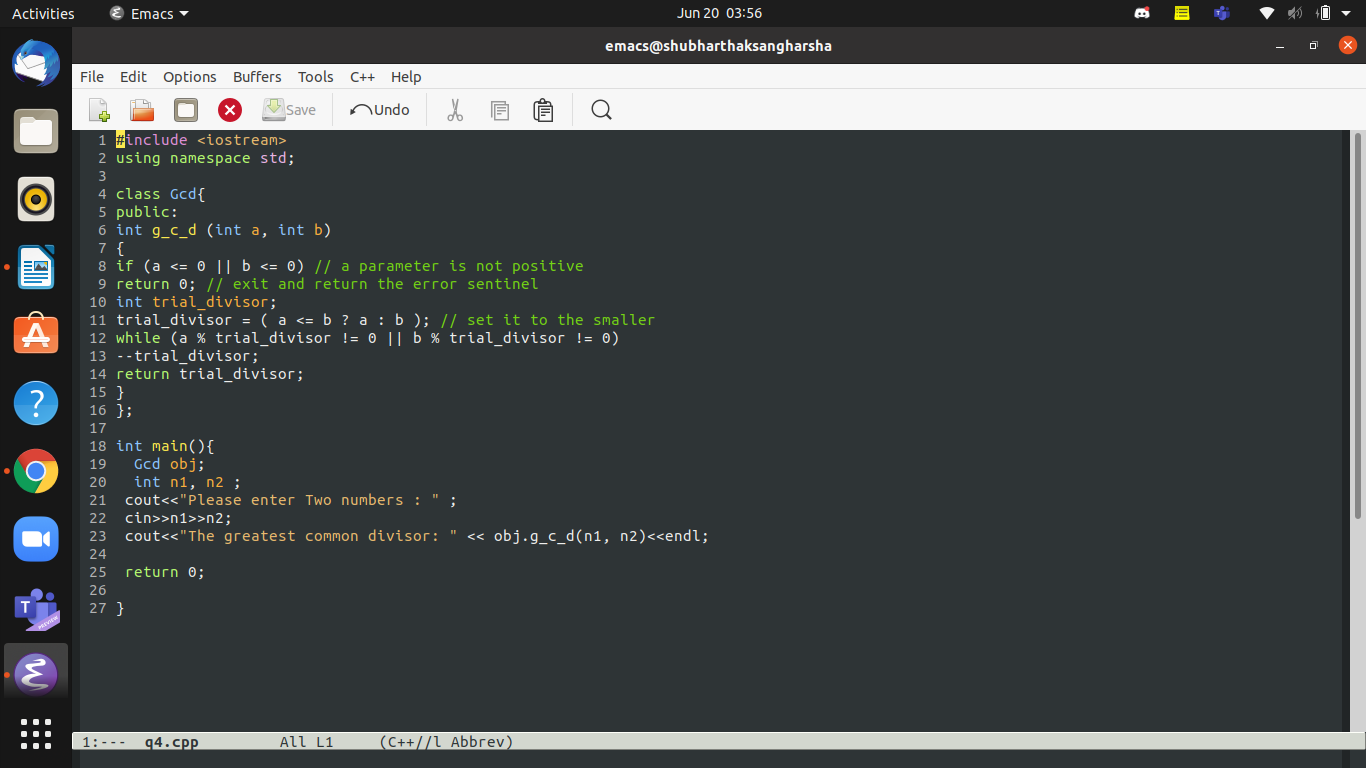
cout<<"Please enter Two numbers : " ;

cin>>n1>>n2;

cout<<"The greatest common divisor: " << obj.g\_c\_d(n1, n2)<<endl;

return 0;

}

**OUTPUT:-**

