LAB FILE FOR   
OBJECT- ORIENTED PROGRAMMING LAB[CS-214]

Jyoti Srivastava

SUBMMITED BY:

RISHABH GANGWAR  
195103

|  |  |  |
| --- | --- | --- |
| SRNO. | TITLE | DATE |
| 1 | **A:Write a program that reads two nos. from keyboard and gives their addition, subtraction, multiplication, division and modulo. User will enter two operands and choice of operation based on that user will get the result**  B: Write a function which will have an argument n and will print all the factors of n. Example: factors for 28-> 1, 2, 4, 7, 14, 28 Write the program to use above function | 2-SEP-2020 |
| 2 | **Demonstrate use of array in C++:**  **a) Write a program to perform matrix addition. Dimensions of the matrices will be entered by  the user.**  **b) Write a program to perform binary search in an array.**  **c) Write a program which takes string as an input and counts numbers of occurrences of each  vowel and total number of vowels in the string. (Use strings as arrays and to access array  elements using pointers).** | 10-SEP-20 |
| 3 | **Demonstrate use of structure and union.**   1. **Define a structure type, personal, that would contain person name, date of joining and  salary. Using this structure, write a program to read this information for one person from  the keyboard and print the same on the screen. Define a Union for the same members and  compare the memory occupied by both (structure and union).**   **b) Write a function to enter rollno, marks of the three subject for 3 students and find total  obtained by each student using structure.** | 17-SEP-20 |
| 4 | **Demonstrate use of function in C++**  **a) Write a function to swap two numbers using call by reference.**  **b) Write a function to check whether given number is prime number or not.**  **c) Write a function to find the multiplication values and the cubic values using inline  function.**  **d) Write a function to calculate the area of circle, rectangle and triangle using function  overloading** | 24-SEP-20 |
| 5 | **a) Define a Class “Student” having following data members and member functions:  ∙ Data members:**  **1. Name of the Student**  **2. Roll number**  **3. Department Name**  **∙ Member functions:**  **1. Enter Student Details**  **2. Display Student Details**  **Write a main program to create the object and test the program for 5 students.**  **b) Define a Class “bank\_account” having following data members and member functions:  ∙ Data members:**  **1. Name of the depositor**  **2. Account number**  **3. Type of account**  **4. Balance amount in the account**  **∙ Member functions:**  **1. To assign the initial values**  **2. To deposit an amount**  **3. To withdraw an amount after checking the balance**  **4. To display name and balance** | 1-OCT-2020 |
| 6 | **Demonstrate use of friend function with class in C++**  **a) Create two classes DM and DB which store the value of distances. DM stores distance in  meters and centimeters and DB in feet and inches. Write a program that can read values  for the class objects and add one object of DM with another object of DB by keeping  following in mind.**  **i. Use friend function to carry out the addition.**  **ii. The object that stores the results may be a DM or DB object, depending on  the unit in which the results are required.**  **b) Write a program to calculate number of object created for particular class.** | 15-OCT-20 |
| 7 | **Demonstrate operator overloading in C++**  **a) Create a class coordinate containing x, y and z private variables. Perform operations for  incrementing, decrementing, adding and comparing object(s) by overloading ++, --, += and == operators respectively. Define necessary functions to set and display the variables** | 22-OCT-20 |
| 8 | **Demonstrate different kind of inheritance in C++**  **a. Implement following class relationship and test with main class.**  **Staff(code,name)**  **Teacher Typist Officer**  **(subject, Publication) (Speed) (grade)**    **Regular Casual (daily wages)**  **b. Add education details for above classes except for typist. This class will include two  information namely, highest education in general studies and highest professional  qualification.**  **c. Implement the following class using abstract class.**  **A. Shape**  **i. TwoDimensional**  **1. Triangle**  **2. Rectangle**  **3. Circle**  **ii. ThreeDimensional**  **1. Box**  **2. Cone**  **3. Cylinder**  **4. Sphere**  **Use the concept of virtual function for the following:**  **∙ Write a function find\_area() to calculate area of 2-D objects.**  **∙ Write a function find\_volume() to calculate the volume of 3-D objects. ∙ Write a function to display area and volume of corresponding objects.**  **∙ In main function call these functions from base classes. (Hint: Call display() from  object of shape class. Call find\_area() from object of TwoDimensional class. Call  find\_volume() from object of ThreeDimensional class).** | 22-OCT-20 |
| 9 | **Write a program that shows that two files named ‘Source 1’ and ‘Source 2’ contains sorted list of integers. Read the content of both the files and stores the merged list in sorted form in a new file named ‘Target** | 29-OCT-20 |
| 10 | **Write a function template to perform linear search in an array** | 29-OCT-20 |
| 11 | **Demonstrate exception handling in C++**  **Write a program with the following:**  **a. A function to read two double type numbers from keyboard**  **b. A function to calculate the division of these two numbers**  **c. A try block to throw an exception when wrong type of data is keyed in d. A try block to detect and throw an exception if the condition “divide-by-zero” occurs**  **e. Appropriate catch block to handle the exception thrown.** | 5-NOV-20 |
| 12 | **A table gives a list of car models, its cost and the number of units sold in each type in a specified  period. Write a program to store this table in suitable container and to display interactively the  total value of a particular model sold, given the unit-cost of that model.** | 12-NOV-20 |
| 13 | **Write a class template to represent a generic vector. Include the member functions to perform the  following tasks:**  **a. To create a vector**  **b. To modify the value of given element**  **c. To multiply by a scalar value.** | 12-NOV-20 |

**EXPERIMENT-1**

**Write a program that reads two nos. from keyboard and gives their addition, subtraction, multiplication, division and modulo. User will enter two operands and choice of operation based on that user will get the result**

**A:**

#include <iostream>

using namespace std;

int main()

{ int input,T;

double A,B;

cout<<"Enter the number of test case:"<<endl;

cin>>T;

while(T--){

cout<<"Enter your choice of operation:"<<endl;

cout<<"1.addition\n2.substraction\n3.multiplication\n4.division\n5.modulo\n";

cin>>input;

cout<<"Enter your two numbers:"<<endl;

cin>>A>>B;

if(B==0) {cout<<"cannot divide by 0";

return 0;}

if(input==1)

cout<<"addition = "<<A+B;

else if(input==2)cout<<"substraction = "<<A-B;

else if(input==3)cout<<"multiplication = "<<A\*B;

else if(input==4)cout<<"quotient = "<<A/B;

else if(input==5){

int modB=B;

int modA=A;

cout<<"modulo = "<<modA%modB;

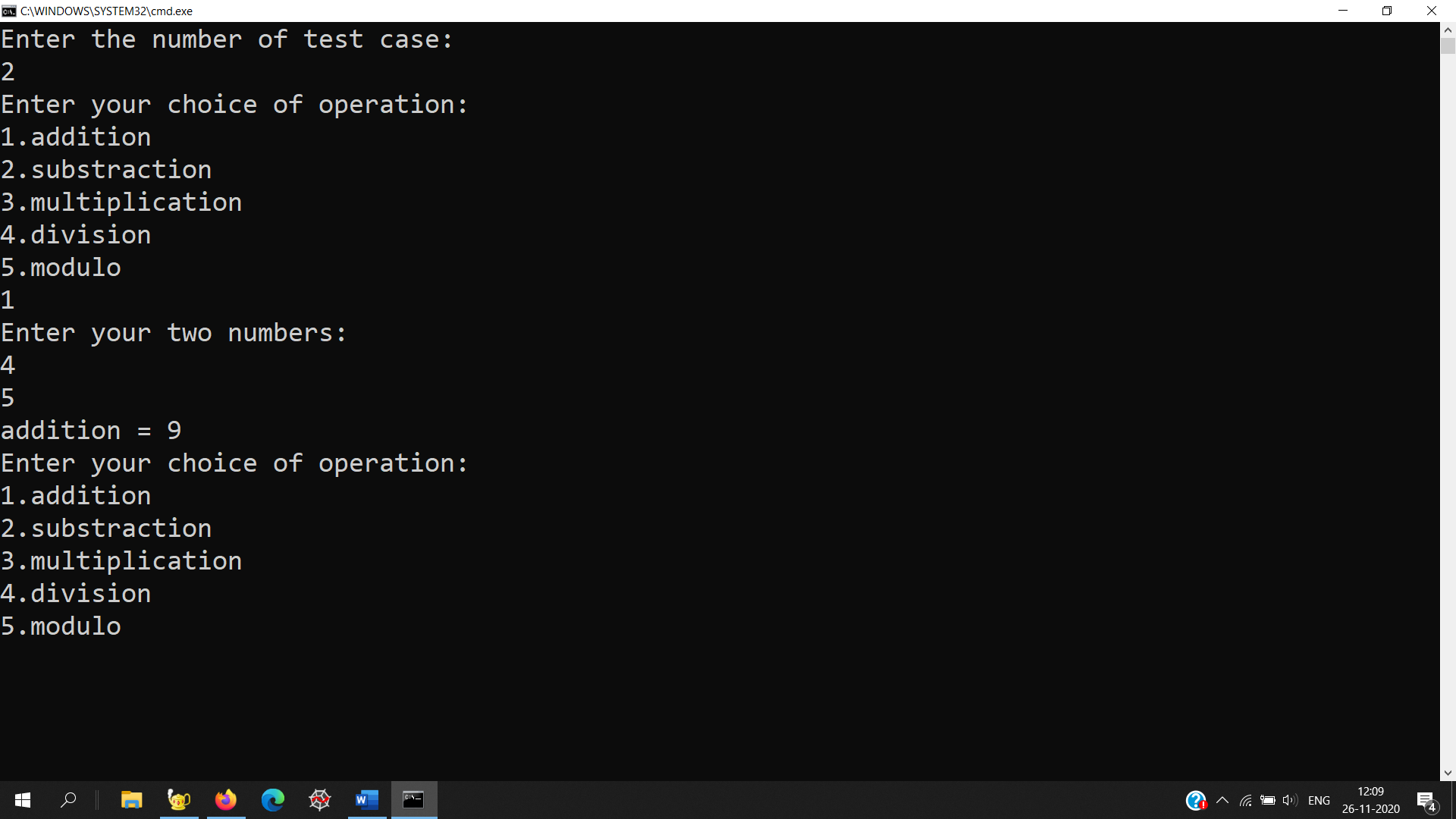
}

cout<<endl;

}

return 0;

}



# B:

# Write a function which will have an argument n and will print all the factors of n. Example: factors for 28-> 1, 2, 4, 7, 14, 28 Write the program to use above function

# #include <iostream>

# using namespace std;

# void divisors(int n)

# {

# int i;

# for(i=1;i\*i<n;i++)

# {

# if((n%i)==0)

# cout<<i<<endl;

# }

# for(i=i-1;i>=1;i--)

# {

# if((n%i)==0)

# cout<<n/i<<endl;

# }

# }

# int main()

# {

# int T;

# cout<<"Enter the number of testcase:\n";

# cin>>T;

# while(T--)

# { int N;

# cout<<"Enter the number of which divisors are to be found:\n";

# cin>>N;

# divisors(N);

# }

# }

# 

**EXPERIMENT-2**

**Demonstrate use of array in C++:**

**a) Write a program to perform matrix addition. Dimensions of the matrices will be entered by  the user.**

**b) Write a program to perform binary search in an array.**

**c) Write a program which takes string as an input and counts numbers of occurrences of each  vowel and total number of vowels in the string. (Use strings as arrays and to access array  elements using pointers).**

# A:

# #include <iostream>

# using namespace std;

# int binarySearch(int arr[], int left, int right, int x)

# {

# if (right >= left) {

# int mid = left + ((right - left) / 2);

# 

# if (arr[mid] == x)

# return mid;

# if (arr[mid] > x)

# return binarySearch(arr, left, mid - 1, x);

# 

# return binarySearch(arr, mid + 1, right, x);

# }

# return -1;

# }

# int main()

# { int n;

# cout<<"Enter the number of the total number of elements:"<<endl;

# cin >> n;

# int arr[n];

# cout<<"Enter the number of the all elements:"<<endl;

# for(int i=0;i<n;i++)

# cin>>arr[n];

# cout<<"Enter the element you want search:"<<endl;

# int f;

# cin>>f;

# cout<<binarySearch(arr,0,n-1,f);

# 

# }

# 

# 

# B:

# #include <iostream>

# using namespace std;

# int main()

# {

# int r, c;

# cout<<"Enter the rows and columns of the Matrix:"<<endl;

# cin >> r;

# cin >> c;

# 

# int a[r][c], b[r][c], sum[r][c], i, j;

# cout << endl << "Enter elements of 1st matrix: " << endl;

# for(i = 0; i < r; ++i)

# for(j = 0; j < c; ++j)

# {

# cout << "Enter element a" << i + 1 << j + 1 << " : ";

# cin >> a[i][j];

# }

# cout << endl << "Enter elements of 2nd matrix: " << endl;

# for(i = 0; i < r; ++i)

# for(j = 0; j < c; ++j)

# {

# cout << "Enter element b" << i + 1 << j + 1 << " : ";

# cin >> b[i][j];

# }

# for(i = 0; i < r; ++i)

# for(j = 0; j < c; ++j)

# sum[i][j] = a[i][j] + b[i][j];

# // Displaying the resultant sum matrix.

# cout << endl << "Sum of two matrix is: " << endl;

# for(i = 0; i < r; ++i)

# for(j = 0; j < c; ++j)

# {

# cout << sum[i][j] << " ";

# if(j == c - 1)

# cout << endl;

# }

# return 0;

# }

# 

# C:

# #include <iostream>

# using namespace std;

# int main()

# {

# int i=0,cA=0,cE=0,cI=0,cO=0,cU=0;

# char vowels[200];

# cout << "Enter the string: ";

# for (i = 0; (\*(vowels+i)) != '\0'; i++)

# cin>>(\*(vowels+i));

# 

# for (int i = 0;(\*(vowels+i))!= '\0'; i++)

# {

# if ((\*(vowels+i) == 'A') || (\*(vowels+i) == 'a'))

# {

# cA++;

# }

# if ((\*(vowels+i) == 'E' )||(\*(vowels+i) == 'e'))

# {

# cE++;

# }

# if ((\*(vowels+i) == 'I' )||( \*(vowels+i) == 'i'))

# {

# cI++;

# }

# if ((\*(vowels+i) == 'O' )|| (\*(vowels+i) == 'o'))

# {

# cO++;

# }

# if ((\*(vowels+i) == 'U') || (\*(vowels+i) == 'u'))

# {

# cU++;

# }

# }

# cout << "No. of 'A' characters: " << cA << endl;

# cout << "No. of 'E' characters: " << cE << endl;

# cout << "No. of 'I' characters: " << cI << endl;

# cout << "No. of 'O' characters: " << cO << endl;

# cout << "No. of 'U' characters: " << cU << endl;

# cout << "Total No. of vowels: " << cA+cE+cI+cO+cU<< endl;

# }

# 

# EXPERIMENT-3

**Demonstrate use of structure and union.**

1. **Define a structure type, personal, that would contain person name, date of joining and  salary. Using this structure, write a program to read this information for one person from  the keyboard and print the same on the screen. Define a Union for the same members and  compare the memory occupied by both (structure and union).**

**b) Write a function to enter rollno, marks of the three subject for 3 students and find total  obtained by each student using structure.**

# A:

# #include<iostream>

# #include<string>

# using namespace std;

# struct Personal

# {

# string fname,lname;

# int DD,MM,YYYY;

# long int Salary;

# void enterinfo(string f1name,string l1name,int day,int Month,int Year,long int salary)

# {

# fname=f1name;

# lname=l1name;

# DD=day;

# MM=Month;

# YYYY=Year;

# Salary=salary;

# }

# void printinfo()

# { cout<<"\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n";

# cout<<"Name = "<<fname<<" "<<lname<<endl;

# cout<<"Date of joining = "<<DD<<"/"<<MM<<"/"<<YYYY<<endl;

# cout<<"Salary = Rs "<<Salary<<endl;

# cout<<"\n\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n\n";

# }

# };

# union Personal\_union

# {

# string fname,lname;

# int DD,MM,YYYY;

# long int Salary;

# void enterinfo(string f1name,string l1name,int day,int Month,int Year,long int salary)

# {

# fname=f1name;

# lname=l1name;

# DD=day;

# MM=Month;

# YYYY=Year;

# Salary=salary;

# }

# void printinfo()

# { cout<<"\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n";

# cout<<"Name = "<<fname<<" "<<lname<<endl;

# cout<<"Date of joining = "<<DD<<"/"<<MM<<"/"<<YYYY<<endl;

# cout<<"Salary = Rs."<<Salary<<endl;

# cout<<"\n\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n\n";

# }

# };

# int main()

# { string fname,lname;

# int DD,MM,YYYY;

# long int Salary;

# Personal P1;

# union u\_P1;

# cout<<"Enter the first name: ";

# cin>>fname;

# cout<<"Enter the last name: ";

# cin>>lname;

# cout<<"\n";

# //enter right date:

# while(true)

# {

# cout<<"Enter the day of joining(DD): ";

# cin>>DD;

# if((0<DD)&&(DD<31)) break;

# else cout<<"Invalid Entry! please Enter again: "<<endl;

# }

# cout<<"\n";

# //enter right date:

# while(true)

# {

# cout<<"Enter the Month of joining(MM): "<<endl;

# cin>>MM;

# if((1<=MM)&&(MM<=12)) break;

# else cout<<"Invalid Entry! please Enter again: "<<endl;

# }

# cout<<"\n";

# //enter right date:

# while(true)

# {

# cout<<"Enter the year of joining(YYYY):"<<endl;

# cin>>YYYY;

# if((1900<YYYY)&&(2020>=YYYY)) break;

# else cout<<"Invalid Entry! please Enter again:"<<endl;

# }

# if(YYYY==2020)

# {if((MM>=9)&&(DD>17))

# {cout<<"Time traveling not allowed";

# return 1;}

# }

# cout<<"Enter the salary( in Rs/annum): ";

# cin>>Salary;

# cout<<"\n\n";

# P1.enterinfo(fname,lname,DD,MM,YYYY,Salary);

# P1.printinfo();

# cout<<"\n";

# cout<<"The size occupied by an instance of struct Personal="<<sizeof(Personal)<<"bytes"<<endl;

# cout<<"The size occupied by an instance of union Personal="<<sizeof(Personal\_union)<<"bytes"<<endl;

# cout<<"The size occupied by an instance union is "<<sizeof(Personal)/sizeof(Personal\_union)<<" times less than that of struct's instance"<<endl;

# return 1;

# }

# 

# B:

# #include<iostream>

# #include<string>

# using namespace std;

# struct marks

# {

# string fname,lname;

# int Rollno;

# int Total=0;

# void enterinfo(string f1name,string l1name,int rollno,int M[],int no)

# {

# fname=f1name;

# lname=l1name;

# rollno=rollno;

# for(int i=0;i<no;i++)

# {

# Total+=M[i];

# }

# 

# }

# void printinfo(int M[])

# { cout<<"\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n";

# cout<<"Name of student = "<<fname<<" "<<lname<<endl;

# cout<<" total marks ="<<Total;

# cout<<"\n\*\*\*\*\*\_\_\_\*\*\*\*\*\_\_\_\_\*\*\*\*\*\*\n\n";

# }

# };

# int main()

# { string fname,lname;

# int rollno;

# int no;

# cout<<"Enter the number of subjects in the institution: "<<endl;

# cin>>no;

# int M[no],i=0,i0=0;

# marks P1[3];

# while(i0<3)

# {cout<<"Information regarding student No: "<<i0+1<<"\n\n"<<endl;

# cout<<"Enter the first name: ";

# cin>>fname;

# cout<<"Enter the last name: ";

# cin>>lname;

# cout<<"Enter the roll no: ";

# cin>>rollno;

# cout<<"\n\n";

# i=0;

# while(i<no)

# {cout<<"Enter the marks of subject number " <<(i+1)<<" :";

# cin>>M[i];

# if(M[i]<0)

# {i--;

# cout<<"Invalid Entry !! so Again ";

# }

# i++;

# }

# P1[i0].enterinfo(fname,lname,rollno,M,no);

# cout<<"\n";

# i0++;

# }

# i0=0;

# while(i0<3)

# {

# P1[i0].printinfo(M);

# i0++;

# }

# return 1;

# }

# 

# EXPERIMENT-4

**Demonstrate use of function in C++**

**a) Write a function to swap two numbers using call by reference.**

**b) Write a function to check whether given number is prime number or not.**

**c) Write a function to find the multiplication values and the cubic values using inline  function.**

**d) Write a function to calculate the area of circle, rectangle and triangle using function  overloading.**

# A:

# #include<iostream>

# #include<cstdlib>

# #include <math.h>

# using namespace std;

# float area(float r)

# {

# return(3.14 \* r \* r);

# }

# float area(float b,float h)

# {

# return(0.5 \* b \* h);

# }

# float area(float side1,float side2,float side3)

# {

# 

# 

# float s = (side1+side2+side3)/2;

# float areas = sqrt(s\*(s-side1)\*(s-side2)\*(s-side3));

# return areas;

# }

# int main()

# {

# float b,r,l;

# int ch;

# do

# {

# cout<<"\n\n \*\*\*\*\*Menu\*\*\*\*\* \n";

# cout<<"\n 1. Area of Circle";

# cout<<"\n 2. Area of Triangle";

# cout<<"\n 3. Area of Rectangle";

# cout<<"\n 4. Exit";

# cout<<"\n\n Enter Your Choice : ";

# cin>>ch;

# switch(ch)

# {

# case 1:

# {

# cout<<"\n Enter the Radius of Circle : ";

# cin>>r;

# cout<<"\n Area of Circle : "<<area(r);

# break;

# }

# case 2:

# { float side1, side2, side3;

# cout << "\n\n Find the area of any triangle using Heron's Formula :\n";

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

# cout<<" Input the length of 1st side of the triangle : ";

# cin>>side1;

# cout<<" Input the length of 2nd side of the triangle : ";

# cin>>side2;

# cout<<" Input the length of 3rd side of the triangle : ";

# cin>>side3;

# cout<<area(side1, side2, side3);

# }

# case 3:

# {

# cout<<"\n Enter the Length & Bredth of Rectangle : ";

# cin>>l>>b;

# cout<<"\n Area of Rectangle : "<<area(l,b);

# break;

# }

# case 4:

# exit(0);

# default:

# cout<<"\n Invalid Choice... ";

# }

# }while(ch!=4);

# return 0;

# }

# 

# B:

# #include<iostream>

# using namespace std;

# 

# inline float mul( float a, float b)

# {

# return(a\*b);

# }

# 

# inline float cubic(float a)

# {

# return(a\*a\*a) ;

# }

# 

# int main()

# {

# float x,y;

# cout<<"enter the value of x:";

# cin>>x;

# cout<<"enter the value of y:";

# cin>>y;

# cout<<"multiplication of X & Y: ";

# cout<<mul(x,y)<<"\n";

# cout<<"cube of X: ";

# cout<<cubic(x)<<"\n";

# cout<<"cube of Y: ";

# cout<<cubic(y)<<"\n";

# 

# return 0;

# }

# 

# C:

# #include <iostream>

# using namespace std;

# bool checkPrimeNumber(int);

# int main() {

# int n;

# cout << "Enter a positive integer: ";

# cin >> n;

# if (checkPrimeNumber(n))

# cout << n << " is a prime number.";

# else

# cout << n << " is not a prime number.";

# return 0;

# }

# bool checkPrimeNumber(int n) {

# bool isPrime = true;

# // 0 and 1 are not prime numbers

# if (n == 0 || n == 1) {

# isPrime = false;

# }

# else {

# for (int i = 2; i \*i<= n; ++i) {

# if (n % i == 0) {

# isPrime = false;

# break;

# }

# }

# }

# return isPrime;

# }

# 

# D:

# #include <iostream>

# using namespace std;

# void swap(int &n1, int &n2) {

# int temp;

# temp = n1;

# n1 = n2;

# n2 = temp;

# }

# int main()

# {

# // initialize variables

# int a,b;

# cout<<"Enter the number to be swapped"<<endl;

# cin>>a>>b;

# cout << "Before swapping" << endl;

# cout << "a = " << a << endl;

# cout << "b = " << b << endl;

# // call function to swap numbers

# swap(a, b);

# cout << "\nAfter swapping" << endl;

# cout << "a = " << a << endl;

# cout << "b = " << b << endl;

# return 0;

# }

# 

# EXPERIMENT-5

**a) Define a Class “Student” having following data members and member functions:  ∙ Data members:**

**1. Name of the Student**

**2. Roll number**

**3. Department Name**

**∙ Member functions:**

**1. Enter Student Details**

**2. Display Student Details**

**Write a main program to create the object and test the program for 5 students.**

**b) Define a Class “bank\_account” having following data members and member functions:  ∙ Data members:**

**1. Name of the depositor**

**2. Account number**

**3. Type of account**

**4. Balance amount in the account**

**∙ Member functions:**

**1. To assign the initial values**

**2. To deposit an amount**

**3. To withdraw an amount after checking the balance**

**4. To display name and balance**

# A:

#include<iostream>

using namespace std;

class student

{

private:

char f1name[31];

char f2name[31];

int roll;

char department[100];

public:

void getd(void);

void showd(void);

};

void student::getd()

{

cin>>f1name;

cin>>f2name;

cin >> roll;

cin>>department;

}

void student::showd()

{

cout<<"Name of the student is "<<f1name<<" "<<f2name<<endl;

cout<<"Department of the student is "<<department<<endl;

cout<<"Roll number of the student is "<<roll<<endl;

}

int main()

{

int i;

student s[5];

for(i=0;i<5;i++)

{

cout<<"Enter the details of the student "<<i+1<<" : "<<endl;

s[i].getd();

cout<<endl;

}

for(int i=0;i<5;i++)

{

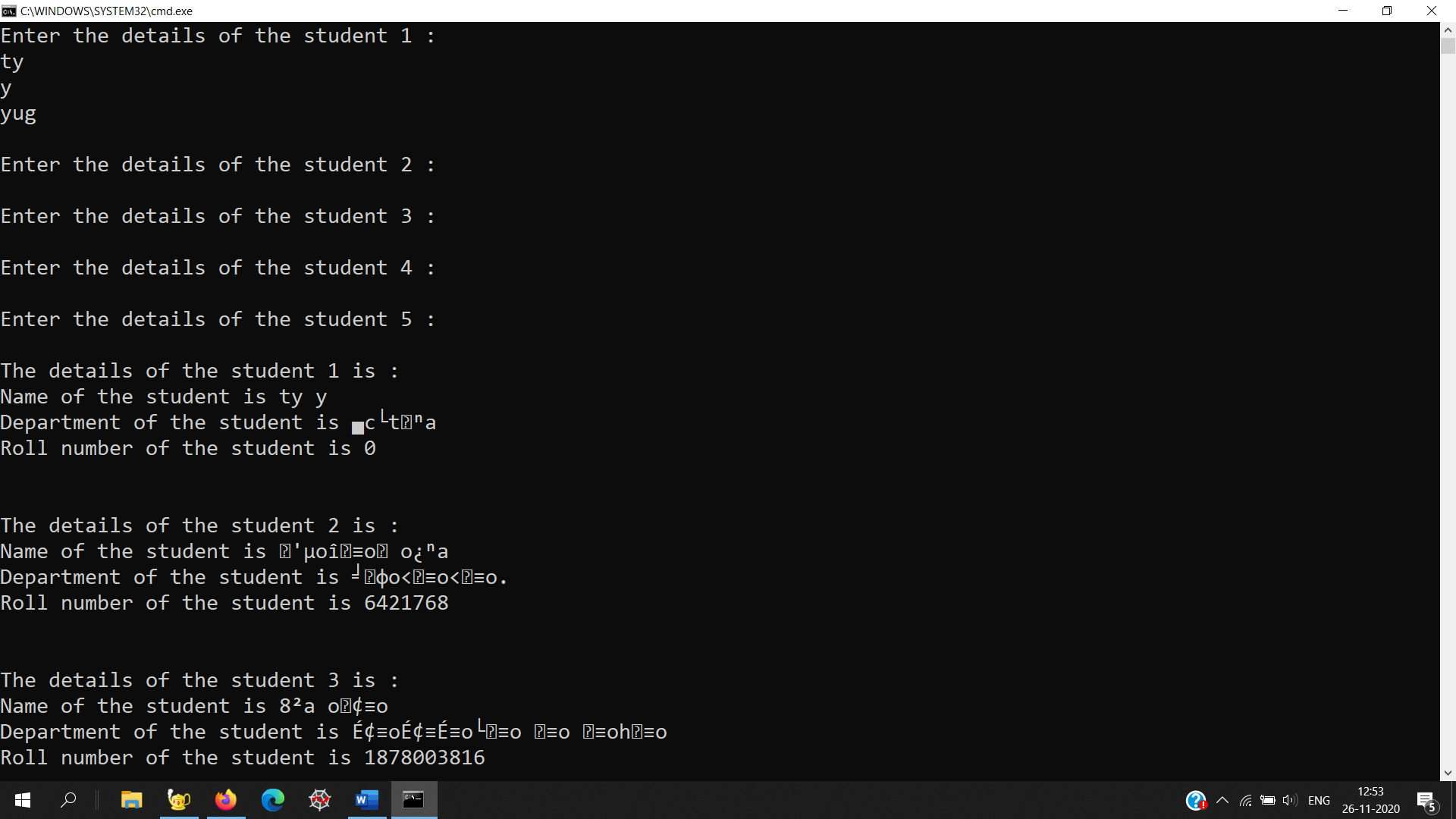
cout<<"The details of the student "<<i+1<<" is : "<<endl;

s[i].showd();

cout<<endl<<endl;

}

}



**B:**

#include <iostream>

using namespace std;

int main() {

// your code goes here

long long int T;

cin>>T;

while(T)

{ long long int N,flag=0;

cin>>N;

int pld=N&(N-1);

if(pld==0)

{cout<<"-1"<<endl;

}

else

{

if(N==3)

{

cout<<1<<" "<<3<<" "<<2<<endl;

}

if(N==1)

{

cout<<1<<endl;

}

if(N>4)

{

cout<<2<<" "<<3<<" "<<1<<" ";

for(int i=4;i<N;i++)

{

if((i&(i-1))==0)

{

if(i+1<N)

{

cout<<i+1<<" "<<i<<" ";

}

else

{

cout<<i+1<<" "<<i<<endl;

flag=1;

}

i=i+1;

}

else{

cout<<i<<" ";

}

}

if (flag!=1)

{cout<<N<<endl;

}

}

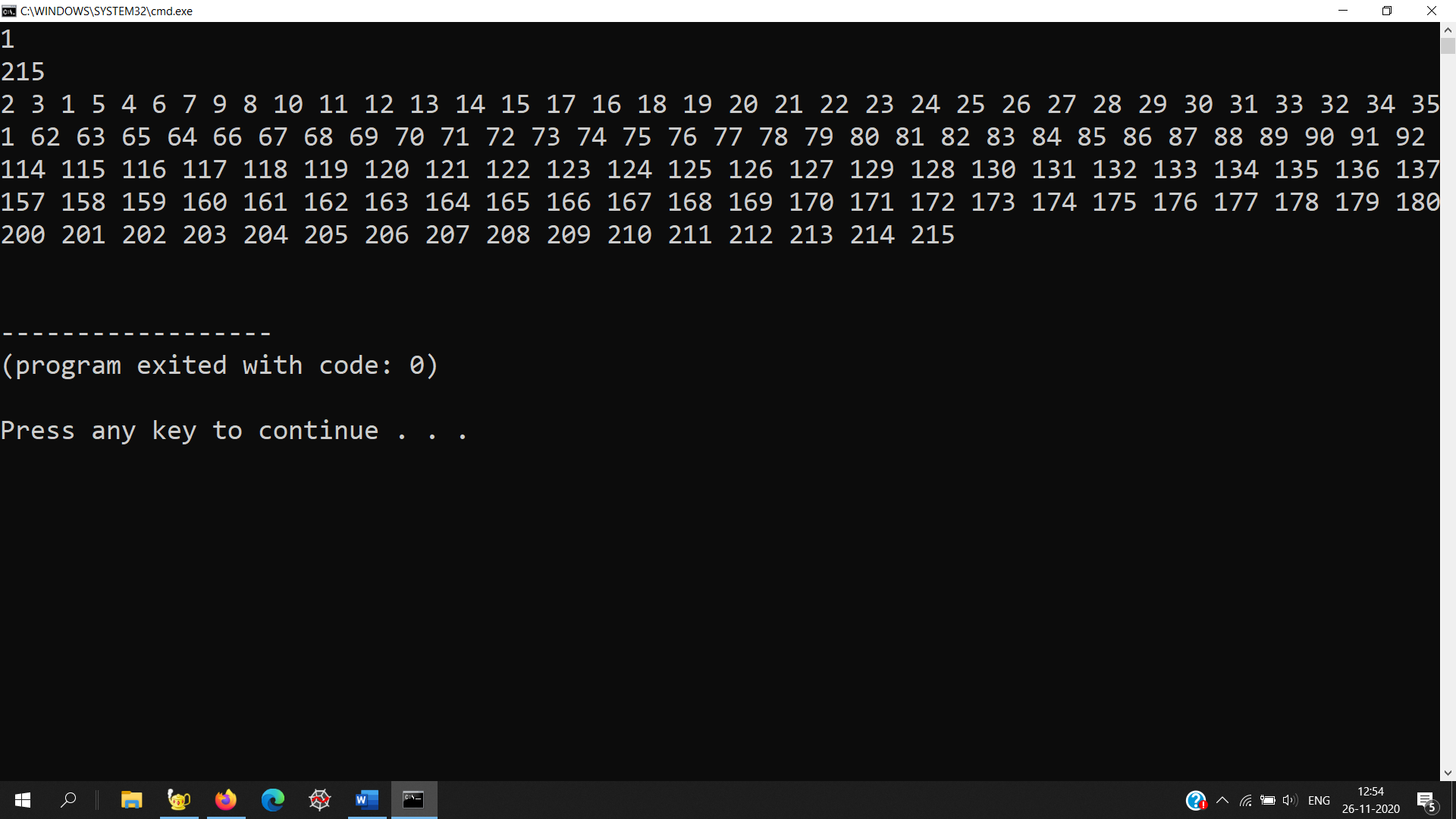
}

T--;

}

return 0;

}



**EXPERIMENT-6**

**Demonstrate use of friend function with class in C++**

**a) Create two classes DM and DB which store the value of distances. DM stores distance in  meters and centimeters and DB in feet and inches. Write a program that can read values  for the class objects and add one object of DM with another object of DB by keeping  following in mind.**

**i. Use friend function to carry out the addition.**

**ii. The object that stores the results may be a DM or DB object, depending on  the unit in which the results are required.**

**b) Write a program to calculate number of object created for particular class.**

**A:**

#include<iostream>

using namespace std;

class DB;

class DM

{

float meter,centimeter;

public:

void getdata()

{

cout<<"\nEnter the distance in(meter-centimeter):";

cin>>meter>>centimeter;

}

void display()

{

cout<<"\nThe distance is:";

cout<<meter<<" meters and "<<centimeter<<" centimeter";

}

friend void add(DM &,DB &);

};

class DB

{

float inch,feet;

public:

void getdata()

{

cout<<"\nEnter the distance in(feet-inch):";

cin>>feet>>inch;

}

void display()

{

cout<<"\nThe distance is:";

cout<<feet<<" feet and "<<inch<<" inch";

}

friend void add(DM &,DB &);

};

void add(DM &a,DB &b)

{

int ch;

cout<<"\nPress 1 for meter-centi:";

cout<<"\nPress 2 for feet-inch:";

cout<<"\nEnter your choice:";

cin>>ch;

if(ch==1)

{

DM d;

int c=(a.meter\*100+a.centimeter+b.feet\*30.48+b.inch\*2.54);

if(c>=100)

{

d.meter=c/100;

d.centimeter=c%100;

}

else

{

d.meter=0;

d.centimeter=c;

}

d.display();

}

else

{

DB d;

int i=(a.meter\*39.37+a.centimeter\*.3937008+b.feet\*12+b.inch);

if(i>=12)

{

d.feet=i/12;

d.inch=i%12;

}

else

{

d.feet=0;

d.inch=i;

}

d.display();

}

}

int main()

{

//clrscr();

DM a;

DB b;

a.getdata();

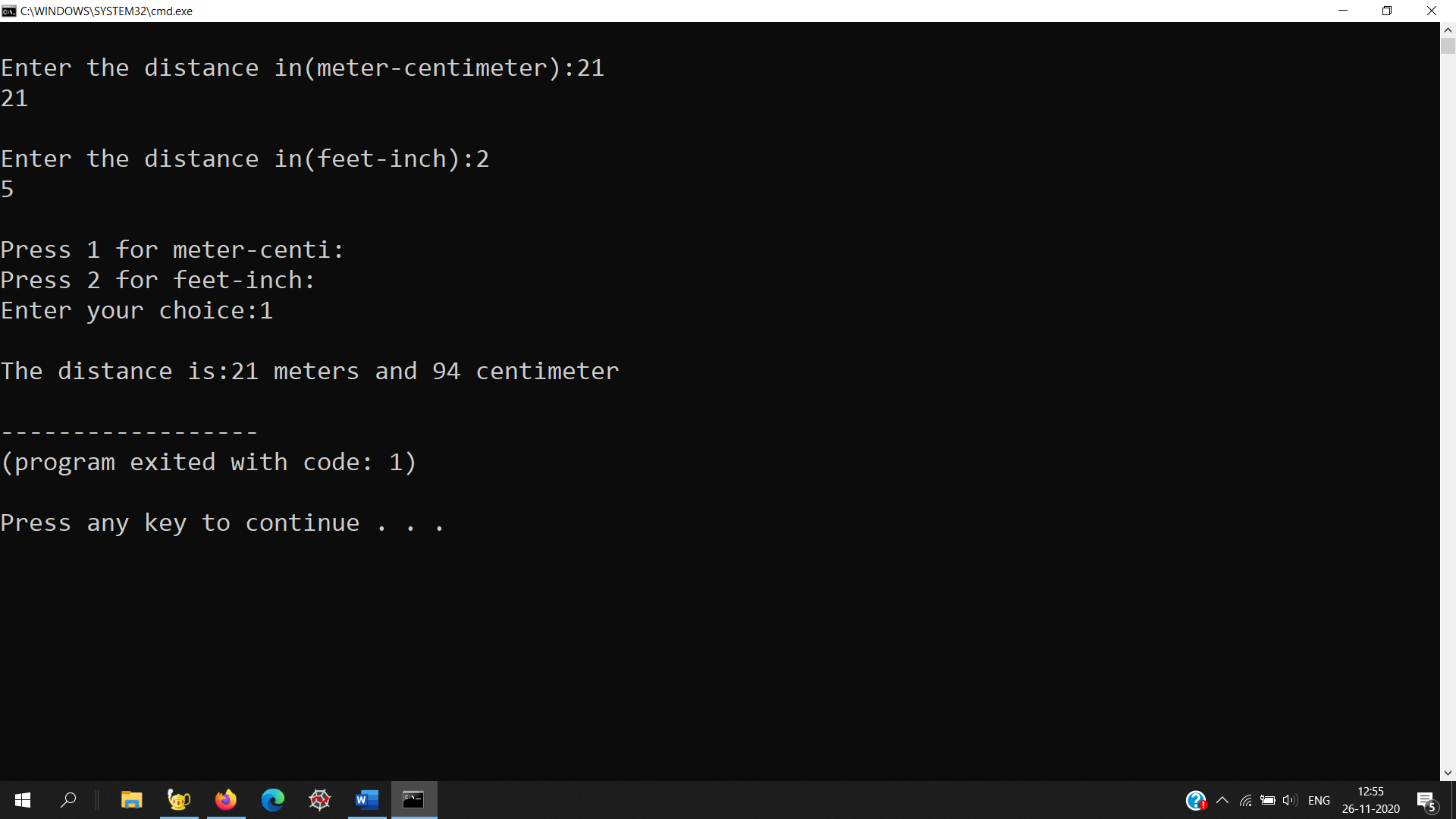
b.getdata();

add(a,b);

//getch();

return 1;

}



**B:**

#include <iostream>

using namespace std;

class Counter

{

private:

static int count;

public:

Counter()

{ count++; }

static void Print()

{

cout<<"\nTotal objects are: "<<count;

}

};

int Counter :: count = 0;

int main()

{

Counter OB1;

OB1.Print();

Counter OB2;

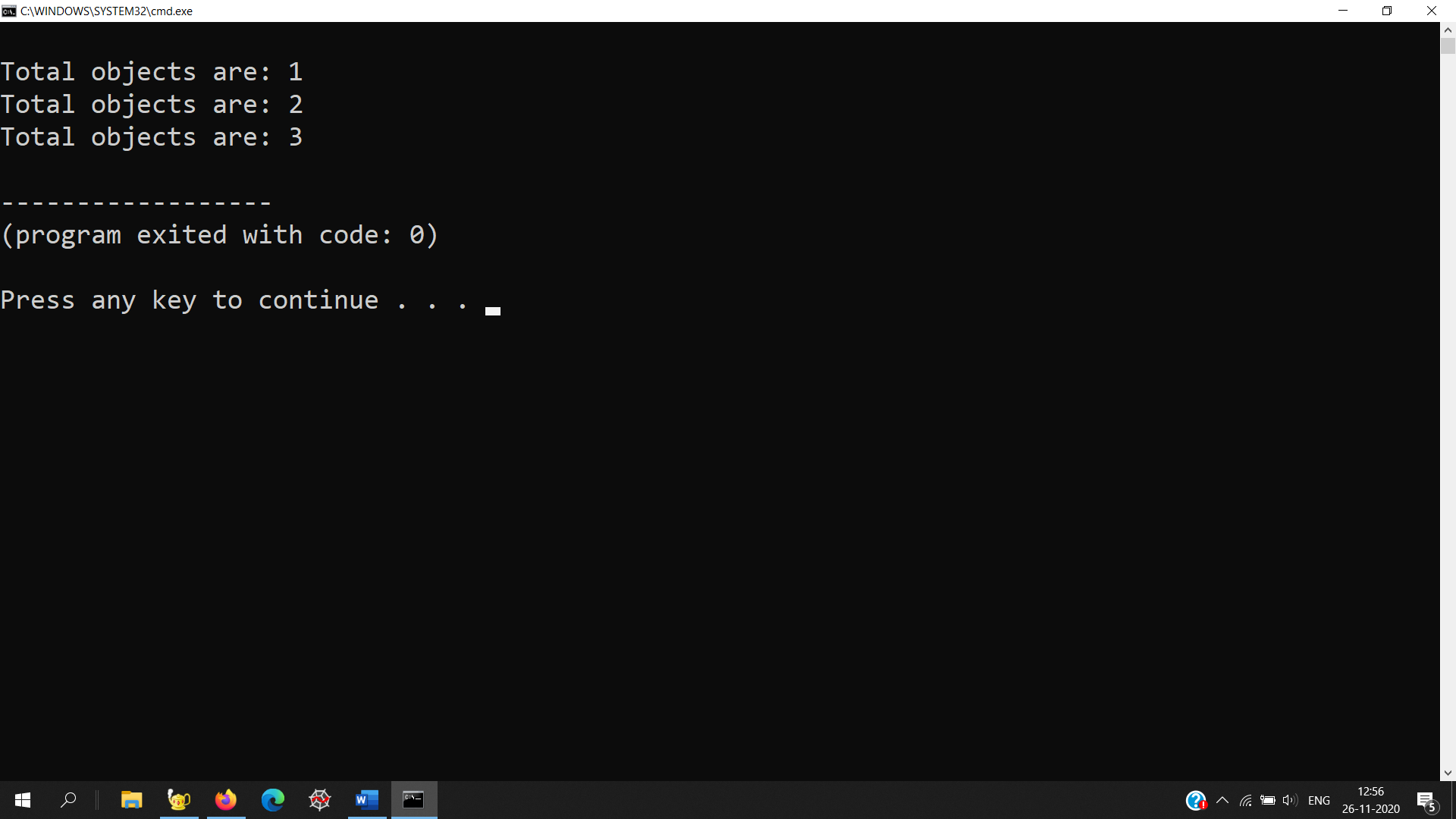
OB2.Print();

Counter OB3;

OB3.Print();

return 0;

}



**EXPERIMENT-7**

Demonstrate operator overloading in C++

a) Create a class coordinate containing x, y and z private variables. Perform operations for  incrementing, decrementing, adding and comparing object(s) by overloading ++, --, += and == operators respectively. Define necessary functions to set and display the variables.

**A:**

#include<iostream>

using namespace std;

class Integer

{

int A;

public:

void setdata(int x)

{

A=x;

}

void showdata()

{cout<<A<<endl;

}

Integer operator --()

{

Integer temp;

temp.A=--A;

return(temp);

}

Integer operator --(int)

{

Integer temp;

temp.A=A--;

return(temp);

}

};

int main()

{

Integer I1,I2,I3;

int L,l;

cin>>L;

l=L;

I1.setdata(L);

I1.showdata();

I2=I1--;

cout<<"I1 after post Dec."<<endl;

I1.showdata();

cout<<"I2 after post Dec."<<endl;

I2.showdata();

L=l;

I1.setdata(L);

I3=--I1;

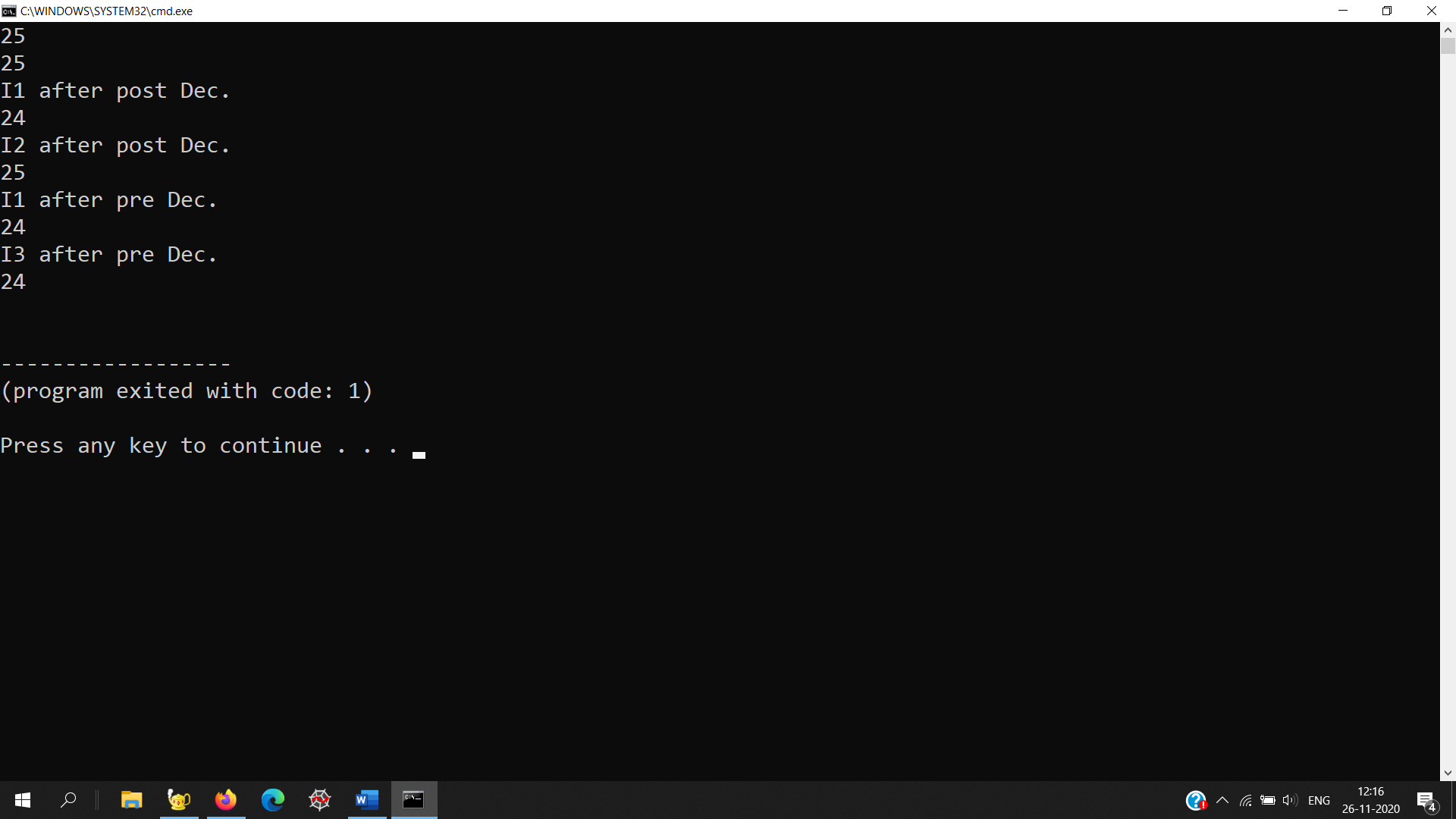
cout<<"I1 after pre Dec."<<endl;

I1.showdata();

cout<<"I3 after pre Dec."<<endl;

I3.showdata();

return(1);}



**EXPERIMENT 8:**

**Demonstrate different kind of inheritance in C++**

**a. Implement following class relationship and test with main class.**

**Staff(code,name)**

**Teacher Typist Officer**

**(subject, Publication) (Speed) (grade)**

**Regular Casual (daily wages)**

**b. Add education details for above classes except for typist. This class will include two  information namely, highest education in general studies and highest professional  qualification.**

**c. Implement the following class using abstract class.**

**A. Shape**

**i. TwoDimensional**

**1. Triangle**

**2. Rectangle**

**3. Circle**

**ii. ThreeDimensional**

**1. Box**

**2. Cone**

**3. Cylinder**

**4. Sphere**

**Use the concept of virtual function for the following:**

**∙ Write a function find\_area() to calculate area of 2-D objects.**

**∙ Write a function find\_volume() to calculate the volume of 3-D objects. ∙ Write a function to display area and volume of corresponding objects.**

**∙ In main function call these functions from base classes. (Hint: Call display() from  object of shape class. Call find\_area() from object of TwoDimensional class. Call  find\_volume() from object of ThreeDimensional class).**

**A:**

#include<iostream>

using namespace std;

class ExpOper

{

private:

int x,y,z;

public:

void getvalue(int,int,int);

void showvalue();

ExpOper operator +=(ExpOper);

ExpOper operator ++(int);//postincrement

ExpOper operator --(int);//postincrement

ExpOper operator ++();//preincrement

ExpOper operator --();//preincrement

bool operator ==(ExpOper);

};

void ExpOper::getvalue(int a,int b,int c)

{

x=a;

y=b;

z=c;

}

void ExpOper::showvalue()

{

cout<<x<<" "<<y<<" "<<z<<endl;

}

ExpOper ExpOper::operator ++()

{ ExpOper C;

C.x=x+1;

C.y=y+1;

C.z=z+1;

x=x+1;

y=y+1;

z=z+1;

return C;

}

ExpOper ExpOper::operator --()

{ ExpOper C;

C.x=x-1;

C.y=y-1;

C.z=z-1;

x=x-1;

y=y-1;

z=z-1;

return C;

}

ExpOper ExpOper::operator ++(int n)

{ ExpOper C;

C.x=x+1;

C.y=y+1;

C.z=z+1;

return C;

}

ExpOper ExpOper::operator --(int n)

{ ExpOper C;

C.x=x-1;

C.y=y-1;

C.z=z-1;

return C;

}

bool ExpOper::operator ==(ExpOper B)

{

return ((B.x==x)&&(B.y==y)&&(B.z==z));

}

int main()

{

ExpOper A,B,C,reseta,resetb;

int a,b,c;

cout<<"Enter The value X,Y,Z for A :"<<endl;

cin>>a>>b>>c;

A.getvalue(a,b,c);//set the value of variables in A

cout<<"Enter The value X,Y,Z for B :"<<endl;

cin>>a>>b>>c;

B.getvalue(a,b,c);//set the value of variables in B

reseta=A;

resetb=B;

C=++A; //after pre increment

cout<<"\n\nPRE INCREMENT\n"<<endl;

cout<<"value of c.. :"<<endl;

C.showvalue();

cout<<"value of A.. :"<<endl;

A.showvalue();

A=reseta;

C=--B; //after pre decrement

cout<<"\n\nPRE DECREMENT\n"<<endl;

cout<<"value of c..:"<<endl;

C.showvalue();

cout<<"value of B.. :"<<endl;

B.showvalue();

B=resetb;

C=A++; //after post increment

cout<<"\n\nPOST INCREMENT\n"<<endl;

cout<<"value of c.. :"<<endl;

C.showvalue();

cout<<"value of A.. :"<<endl;

A.showvalue();

A=reseta;

C=B--; //after post decrement

cout<<"\n\nPOST DECREMENT\n"<<endl;

cout<<"value of c..:"<<endl;

C.showvalue();

cout<<"value of B..:"<<endl;

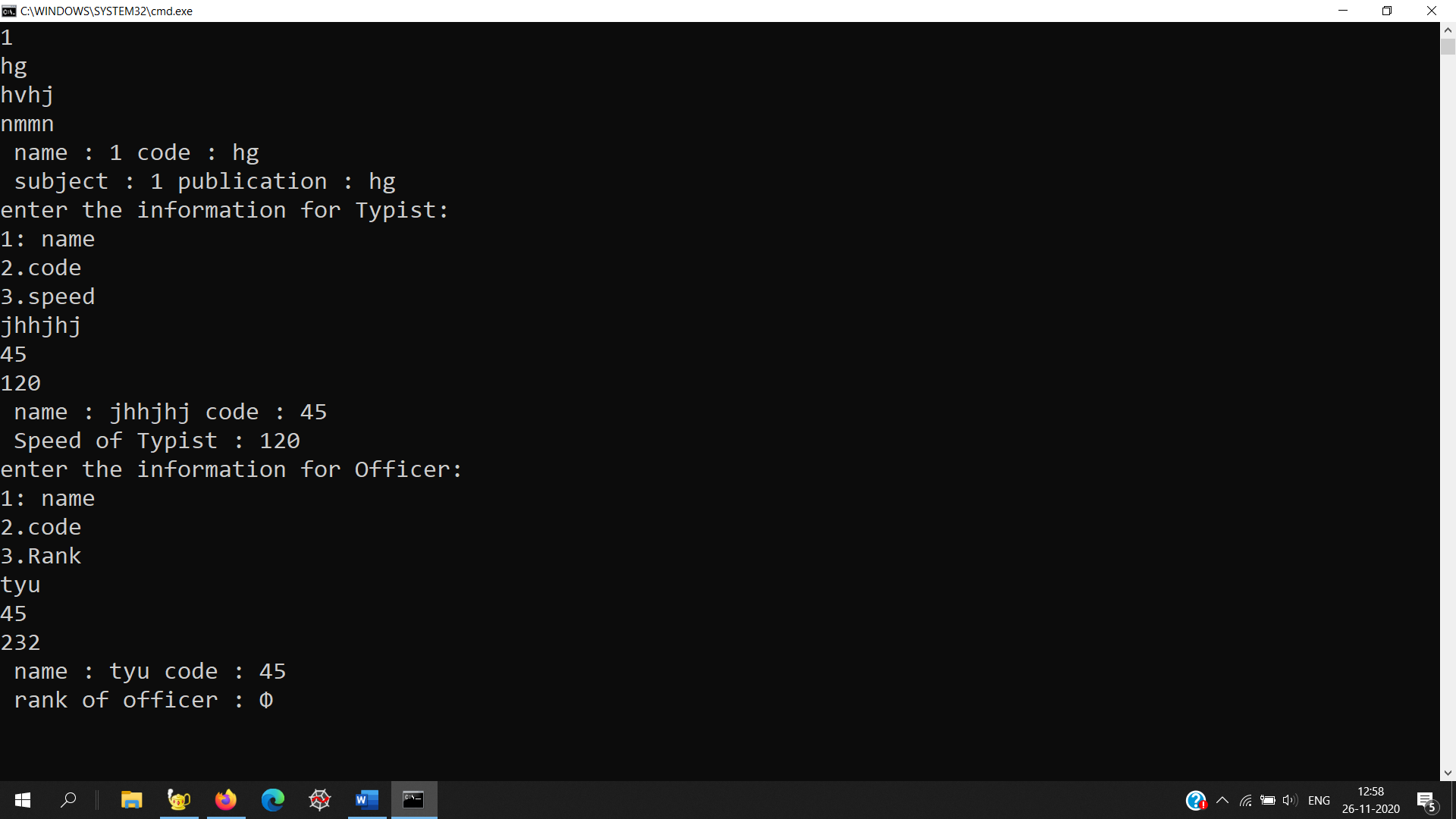
B.showvalue();

B=resetb;

cout<<"value of comparison operator: "<<(A==B)<<endl;

return 1;

}



**B:**

#include<iostream>

using namespace std;

class staff

{ protected:

string code,name;

public:

void getcodeAndname(string,string);

void showCodename();

};

void staff::getcodeAndname(string name1,string code1)

{

name=name1;

code=code1;

}

void staff::showCodename()

{

cout<<" name : "<<name<<" code : "<<code<<endl;

}

class Teacher: public staff

{ protected:

string subject,publication;

public:

void getSubPub(string,string);

void showSubPub();

};

void Teacher::getSubPub(string subject1,string publication1)

{

subject=subject1;

publication=publication1;

}

void Teacher::showSubPub()

{

cout<<" subject : "<<name<<" publication : "<<code<<endl;

}

class Typist: public staff

{ protected:

int Speed;

public:

void speedfunction(int);

void showSpeed();

};

void Typist::speedfunction(int speed1)

{

Speed=speed1;

}

void Typist::showSpeed()

{

cout<<" Speed of Typist : "<<Speed<<endl;

}

class Officer: public staff

{ protected:

string ranked;

public:

void rank(int);

void showRank();

};

void Officer::rank(int ranked1)

{

ranked=ranked1;

}

void Officer::showRank()

{

cout<<" rank of officer : "<<ranked<<endl;

}

class Regular :public Typist

{

};

class casual : public Typist

{

int Dailywage;

public:

casual(int dailywage)

{

Dailywage=dailywage;

}

void ShowDaily()

{

cout<<"Daily wage of casual typist is: "<<Dailywage<<endl;

}

};

int main()

{ Teacher T1;

string teacher\_Name1,T\_Code1,T\_subject1,T\_publication1;

cout<<"enter the information for teacher:\n1: name\n2.code\n3.Subject\n4.Publication"<<endl;

cin>>teacher\_Name1>>T\_Code1>>T\_subject1>>T\_publication1;

T1.getcodeAndname(teacher\_Name1,T\_Code1);

T1.getSubPub(T\_subject1,T\_publication1);

T1.showCodename();

T1.showSubPub();

Typist Ty1;

string typist\_Name1,Ty\_Code1;

int speed;

cout<<"enter the information for Typist:\n1: name\n2.code\n3.speed"<<endl;

cin>>typist\_Name1>>Ty\_Code1>>speed;

Ty1.getcodeAndname(typist\_Name1,Ty\_Code1);

Ty1.speedfunction(speed);

Ty1.showCodename();

Ty1.showSpeed();

Officer o1;

string officier\_Name1,O\_Code1,O\_subject1,O\_publication1;

int rank;

cout<<"enter the information for Officer:\n1: name\n2.code\n3.Rank"<<endl;

cin>>officier\_Name1>>O\_Code1>>rank;

o1.getcodeAndname(officier\_Name1,O\_Code1);

o1.rank(rank);

o1.showCodename();

o1.showRank();

return 1;

}

**C:**

#include<iostream>

#include<conio.h>

#include <cmath>

#include <iomanip>

using namespace std;

const float PI = 3.14;

class Solid {

float radii;

float height;

public:

Solid() { }

virtual float area() = 0;

virtual float volume() = 0;

virtual float getRadius() = 0;

virtual float getHeight() = 0;

virtual ~Solid() { }

};

class Sphere : public Solid {

float radii;

public:

Sphere(float r = 0): radii(r) { }

float area() { return 4 \* PI \* pow(radii, 2); }

float volume() { return (4 \* PI / 3) \* pow(radii, 3); }

float getRadius() { return radii; }

float getHeight() {}

virtual ~Sphere() {}

};

class Cylinder : public Solid {

float radii;

float height;

public:

Cylinder(float r = 0, float h = 0): radii(r), height(h) { }

float area() { return 2 \* PI \* radii \* (radii + height); }

float volume() { return PI \* pow(radii, 2) \* height;}

float getRadius() { return radii; }

float getHeight() { return height; }

virtual ~Cylinder() { }

};

class Cone : public Solid {

float radii;

float height;

public:

Cone(float r = 0, float h = 0): radii(r), height(h) { }

float area() { return PI \* radii \* ( sqrt(pow(radii,2 ) + pow(height, 2)) + radii ); }

float volume() { return (PI / 3.0) \* pow(radii, 2) \* height;}

float getRadius() { return radii; }

float getHeight() { return height; }

virtual ~Cone() { }

};

class shape

{

public:

float area;

void display()

{

cout<<"Area is : "<<area<<endl;

}

};

class D2:public shape

{

public:

float dim1;

float dim2;

void get\_d2()

{

cout<<"\nEnter Dim1 :";

cin>>dim1;

cout<<"Enter Dim2 :";

cin>>dim2;

}

void put\_d2()

{

cout<<"Dim1 is :"<<dim1<<endl;

cout<<"Dim2 is :"<<dim2<<endl;

}

};

class Triangle : public D2

{

public:

void cal1()

{

get\_d2();

area=0.5\*dim1\*dim2;

cout<<"\nFor Triangle : \n";

}

};

class Circle: public D2

{

public:

float dim;

void get\_d2()

{

cout<<"\nEnter Dim1 :";

cin>>dim;

}

void cal3()

{

//cout<<"\nEnter Dim1 :";

// cin>>dim;

area=3.14\*dim\*dim;

cout<<"\nFor Circle : \n";

}

};

class Rectangle : public D2

{

public:

void cal2()

{

get\_d2();

area=dim1\*dim2;

cout<<"\nFor Rectangle : \n";

}

};

int main()

{

//clrscr();

int c;

char ch;

re:

cout<<"1.Triangle.\n";

cout<<"2.Rectangle.\n";

cout<<"3.Circle.\n";

cout<<"Enter your Choice :";

cin>>c;

Triangle t;

switch(c)

{

case 1:

cout<<"\n Triangle \n";

cout<<"Enter L & B :";

t.cal1();

t.put\_d2();

t.display();

break;

case 2:

cout<<"\nEnter L & W :";

Rectangle r;

r.cal2();

r.put\_d2();

r.display();

break;

case 3:

cout<<"\nEnter R :";

Circle c;

c.get\_d2();

c.cal3();

c.display();

break;

default:

{

cout<<"Enter wrong Choice.";

}

}

cout<<"Enter 'Y' for continue or press any key.";

cin>>ch;

if(ch=='y'||ch=='Y')

goto re;

getch();

Sphere sphere(5.0);

Cylinder cylinder(5.0, 5.0);

Cone cone(5.0, 5.0);

cout << "Sphere :\nRadius = " << sphere.getRadius()

<< endl;

cout << "Area of Sphere = " << sphere.area()

<< endl;

cout << "Volume of Sphere = " << sphere.volume()

<< endl;

cout << "Cylinder :\nRadius = " << cylinder.getRadius()

<< ", Height = " << cylinder.getHeight()

<< endl;

cout << "Area of Cylinder = " << cylinder.area()

<< endl;

cout << "Volume of Cylinder = " << cylinder.volume()

<< endl;

cout << "Cone :\nRadius = " << cone.getRadius()

<< ", Height = " << cone.getHeight()

<< endl;

cout << "Area of Cone = " << cone.area()

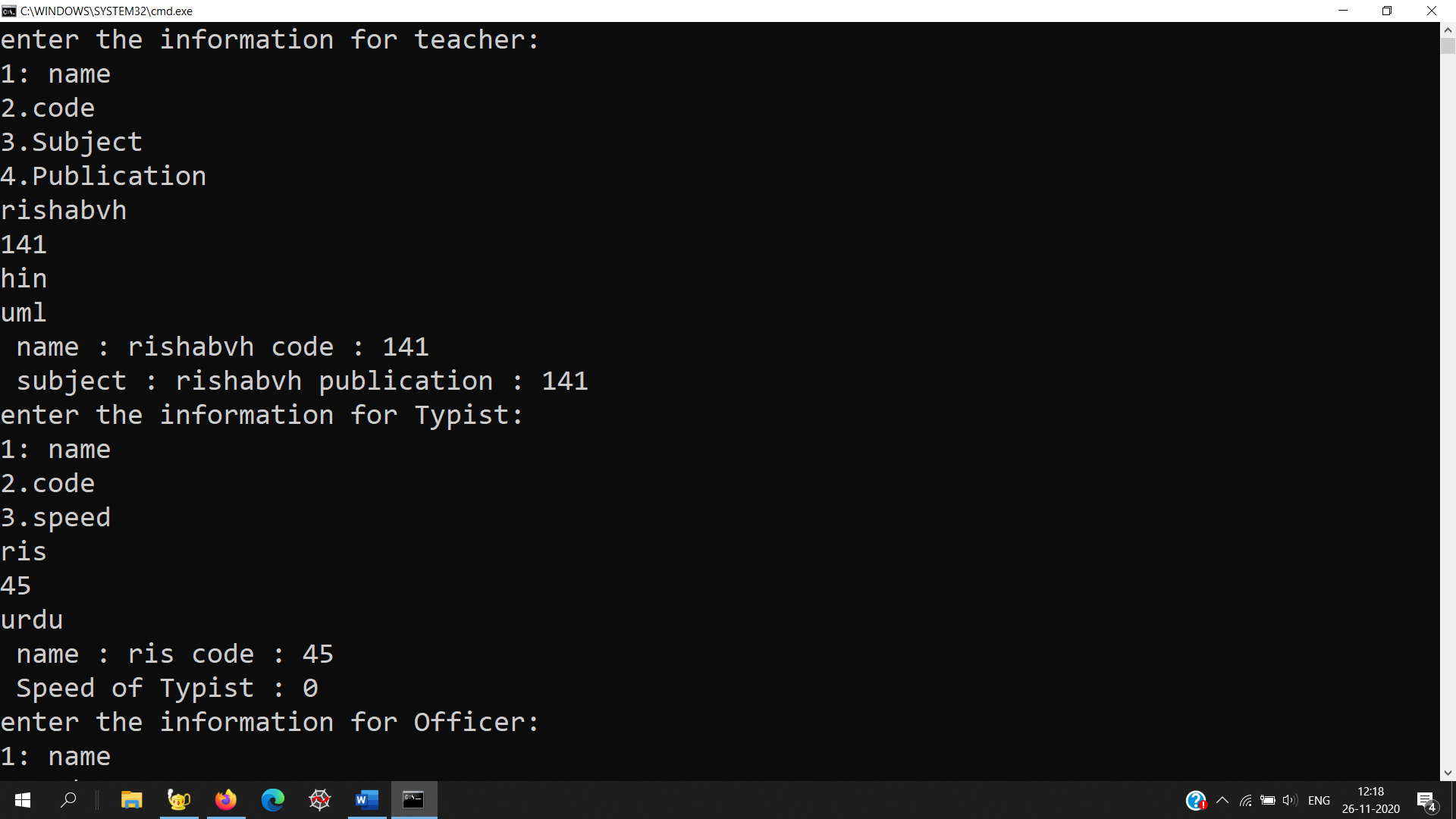
<< endl;

cout << "Volume of Cone = " << cone.volume()

<< endl;

return 0;

}



**EXPERIMENT 9**

**Write a program that shows that two files named ‘Source 1’ and ‘Source 2’ contains sorted list of integers. Read the content of both the files and stores the merged list in sorted form in a new file named ‘Target**

**A:**

#include <iostream>

#include <fstream>

using namespace std;

int main()

{

int num1, num2;

ifstream even;

ifstream odd;

even.open("source1.txt");

odd.open("source2.txt");

ofstream conintegers;

conintegers.open("target.txt");

even >> num1;

odd >> num2;

while(even.eof() && odd.eof())

{

if (num1 < num2)

{

conintegers << num1;

even >> num1;

}

else

{

conintegers << num2;

odd >> num2;

}

}

even.close();

odd.close();

conintegers.close();

return 0;

}

**EXPERIMENT 10**

**Write a function template to perform linear search in an array**

// Find the next lexicographically

// greater permutation of a word

#include <algorithm>

#include <iostream>

using namespace std;

int main()

{

string s = { "fjadbihgec" };

bool val

= next\_permutation(s.begin(),

s.end());

if (val == false)

cout << "No Word Possible"

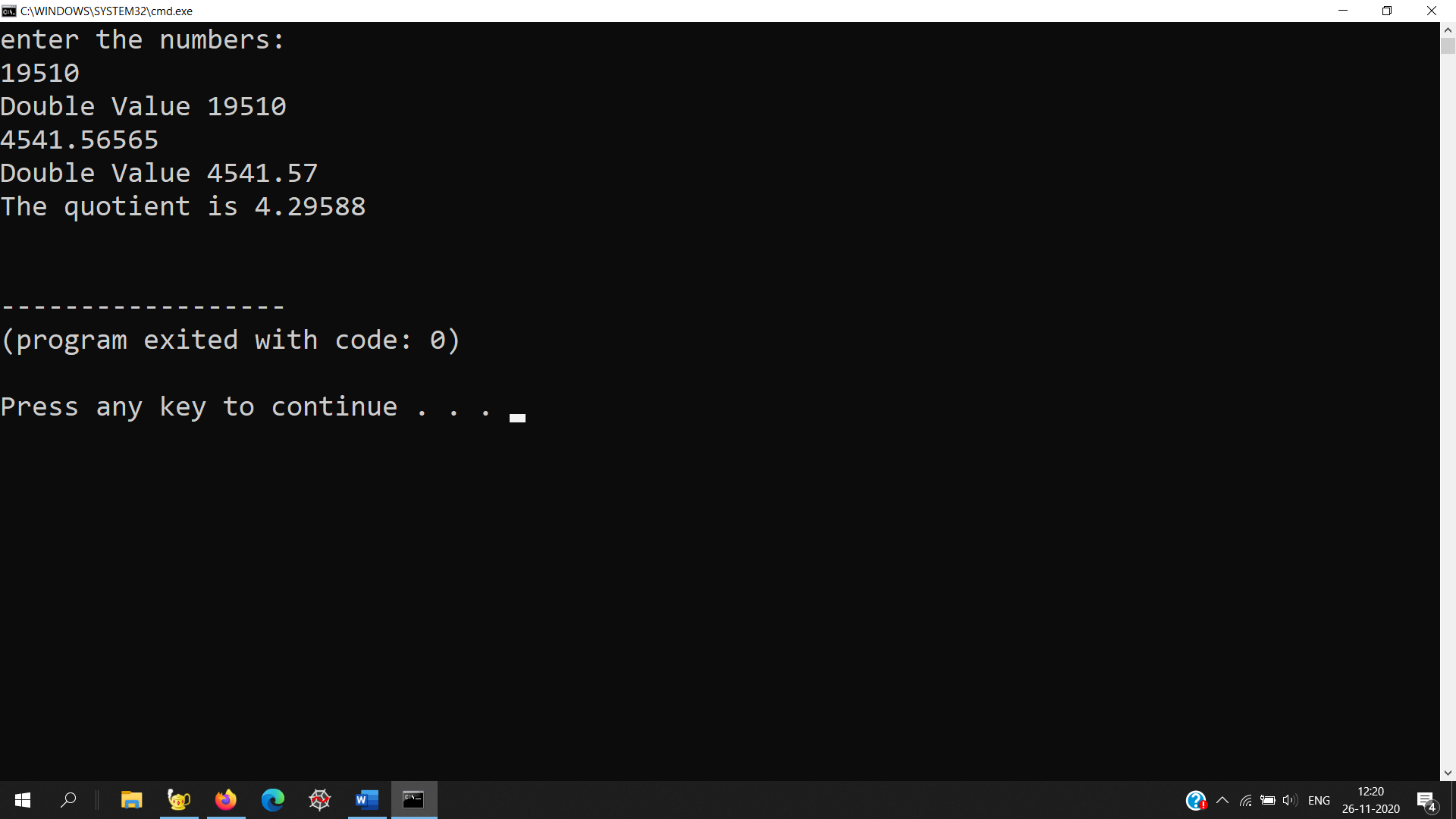
<< endl;

else

cout << s << endl;

return 0;

}



**EXPERIMENT 11**

**Demonstrate exception handling in C++**

**Write a program with the following:**

**a. A function to read two double type numbers from keyboard**

**b. A function to calculate the division of these two numbers**

**c. A try block to throw an exception when wrong type of data is keyed in d. A try block to detect and throw an exception if the condition “divide-by-zero” occurs**

**e. Appropriate catch block to handle the exception thrown.**

#include<iostream>

using namespace std;

void division(double x,double y);

void insert()

{

double num1,num2;

cout<<"enter the numbers: "<<endl;

try{

cin>>num1;

if (cin.fail())

throw (num1);

else

cout << "Double Value " << num1 << endl;

}

catch(double a)

{

cout<<"Incompatible Datatype for value"<<a;

}

try{

cin>>num2;

if (cin.fail())

throw (num2);

else

cout << "Double Value " << num2<< endl;

}

catch(double a)

{

cout<<"Incompatible Datatype for value"<<a;

}

division(num1,num2);

}

double CheckDenominator(double den)

{

if (den == 0)

throw "Error";

else

return den;

} // end CheckDenominator

void division(double num1,double num2)

{

double result;

try {

if (CheckDenominator(num2)) {

result = (num1 / num2);

cout << "The quotient is "

<< result << endl;

}

}

catch (...) {

// Display a that exception has occurred

cout << "Exception occurred" << endl;

}

}

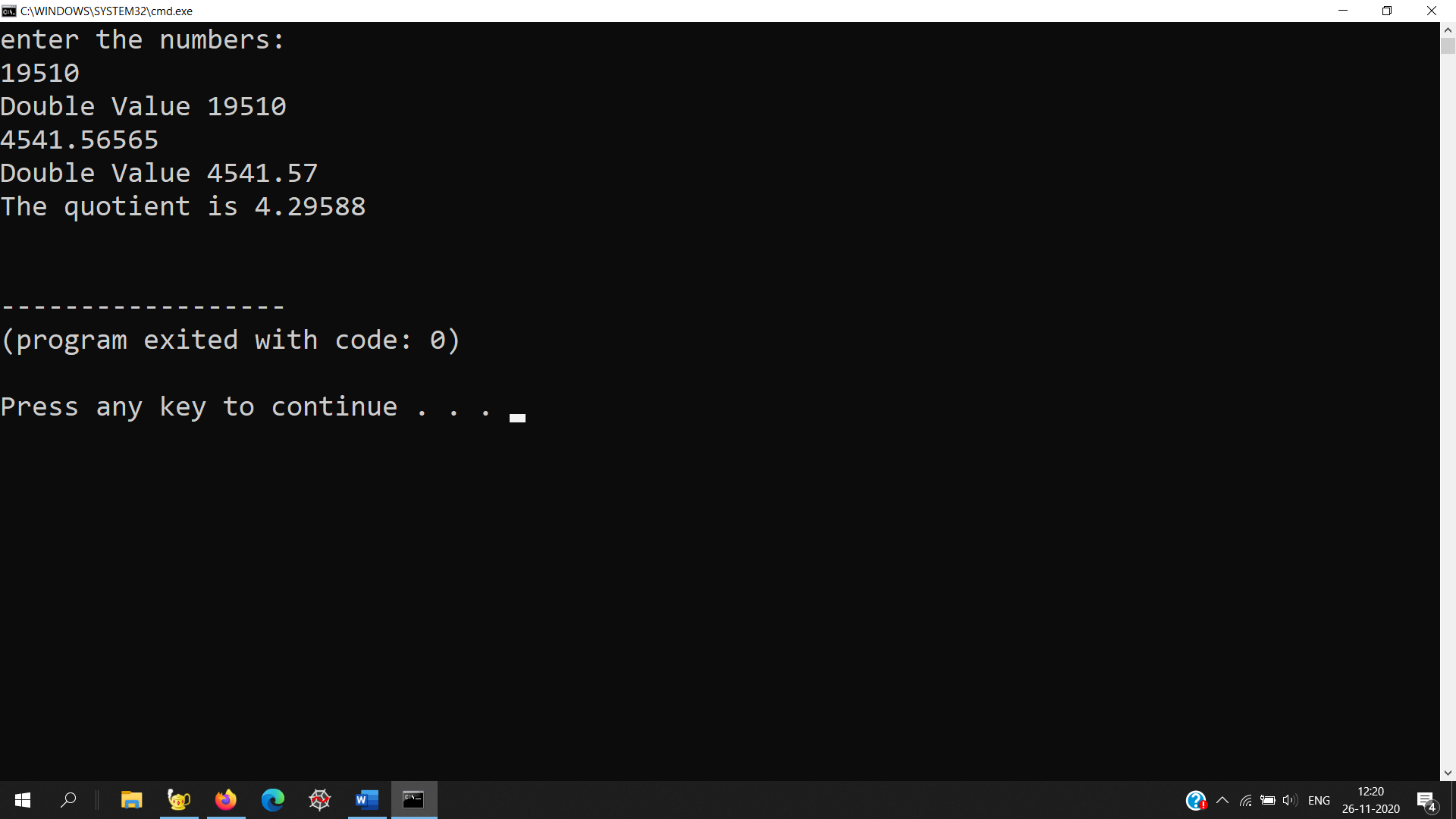
int main()

{

insert();

return 0;

}



**EXPERIMENT 12**

**A table gives a list of car models, its cost and the number of units sold in each type in a specified  period. Write a program to store this table in suitable container and to display interactively the  total value of a particular model sold, given the unit-cost of that model.**

#include<iostream>

#include<vector>

using namespace std;

class car{

string model;

long long cost;

long long unitsSold;

long long totalValue;

public:

car(string model){

this->model=model;

}

string showModel(){

return this->model;

}

void setValues(long long cost, long long unitsSold){

this->cost=cost;

this->unitsSold=unitsSold;

this->totalValue=cost\*unitsSold;

}

void displayNet(){

cout<<"The total price of "<<model<<" sold is "<<totalValue<<endl;

}

};

int main(){

vector<car> carContainer;

while(1){

int choice;

cout<<"Enter your choice"<<endl;

cout<<" '1' To enter the details of a new car"<<endl;

cout<<" '2' To show the details of a car"<<endl;

cout<<" '3' To exit"<<endl;

cin>>choice;

if(choice==1){

cout<<"Enter the name of the car"<<endl;

string name;

cin>>name;

car c(name);

cout<<"Enter the cost and number of units sold"<<endl;

int cost,n;

cin>>cost>>n;

c.setValues(cost,n);

carContainer.push\_back(c);

c.displayNet();

}

else if(choice==2){

cout<<"Enter the name of the car whose details you want to know about"<<endl;

string name;

cin>>name;

int i=0;

for(i=0;i<carContainer.size();i++){

if(carContainer[i].showModel()==name){

carContainer[i].displayNet();

break;

}

}

if(carContainer.size()==i){

cout<<"The name of the car you entered doesn't exsit in our record"<<endl;

}

}

else{

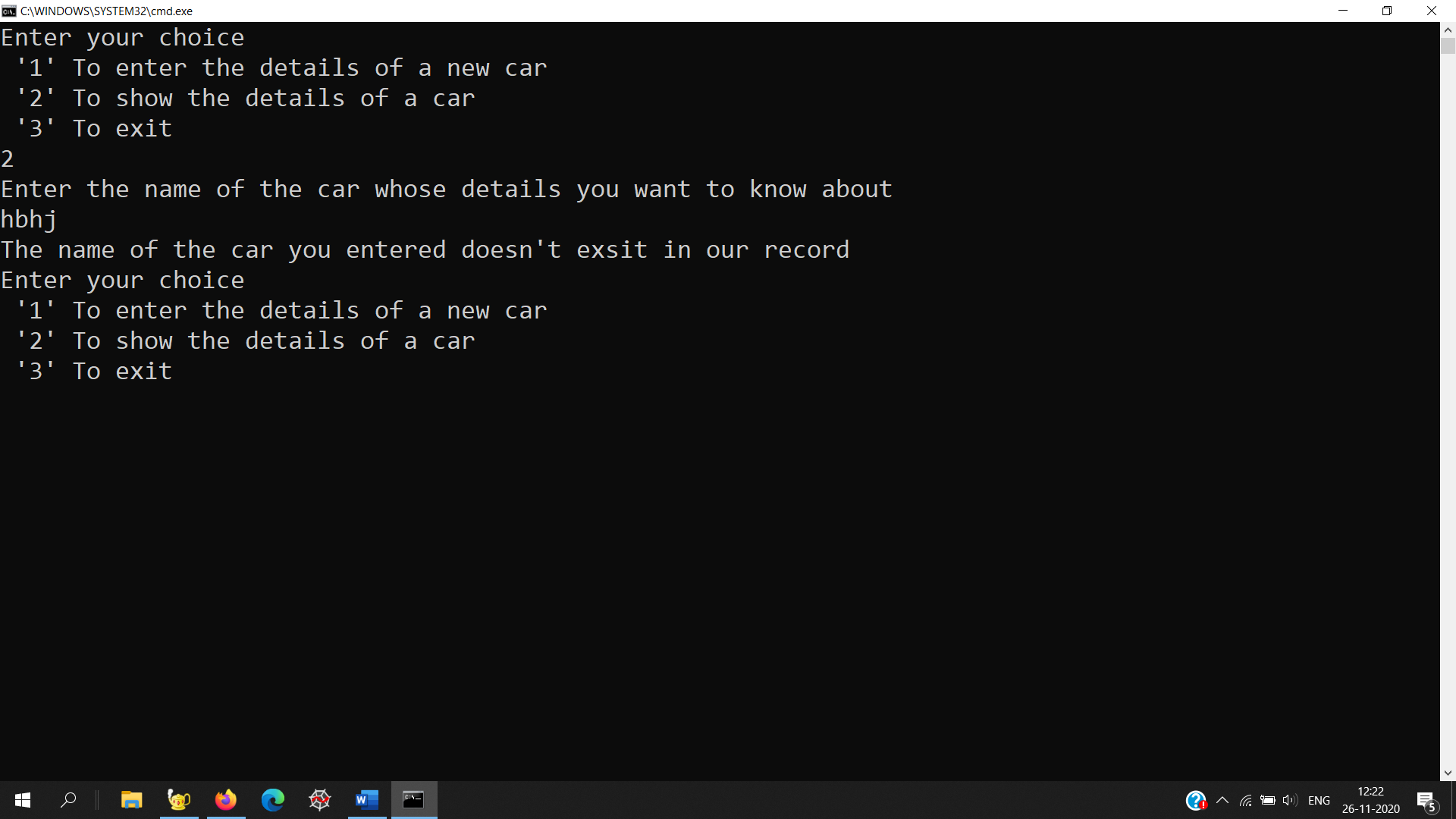
break;

}

}

return 0;

}



**EXPERIMENT 13**

**Write a class template to represent a generic vector. Include the member functions to perform the  following tasks:**

**a. To create a vector**

**b. To modify the value of given element**

**c. To multiply by a scalar value.**

#include<iostream>

using namespace std;

template<class T>

class vector

{

T v[20];

int size;

public:

void create();

void modify();

void mult();

void display();

};

template<class T>

void vector<T>::create()

{

int i;

T value;

char ans;

size=0;

do{

cout<<"\nEnter the index & value:";

cin>>i>>value;

v[i]=value;

size++;

cout<<"\nDo you want more elements?";

cin>>ans;

}while(ans=='y'||ans=='Y');

}

template<class T>

void vector<T>::modify()

{

int key;

T newval;

cout<<"\nEnter index for modificaion:";

cin>>key;

cout<<"\nEnter new value:";

cin>>newval;

v[key]=newval;

}

template<class T>

void vector<T>::mult()

{

int i;

int scalarval;

cout<<"\nEnter scalar value for multiplication";

cin>>scalarval;

for(i=0;i<size;i++)

v[i]=v[i]\*scalarval;

}

template<class T>

void vector<T>::display()

{

int i;

cout<<"\nSize of vector is:"<<size;

cout<<"\nElements in vector are:";

cout<<"(";

for(i=0;i<size;i++)

{

cout<<v[i]<<" ";

}

cout<<")";

}

int main()

{

int ch;

vector<int>obj;

cout<<"\nProgram for template class";

do

{

cout<<"\nMAIN MENU";

cout<<"\n1.Create";

cout<<"\n2.Display";

cout<<"\n3.Mult";

cout<<"\n4.Modify";

cout<<"\n0.Exit";

cout<<"\nEnter your choice:";

cin>>ch;

switch(ch)

{

case 1:

obj.create();

break;

case 2:

obj.display();

break;

case 3:

obj.mult();

break;

case 4:

obj.modify();

break;

case 0:

cout<<"\nExit\n";

break;

default:

cout<<"\nInvalid choice";

break;

}

}while(ch!=0);

return 0;

}

