

SESSION 2021-22

OOPS

(Object Oriented Programming System)

Lab File

**COURSE:- BCA**

**ROLL NO :- 41221139**

**SUBMITTED BY :- SUBMITTED TO:-**

**Sachin Rajbhar Ms. Komal Dhingra**

OOPS C++ program upto generic programming

Index

1. Write a program to determine the percentage of a student in three subjects.

2. Write a program to determine the greatest of three numbers using conditional operator.

3. An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting

is done by marking the candidate number on the ballot paper. Write a program to read the

ballots and count the vote cast for each candidate using an array variable ‘count’. In case, a

number read is outside the range 1 to 5, the ballot should be considered as a ‘spoilt ballot’,

and the program should also count the number of spoilt ballots.

4. Write a program to create a reference variable in C++.

5. Write a program to swap two numbers using call by reference.

6. Determine the sum of series 1 + x + x^2/2 + x^3/3 +...........n terms.

7. Write a program to convert decimal to binary using array.

8. Write a program to convert decimal to hexadecimal using array.

9. Write a program using memory management operators new and delete.

10. Write a program to implement manipulators setw, setprecision and setbase.

11. Write a program to demonstrate Lamda expression calculating sum.

12. Write a program to demonstrate folding expressions.

13. Write a program to classify a character as alphanumeric, digit, alphabet, uppercase and

lowercase.

14. Write a program to calculate voting percentage of 3 candidates in 5 regions.

15. Define a class to represent a bank account. Include the following members:

Data members:

a) Name of the depositor

b) Account number

c) Type of account

d) Balance amount in the account

Member functions:

a) To assign initial values

b) To deposit an amount

c) To withdraw an amount after checking the balance.

d) To display the details.

Write a main program to test the program.

16. Write a program to demonstrate a pre-processor directive calculating the sum of square of

two numbers.

17. Five voters are contesting in three regions. Each voter has name, id, total no of votes obtained

in respective regions. Determine using a C++ program, which voter received the maximum

voting percentage combinedly in three regions using the concept of array of objects.

18. Write a program to access the class members using pointer to member operators.

19. Write a program to overload class member functions on number and type of arguments.

20. Write a program to demonstrate inline functions.

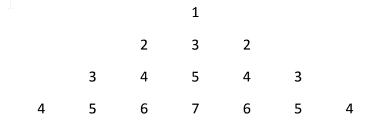
21. Write a program to demonstrate static data members & static member functions.

22. Write a program to calculate the mean using friend functions.

23. Write a friend function to calculate income tax for doctor, scientist, and teacher. Assume that

tax for each the three professionals is 10% of total of hra, basic salary(bs) & dearness

allowance(da).

24. Write a C++ program to print the following pattern 

25. Create a class COMPLEX with members as ‘real’ & ‘imag’. Write a program to add two complex

numbers together. The function should receive two complex objects to be added. Set the real

& imag part through a function setval () with default value as ‘2’ for the imaginary part.

26. WAP to overload constructor complex int terms of number and type of arguments. Create a

default constructor, constructor with one and two arguments respectively. Write a friend

function that should return complex number containing the sum of any two numbers created

by constructor.

27. Design a class string with attributes ‘name’ and ‘length’ which are private members and

method display() in public section. Create a dynamic constructor reading the string in

dynamically allocated memory. Design a destructor for the class which should release the

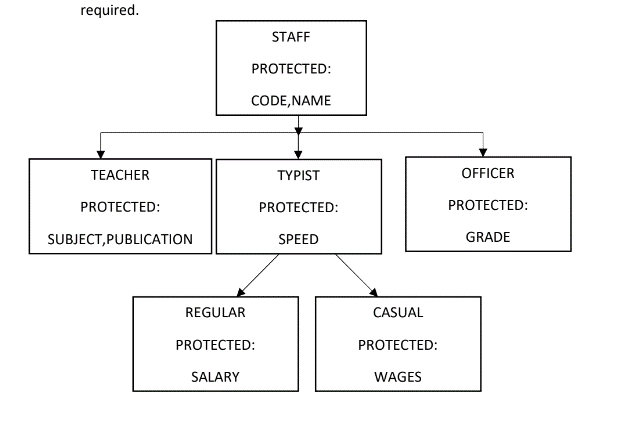
dynamically allocated memory.

28. An educational institution wishes to maintain a database of its employees. The database is

divided into a number of classes whose hierarchical relationship are shown in the fig. The

figure also show the minimum information required for each class. Specify all the classes and

define functions to create the database and retrieve individual information as and when

required. 

29. Write a program to implement the following hybrid inheritance 

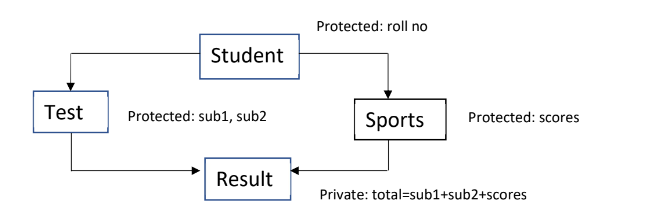
30. Write a program to implement aggregation ‘Course has instructor’ and ‘Course has textbook’.

31. Write a program to implement function overriding.

32. Write a program to initialise base class constructor through derive class constructor in case of

multiple inheritance.

33. Write a program to display the student result in the following scenario using concept of virtual

base class. 

34. Write a program to overload binary operator ‘+’ to concatenate two character arrays together.

35. Write a program to overload ‘>’ operator to determine if length of one character array is

greater than other.

36. Create a base class shape with two double type values to compute area of figures. Derive two

classes Triangle and Rectangle from base class shape. Add two member functions in base class

getdata() - to initialise base class members and displayarea() - to compute and display area of

figures. The displayarea() is virtual function in base class. Redefine this function in derived

class. Accept dimensions from user and display the area using run time polymorphism.

37. Write a program to overload ‘!’ operator to reverse the case of a string.

38. Write a program to implement any sorting algorithm using template functions.

39. Write a program to overload template functions.

40. Write a program using class templates to obtain sum of two complex numbers. The class

template should have two placeholders and a constructor to initialize the members.

41. Create a class student with members as rollno and name, public member functions to read

and display members.

a) Write array of object of class student to a .dat file and read the contents.

b) Modify the contents of nth object in file and display the updated contents.

42. Write a program with a

a) Function to read two integer variables and an operator of type char.

b) Function to calculate the result of operator on operand.

c) try block to throw exception when wrong type of data for operand is entered

d) try block to throw exception when operator is other than ‘+’, ‘-’, ‘\*’, ‘/’.

e) catch block to handle the exceptions.

43. Write a program to perform following password validation(s):

a) Length of password should be minimum 6 characters.

b) It should contain atleast one digit.

Throw exceptions and handle them if validations are not met. In case of successful validation,

print “Password Validation Successful”.

**EXPERIMENT-1**

**AIM:** WAP to determine the percentage of a student in three subjects.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

int main()

{

    string name;

    int sub1,sub2,sub3;

    int sum,per;

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

    cin>>name;

    cout<<"Enter the marks of 1st subject"<<endl;

    cin>>sub1;

     cout<<"Enter the marks of 2nd subject"<<endl;

    cin>>sub2;

     cout<<"Enter the marks of 3rd subject"<<endl;

    cin>>sub3;

    sum=sub1+sub2+sub3;

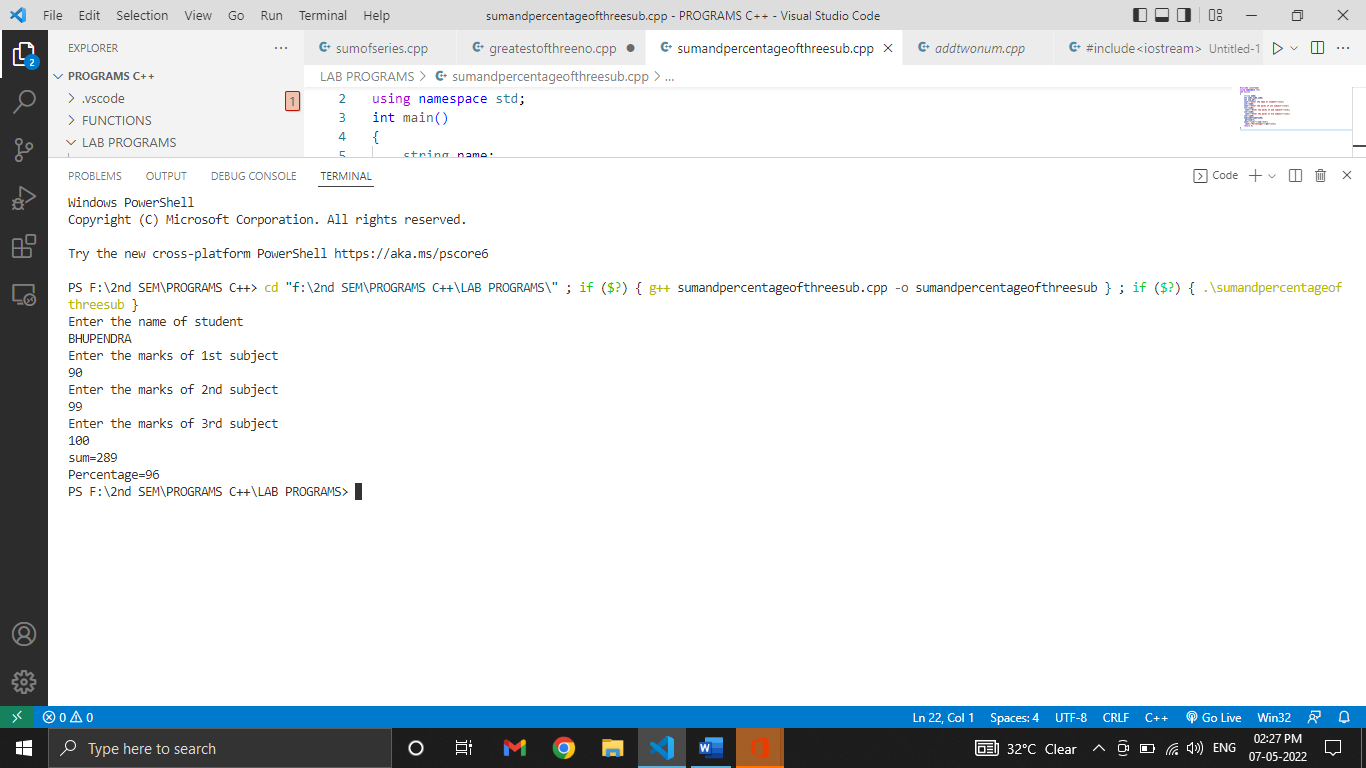
     per=sum/3;

    cout<<"sum="<<sum<<endl;

     cout<<"Percentage="<<per<<endl;

    return 0;

}

**OUTPUT:**

**EXPERIMENT-2**

**AIM:** WAP to determine the greatest of three numbers using conditional operator.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

int main()

{

int a,b,c;

cout<<"ENTER FIRST NUMBER" <<endl;

cin>>a;

cout<<"ENTER SECOND NUMBER" <<endl;

cin>>b;

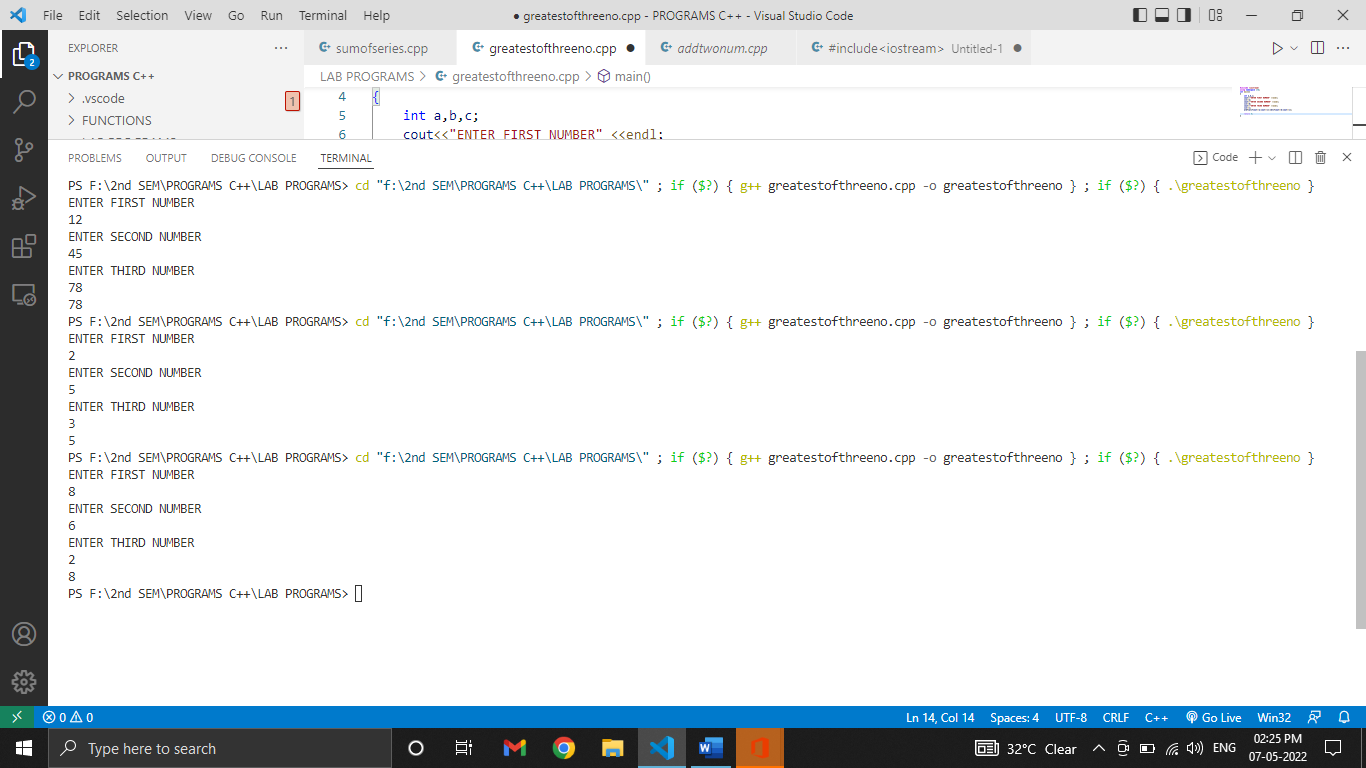
cout<<"ENTER THIRD NUMBER" <<endl;

cin>>c;

a>b?(a>c?cout<<a:cout<<c):(b>c?cout<<b:cout<<c);

return 0;

}

**OUTPUT:**

**EXPERIMENT-3**

**AIM:** An election is contested by five candidates. The candidates are numbered 1 to 5 and the voting is done by marking the candidate number on the ballot paper. Write a program to read the ballots and count the vote cast for each candidate using an array variable ‘count’. In case, a number read is outside the range 1 to 5, the ballot should be considered as a ‘spoilt ballot’, and the program should also count the number of spoilt ballots.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

int input;

char choice='y';

int count[6]={0,0,0,0,0,0};

do{

cout<<"\nCHOOSE THE CONTESTENT YOU WANT TO VOTE (1-5):";

cin>>input;

switch(input){

case 1:count[0]++;

break;

case 2:count[1]++;

break;

case 3:count[2]++;

break;

case 4:count[3]++;

break;

case 5:count[4]++;

break;

default:"SPOILT BALLOT";

count[5]++;

break;

}

cout<<"DO YOU WANT TO CAST MORE VOTES (Y/N):";

cin>>choice;

}while(choice==121);

cout<<"The no. of votes of candidate 1= "<<count[0]<<endl;

cout<<"The no. of votes of candidate 2= "<<count[1]<<endl;

cout<<"The no. of votes of candidate 3= "<<count[2]<<endl;

cout<<"The no. of votes of candidate 4= "<<count[3]<<endl;

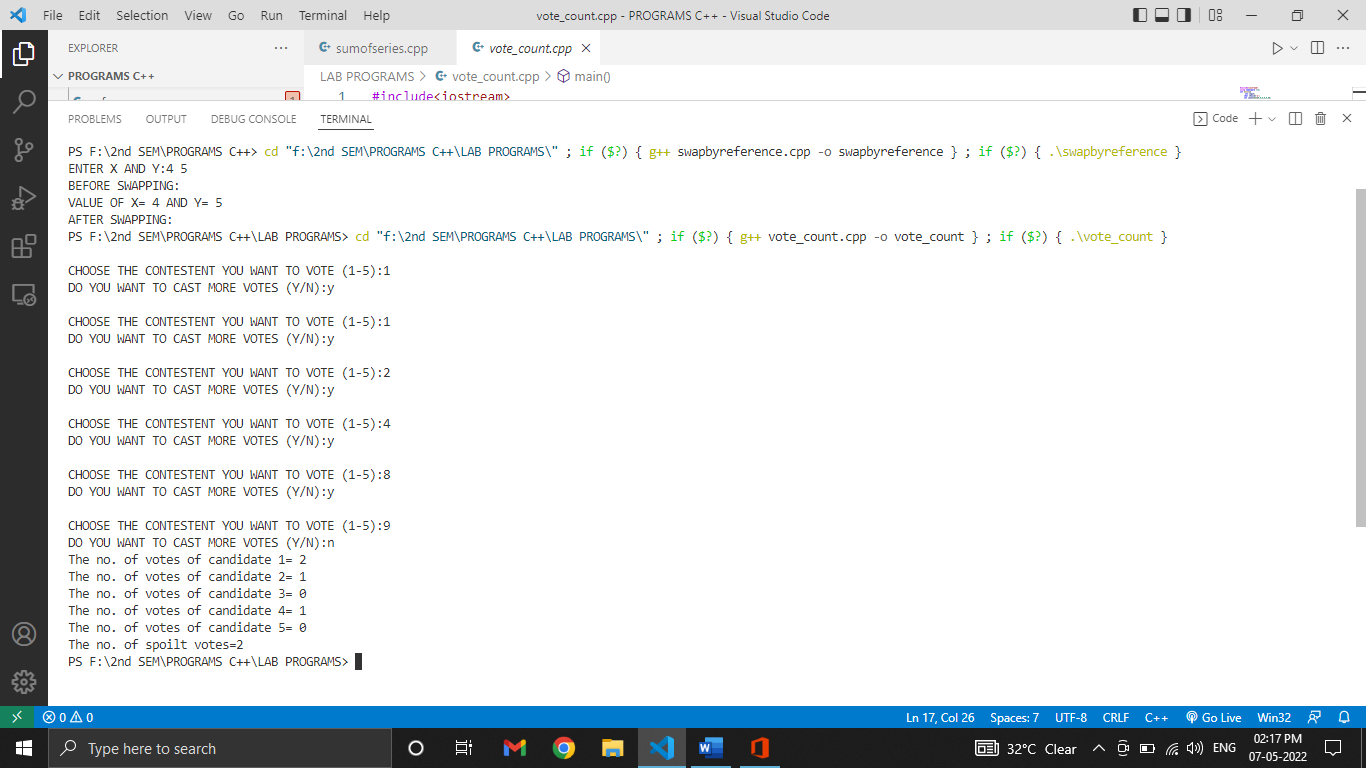
cout<<"The no. of votes of candidate 5= "<<count[4]<<endl;

cout<<"The no. of spoilt votes="<<count[5]<<endl;

return 0;

}

**OUTPUT:**



**EXPERIMENT-4**

**AIM:** WAP to create a reference variable in C++.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

int sum=10;

int &total=sum;

sum=30;

cout<<"SUM= "<<sum<<endl;

cout<<"TOTAL= "<<total<<endl;

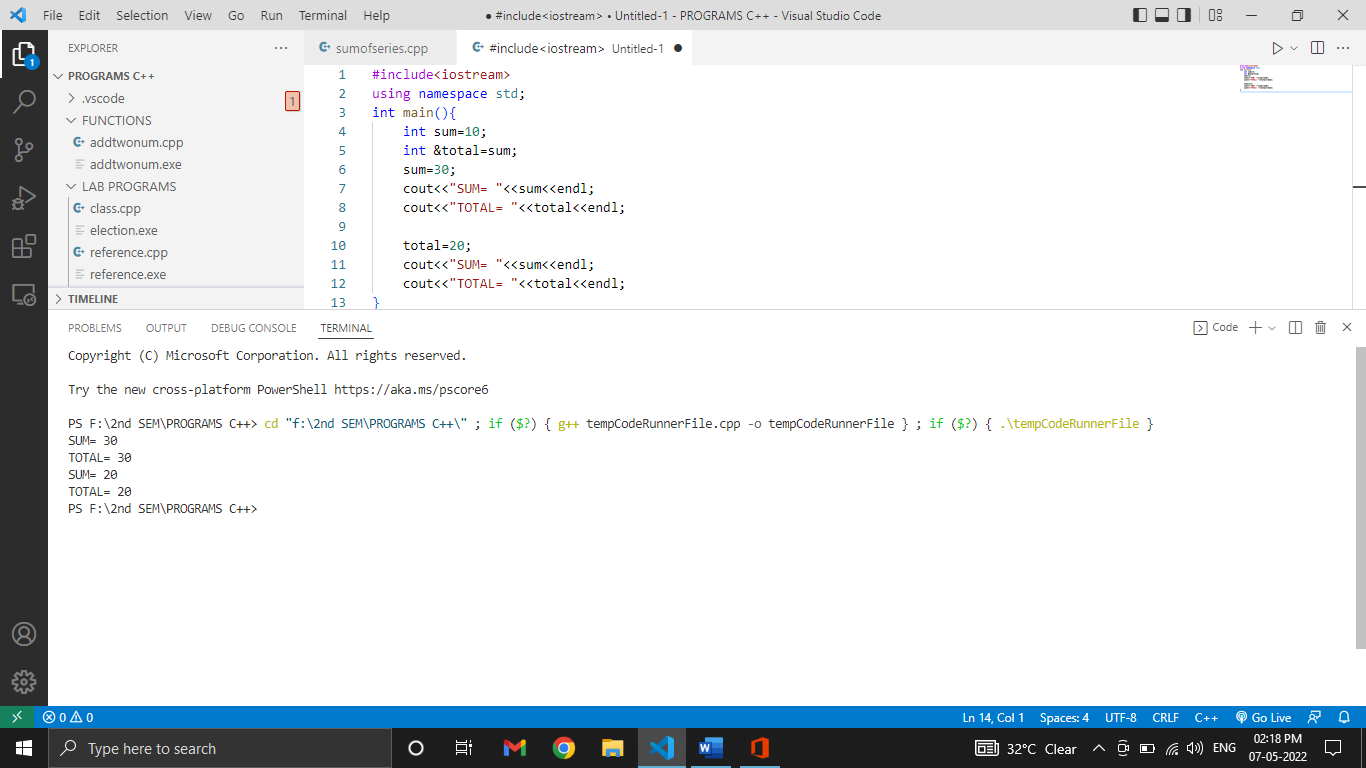
total=20;

cout<<"SUM= "<<sum<<endl;

cout<<"TOTAL= "<<total<<endl;

}

**OUTPUT:**



**EXPERIMENT-5**

**AIM:** WAP to swap two numbers using call by reference.

**SOFTWARE USED:** VS CODE

**PROGRAM:**

void swap(int &a , int &b);

void swap(int &a , int &b){

int temp;

temp=a;

a=b;

b=temp;

}

int main(){

int x,y;

cout<<"ENTER X AND Y:";

cin>>x>>y;

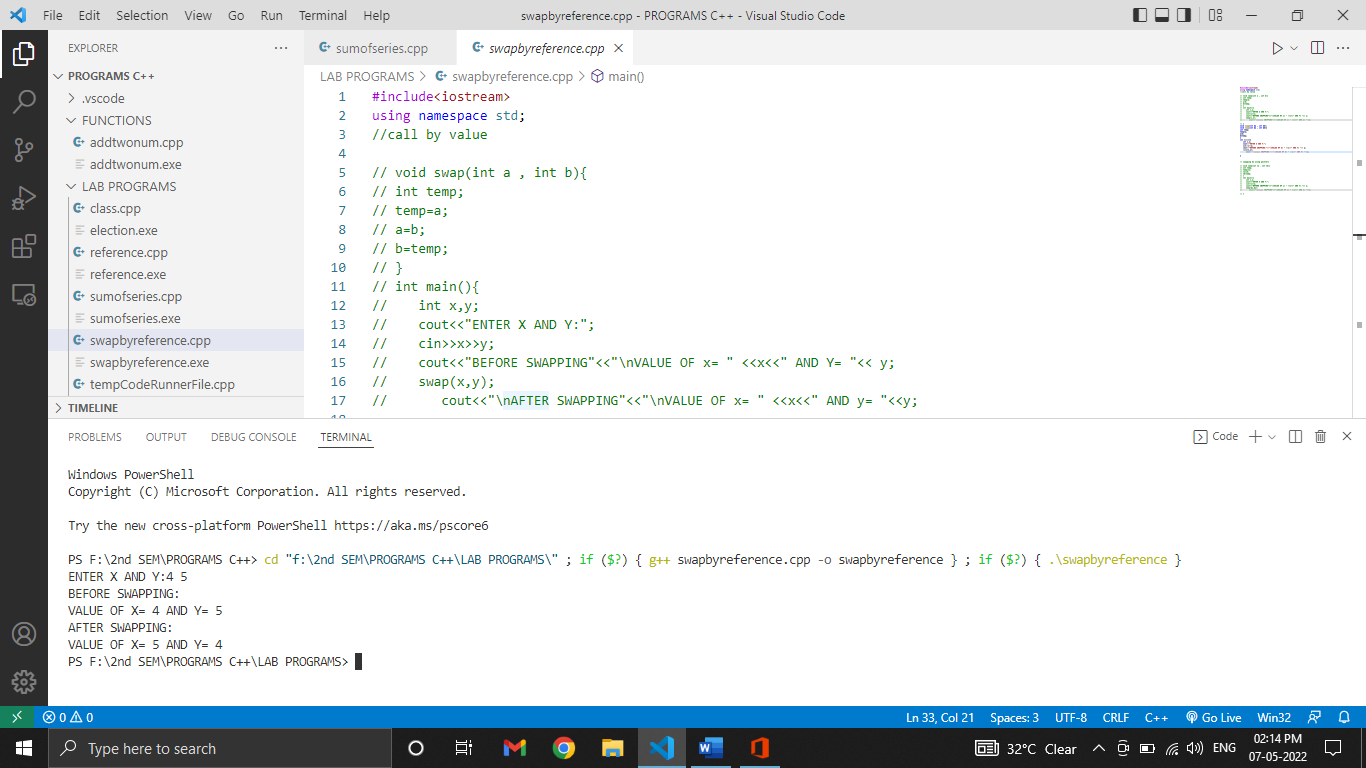
cout<<"BEFORE SWAPPING:"<<"\nVALUE OF X= " <<x<<" AND Y= "<< y;

swap(x,y);

cout<<"\nAFTER SWAPPING:"<<"\nVALUE OF X= " <<x<<" AND Y= "<<y;

}

**OUTPUT:**



**EXPERIMENT-6**

**AIM:** Determine the sum of series 1 + x + x^2/2 + x^3/3 +...........n terms.

**SOFTWARE USED:** VS code

**PROGRAM:**

**#include<iostream>**

#include<math.h>

using namespace std;

int main(){

int i,n,x;

double sum = 1;

cout<<"ENTER X:";

cin>>x;

cout<<"ENTER N:";

cin>>n;

for(i=1;i<=n;i++){

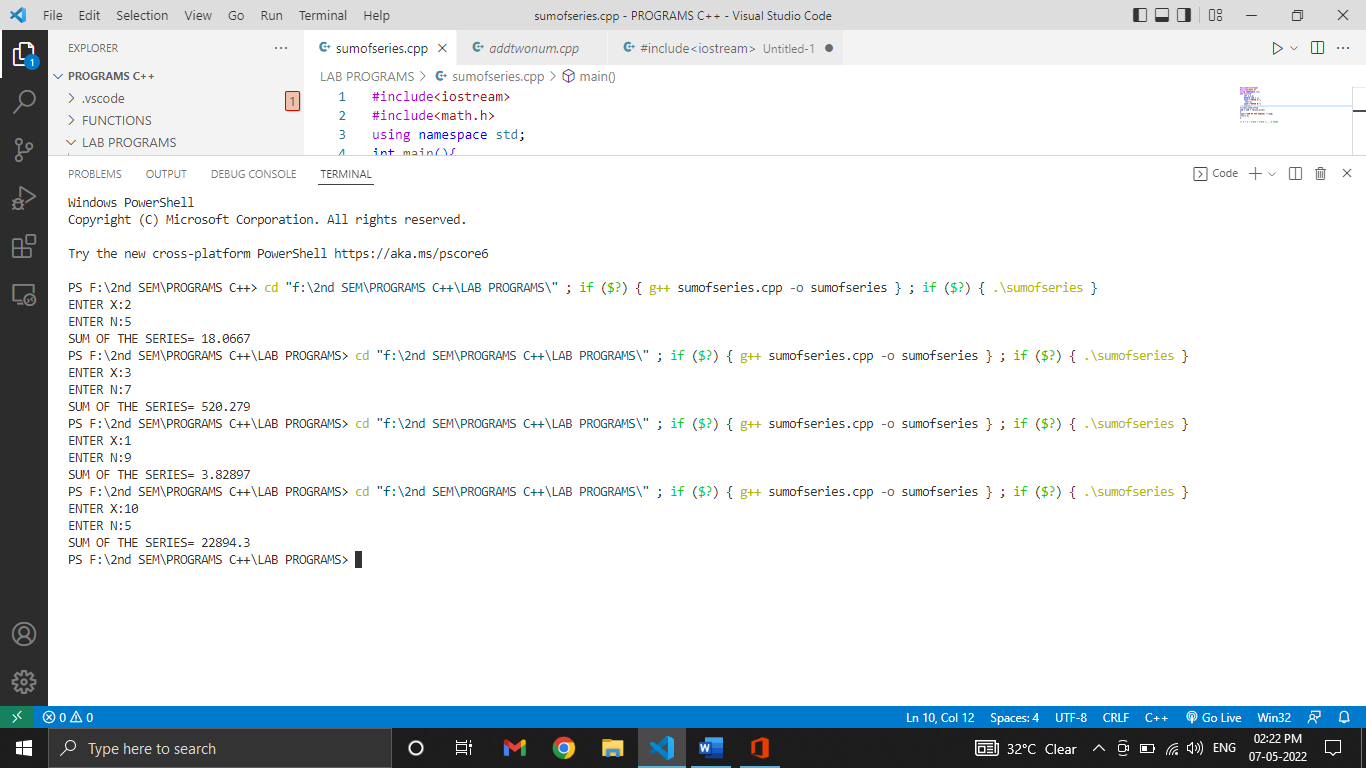
sum = sum + (pow(x,i)/i);

}

cout<<"SUM OF THE SERIES= "<<sum;

return 0;

}

**OUTPUT:**

**EXPERIMENT-7**

**AIM:** WAP to convert decimal to binary using array.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

    int i,j,num;

    int arr[20];

    cout<<"Enter a decimal number:"<<"\n";

    cin>>num;

    for(i=0;num>0;i++){

        arr[i]=num%2;

        num=num/2;

    }

    cout<<"Binary value is: ";

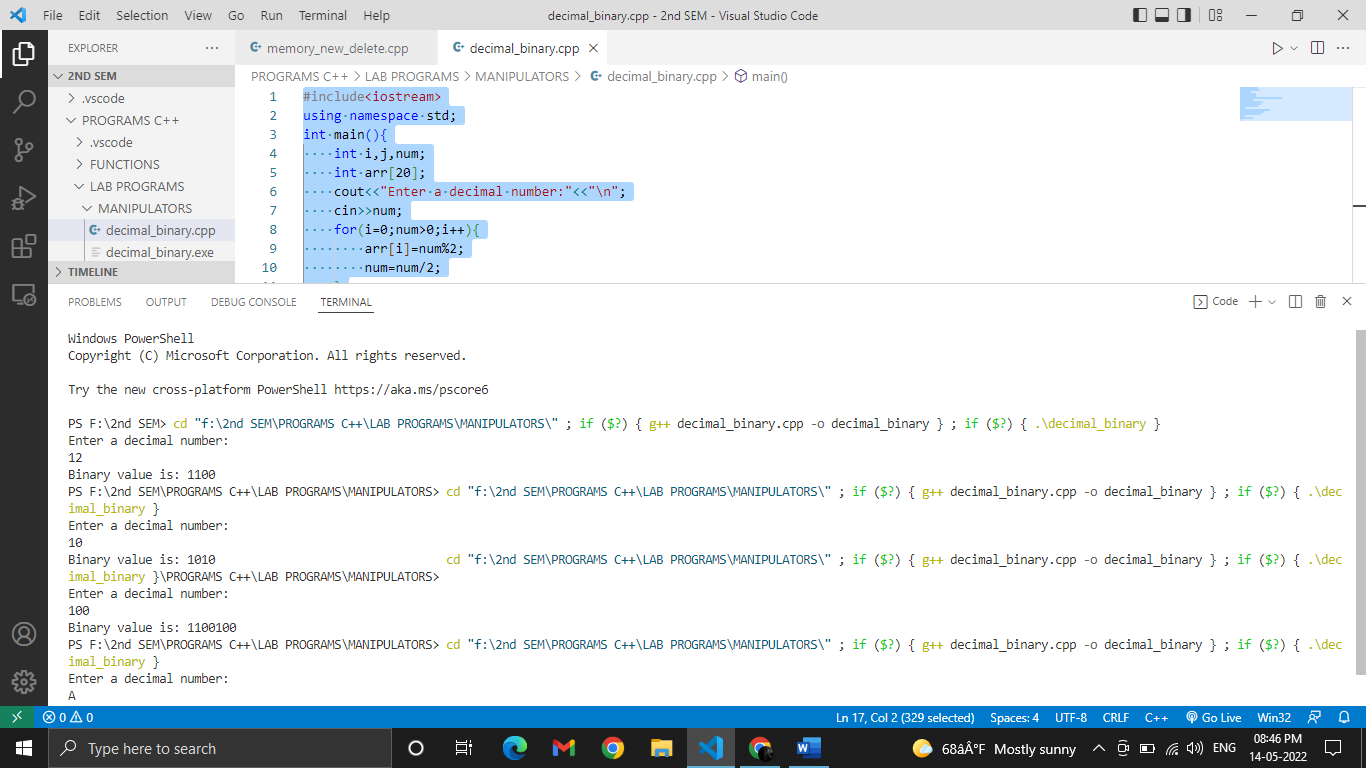
    for(j=i-1;j>=0;j--){

        cout<<arr[j];

    }

    return 0;

}

**OUTPUT**

**EXPERIMENT-8**

**AIM:** WAP convert decimal to hexadecimal using array.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

    int i,j,num;

    char arr[20];

    cout<<"Enter a decimal number:"<<"\n";

    cin>>num;

    for(i=0;num>0;i++){

        arr[i]=num%16;

        if(arr[i]<10){

            arr[i]=arr[i]+48;

        }

        else{

            arr[i]=arr[i]+55;

        }

        num=num/16;

    }

    cout<<"Hexadecimal value is: ";

    for(j=i-1;j>=0;j--){

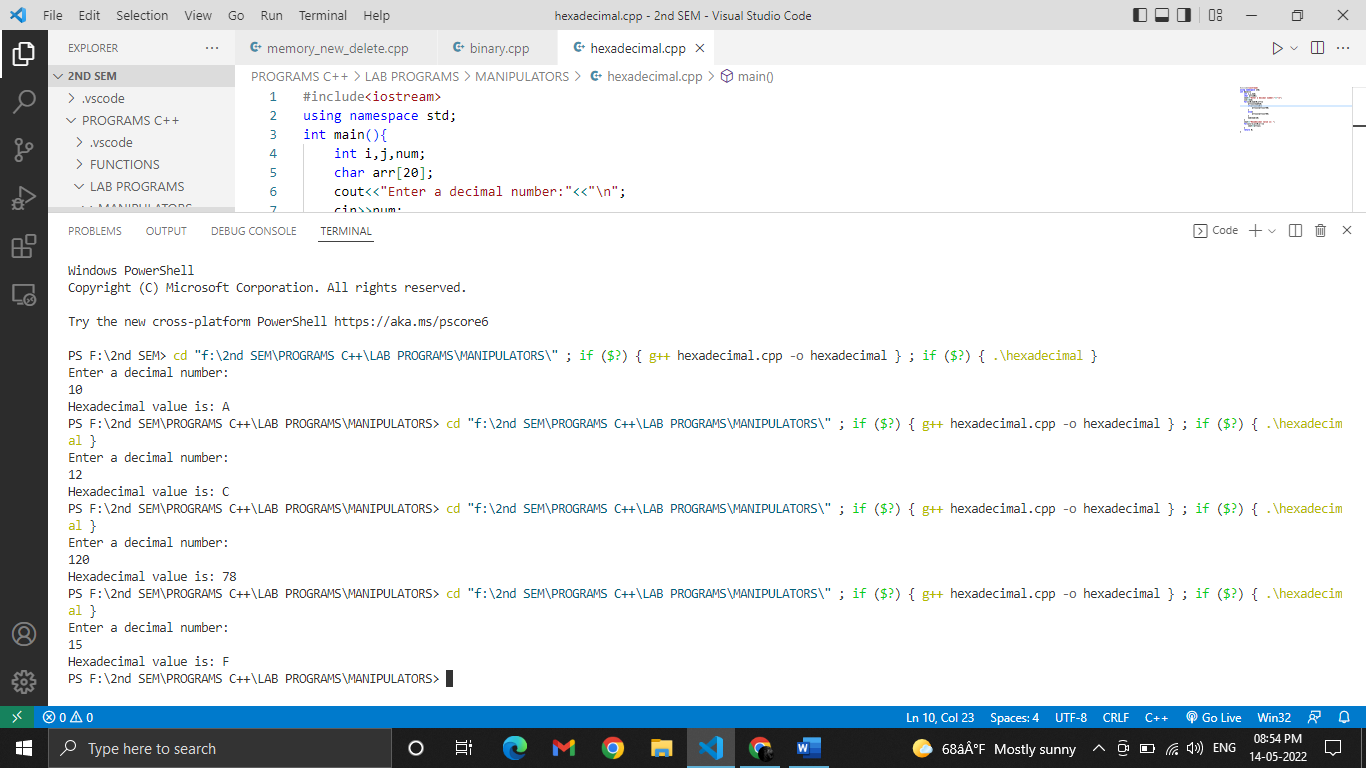
        cout<<arr[j];

    }

    return 0;

}

**OUTPUT:**



**EXPERIMENT-9**

**AIM:** WAP using memory management operators new and delete.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

    int i;

    int num;

    float \*ptr=new float[num];

    cout<<"Enter num:";

    cin>>num;

    cout<<"Enter elements of array:"<<"\n";

    for(i=0;i<num;i++)

    cin>>\*(ptr+i);

    cout<<"\n"<<"Entered elements are"<<"\n";

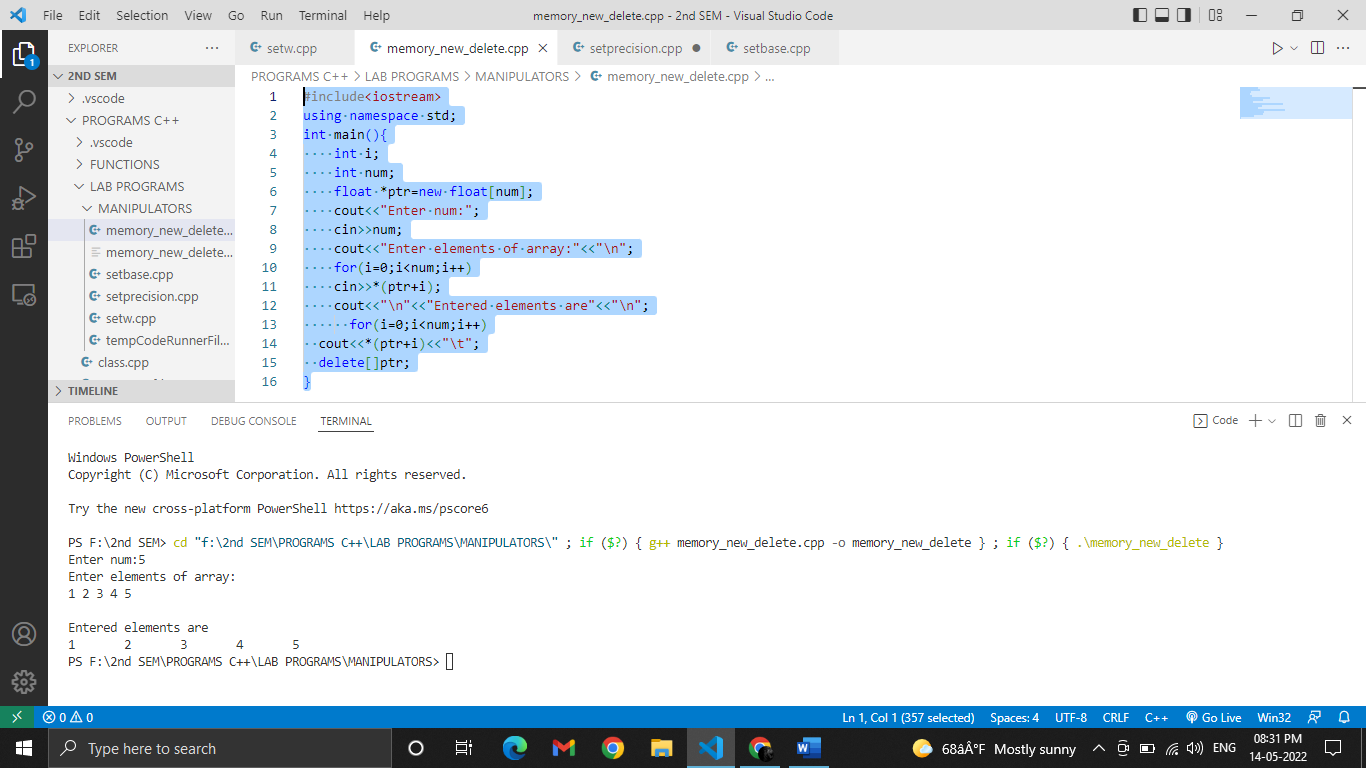
      for(i=0;i<num;i++)

  cout<<\*(ptr+i)<<"\t";

  delete[]ptr;

}

**OUTPUT:**



**EXPERIMENT-10**

**AIM:** WAP to implement manipulators setw, setprecision and setbase.

**SOFTWARE USED:** VS code

**PROGRAM:**

**//setw**

#include<iostream>

#include<iomanip>

using namespace std;

int main(){

    int num;

    cout<<"Enter any number:";

    cin>>num;

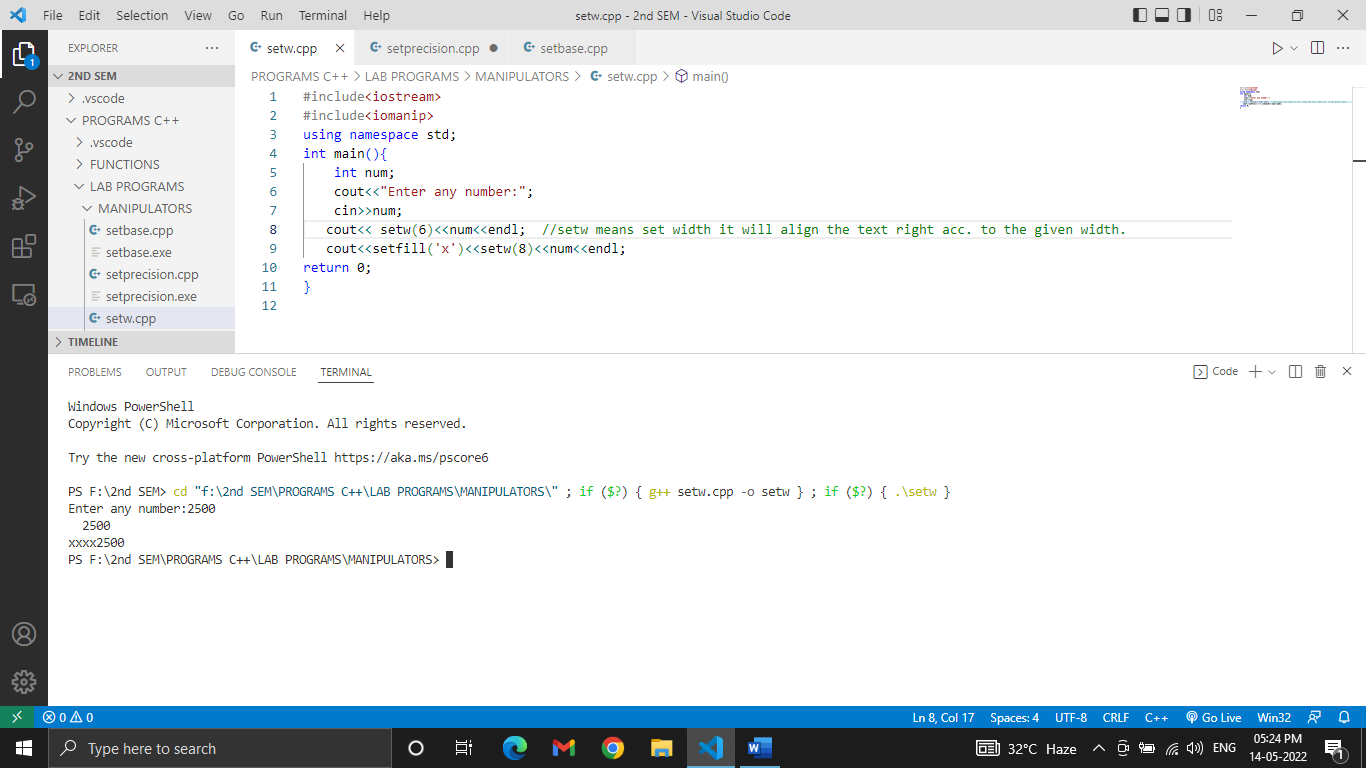
   cout<< setw(6)<<num<<endl;

   cout<<setfill('x')<<setw(8)<<num<<endl;

return 0;

}

**OUTPUT:**



**PROGRAM:**

**//setprecision**

#include<iostream>

#include<iomanip>

using namespace std;

int main(){

    float num;

    cout<<"Enter any number:";

    cin>>num;

    cout<<"Significant digit= "<< setprecision(3)<<num<<endl

    cout<<"Total Digits after decimal= "<<fixed<<setprecision(3)<<num;

return 0;

}

**OUTPUT:**



**PROGRAM:**

**//setbase**

#include<iostream>

#include<iomanip>

using namespace std;

int main(){

    int num;

    cout<<"Enter any number:";

    cin>>num;

   cout<<"DECIMAL= "<<setbase(10)<<num<<"\n";

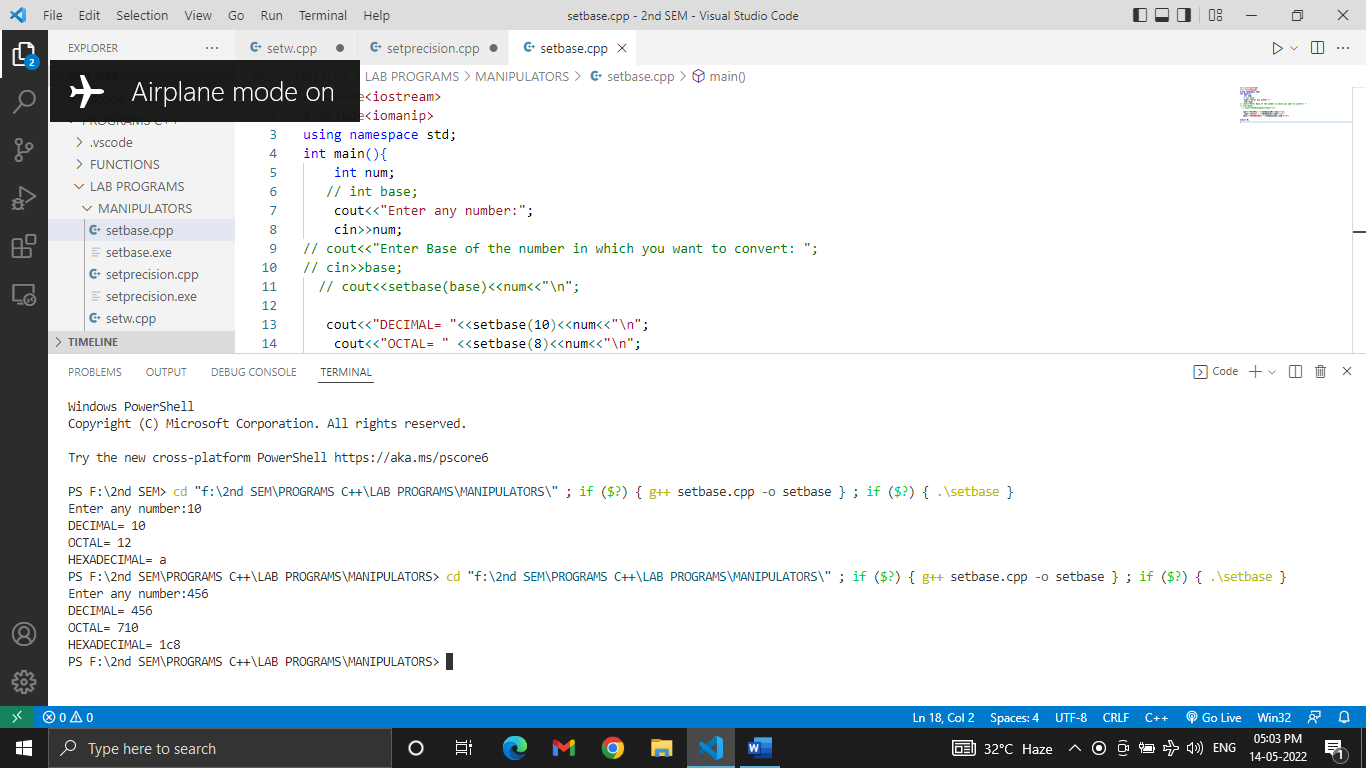
    cout<<"OCTAL= " <<setbase(8)<<num<<"\n";

   cout<<"HEXADECIMAL= "<<setbase(16)<<num<<"\n";

return 0;

}

**OUTPUT:**



**EXPERIMENT-11**

**AIM:** WAP to demonstrate Lamda expression calculating sum.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

int main(){

    int a=20,b=30;

    auto sum=[](int a, int b){

    return a+b;

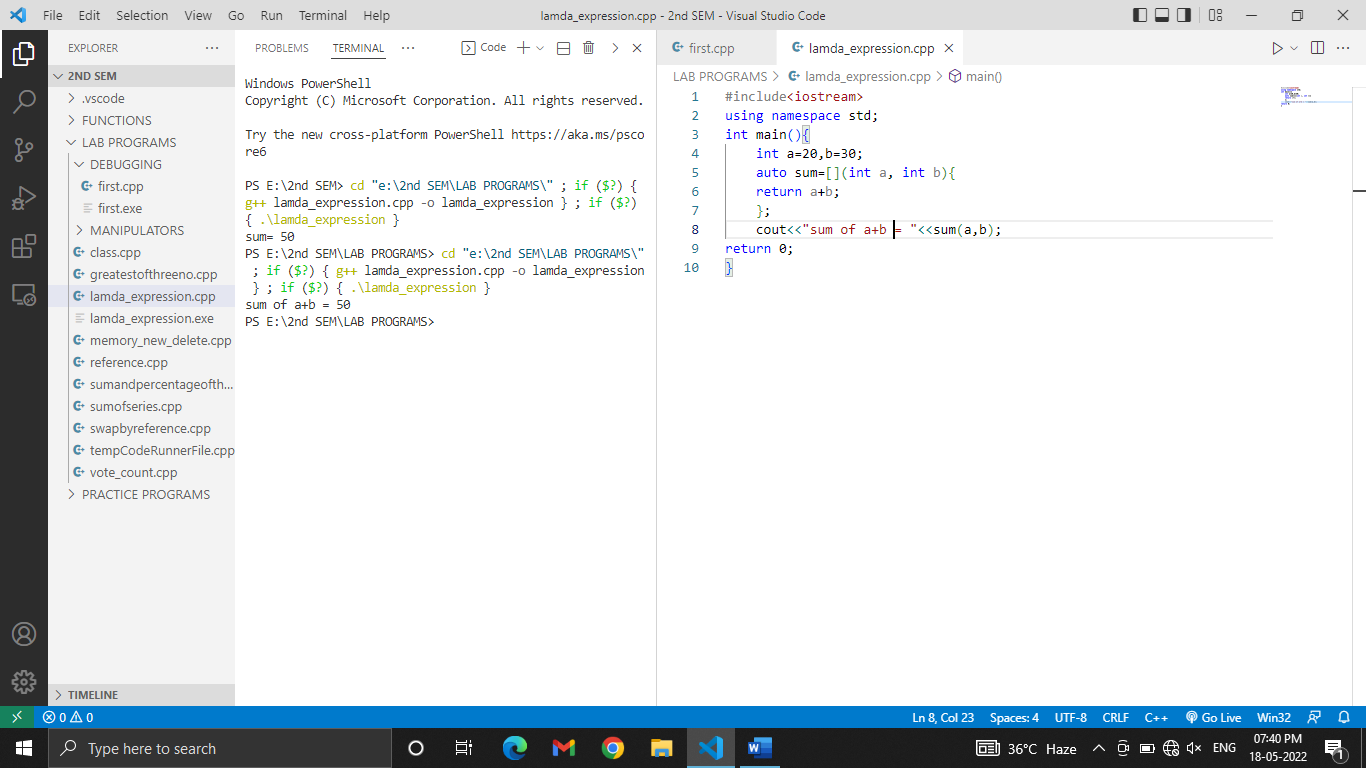
    };

    cout<<"sum= "<<sum(a,b);

return 0;

}

**OUTPUT:**



**EXPERIMENT-12**

**AIM:** WAP to demonstrate folding expressions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

auto sum(auto...val)

{

    int res;

    res=(...+val);

 return res;

}

int main(){

    int a,b;

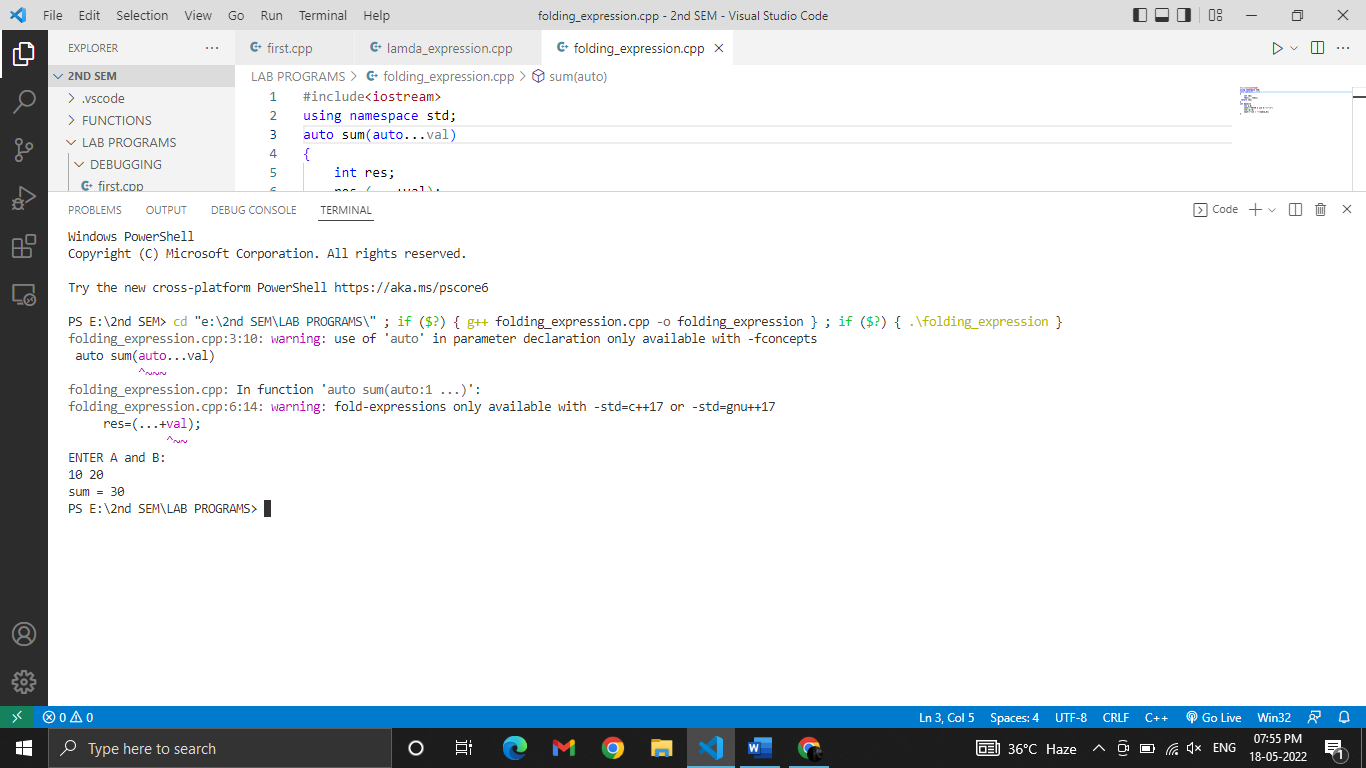
    cout<<"ENTER A and B:"<<"\n";

    cin>>a>>b;

    cout<<"sum = "<<sum(a,b);

}

**OUTPUT:**



**EXPERIMENT-13**

**AIM:** WAP to classify a character as alphanumeric, digit, alphabet, uppercase and

lowercase.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

#include<cctype>

#include<cstring>

#include<stdlib.h>

using namespace std;

int main(){

    char ch[100];

   cout<<"Enter the data to check for alphanumeric\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        if(isalnum(ch[i]))

        cout<<ch[i]<<" is alphanumeric"<<endl;

        else

         cout<<ch[i]<<" is not alphanumeric"<<endl;

}

cout<<"Enter the data to check for digit\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        if(isdigit(ch[i]))

        cout<<ch[i]<<" is digit"<<endl;

        else

         cout<<ch[i]<<" is not digit"<<endl;

}

cout<<"Enter the data to check for alphabet\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        if(isalpha(ch[i]))

        cout<<ch[i]<<" is alphabet"<<endl;

        else

         cout<<ch[i]<<" is not alphabet"<<endl;

}

cout<<"Enter the data to check for lowercase\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        if(islower(ch[i]))

        cout<<ch[i]<<" is in lowercase"<<endl;

        else

         cout<<ch[i]<<" is not in lowercase"<<endl;

}

cout<<"Enter the data to check for uppercase\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        if(isupper(ch[i]))

        cout<<ch[i]<<" is in uppercase"<<endl;

        else

         cout<<ch[i]<<" is not in uppercase"<<endl;

}

cout<<"Enter the data to convert into uppercase\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        cout<<toupper((ch[i]))<<" ";

}

  cout<<endl;

cout<<"Enter the data to convert into lowercase\n";

    cin>>ch;

    for(int i=0;i<strlen(ch);i++){

        cout<<tolower((ch[i]))<<" "<<endl;

}

Cout<<"Random Numbers are:\n";

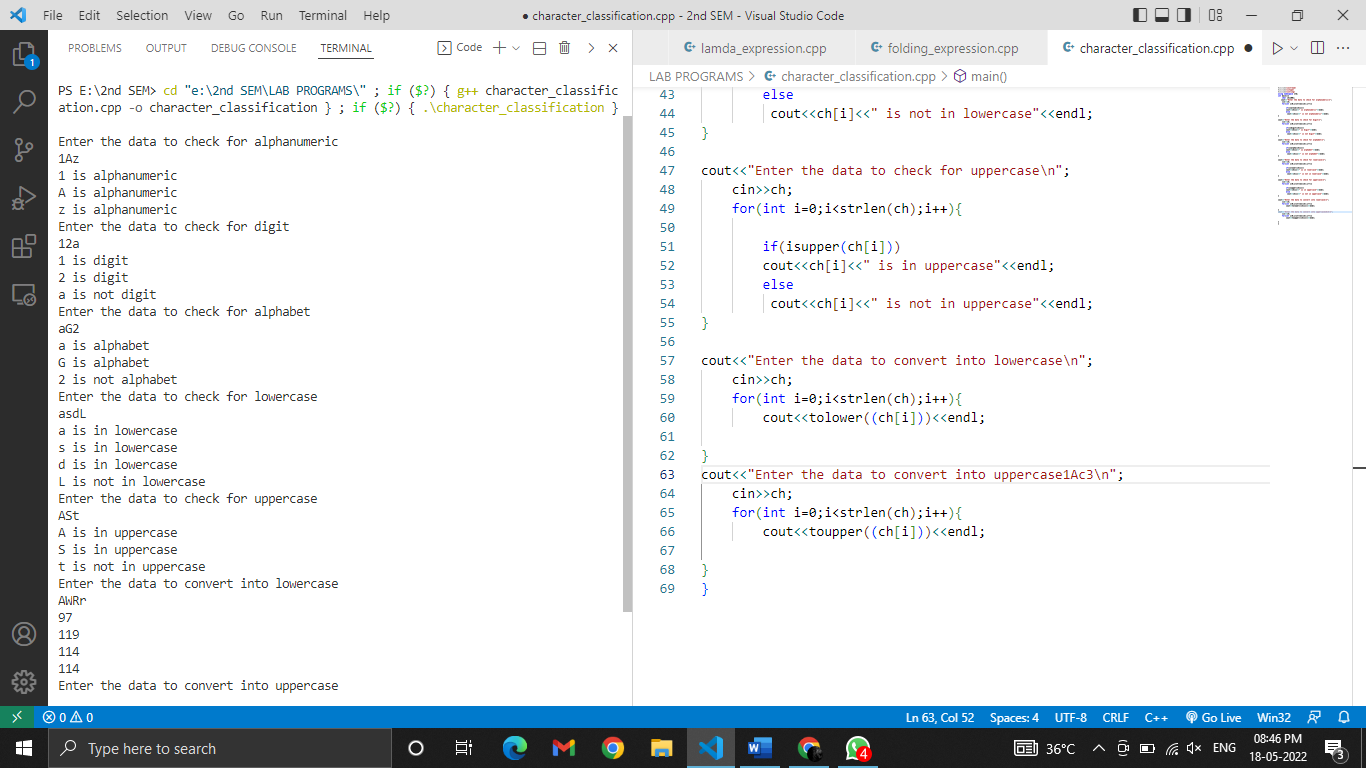
for(int i=0;i<5;i++){

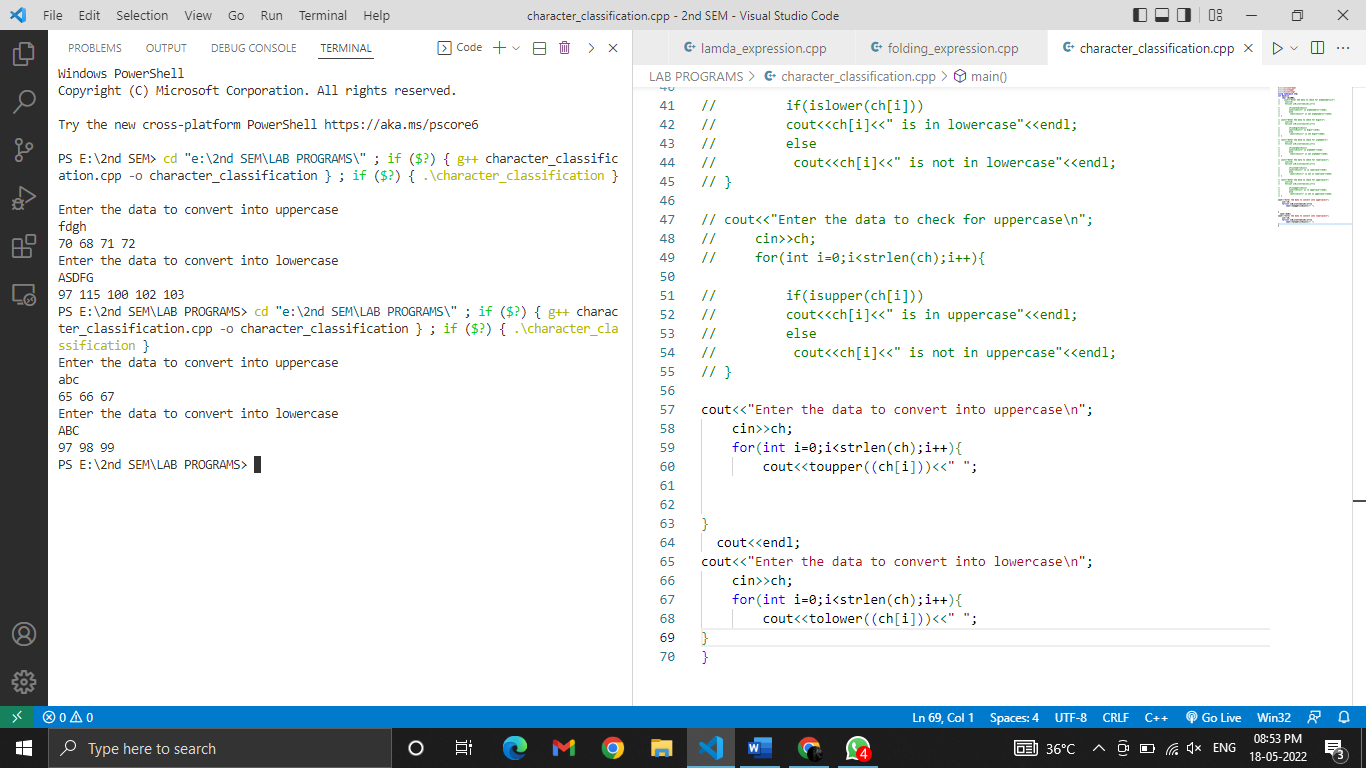
cout<<rand()<<" ";

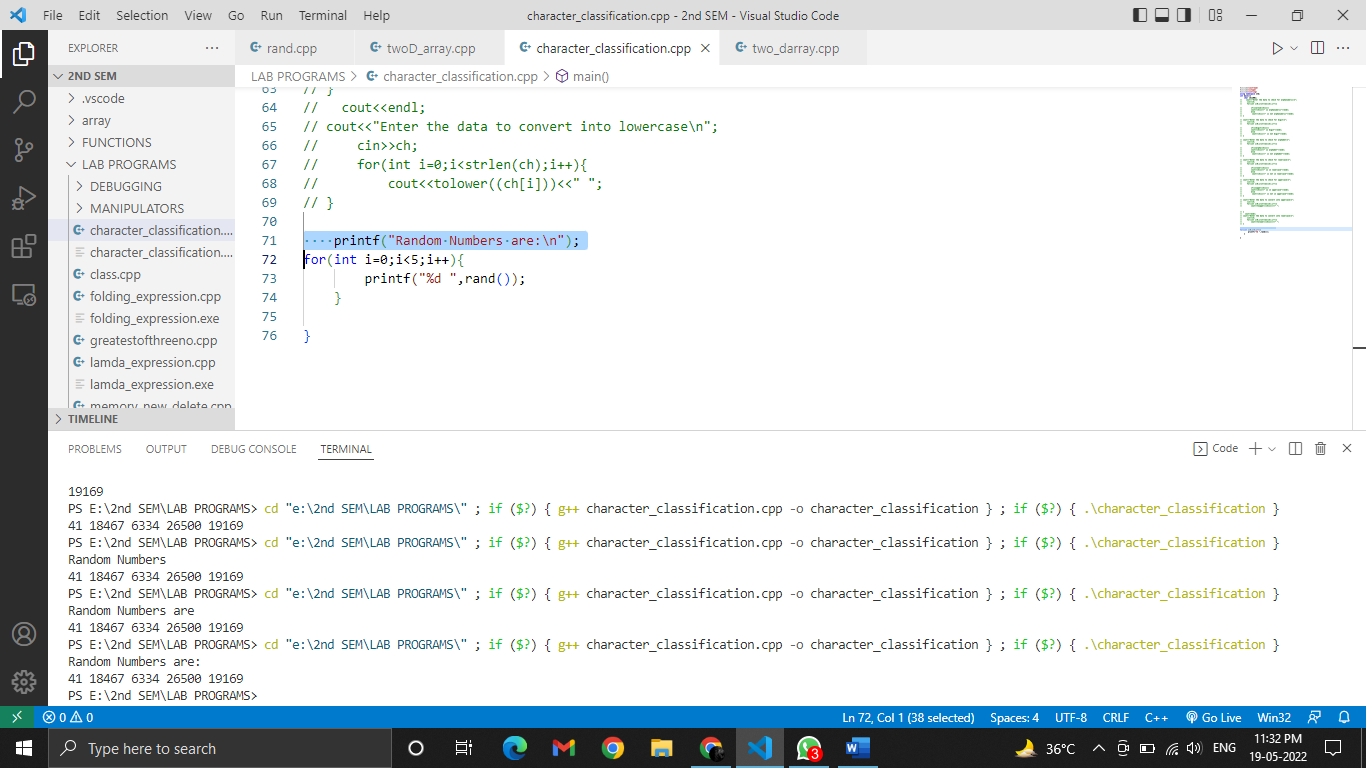
}

}

**OUTPUT:**







**EXPERIMENT-14**

**AIM:** WAP to calculate voting percentage of 3 candidates in 5 regions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

class voting\_candidate{

   char name[20];

   int id;

    int reg1;

    int reg2;

    int reg3;

    int reg4;

    int reg5;

float percentage;

    public:

    void getdata();

    void calcper();

    void putdata();

};

void voting\_candidate::getdata()

{

    cout<<"\nEnter the Name of candidate:"<<endl;

   cin>> name;

   cout<<"Enter id of candidate"<<endl;

   cin>>id;

   cout<<"Enter Number of votes in 5 regions"<<endl;

   cin>> reg1>> reg2>> reg3>> reg4>> reg5;

}

void voting\_candidate::calcper()

{

   // int reg1\_vote, reg2\_vote,reg3\_vote, reg4\_vote,reg5\_vote;

    percentage = (reg1 + reg2 + reg3 + reg4 + reg5)/5.0;

}

void voting\_candidate::putdata()

{

cout<<"Percentage is: "<<percentage<<endl;

}

int main(){

    voting\_candidate candidate1;

    voting\_candidate candidate2;

    voting\_candidate candidate3;

     candidate1.getdata();

     candidate1.calcper();

     candidate1.putdata();

     candidate2.getdata();

     candidate2.calcper();

     candidate2.putdata();

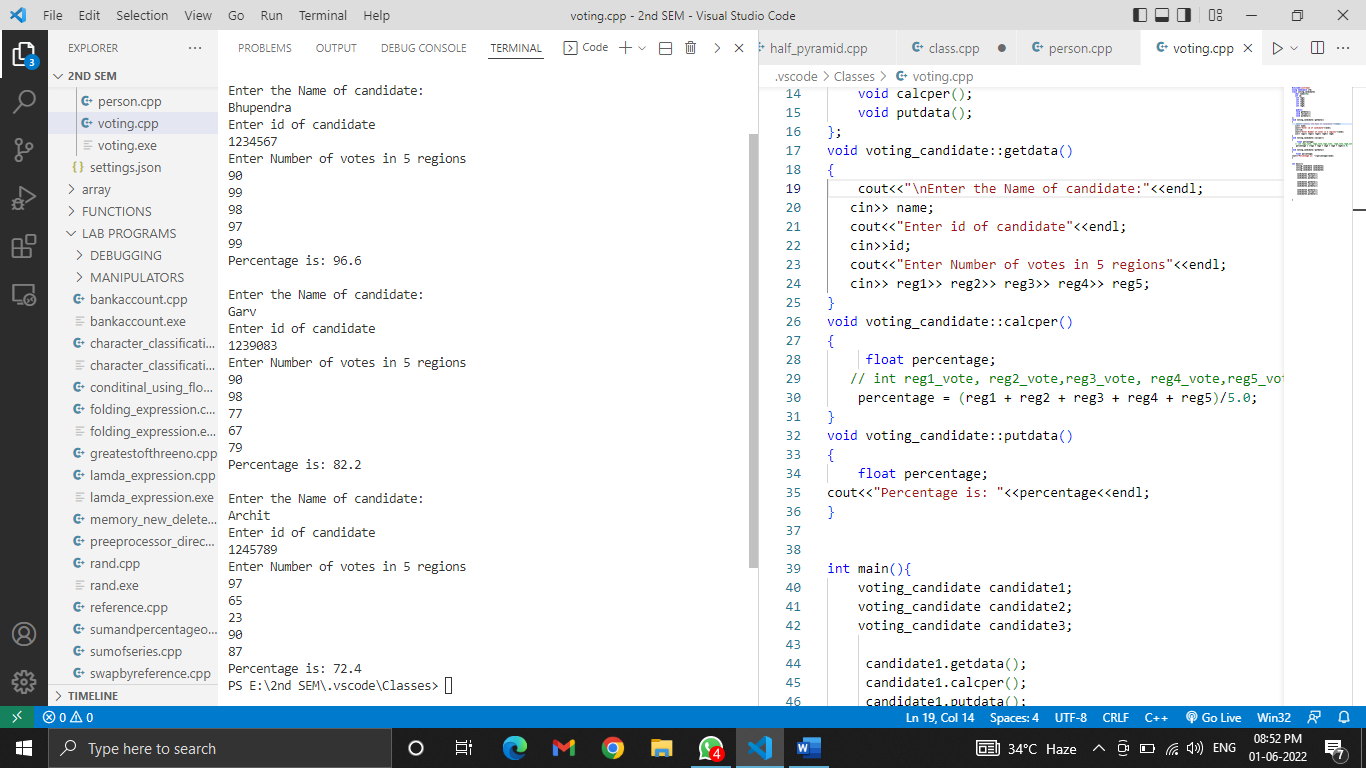
     candidate3.getdata();

     candidate3.calcper();

     candidate3.putdata();

}

**OUTPUT:**



**EXPERIMENT-15**

**AIM:** Define a class to represent a bank account & Include the given members

Write a main program to test the program.

**SOFTWARE USED:** VS code

**PROGRAM:**

 #include<iostream>

using namespace std;

class bank\_ac{

char name[10];

int acc\_no;

char type[10];

float balance;

public:

 void getdata();

void deposit(int dep);//passing arguments

void withdraw(int withdraw);//passing arguments

void putdata();

};

void bank\_ac::getdata(){

cout<<"Enter name"<<endl;

cin>>name;

cout<<"Enter account no."<<endl;

cin>>acc\_no;

cout<<"Account type"<<endl;

cin>>type;

cout<<"Enter total balance"<<endl;

cin>>balance;

}

void bank\_ac::deposit(int dep)

{

  //int deposit;   //without arguments

 // cout<<"Enter amount to deposit"<<endl;

  //cin>>deposit;

  balance+=dep;

    cout<<"Deposited amount is"<<dep;

}

void bank\_ac::withdraw(int withdraw)

{

//  int withdraw;

  //cout<<"\nEnter amount to withdraw"<<endl;

  //cin>>withdraw;

  if(withdraw>balance)

    cout<<"cant't withdraw";

  else

  balance-=withdraw;

      cout<<"\nWithdraw amount is: "<<withdraw;

}

void bank\_ac::putdata(){

cout<<"\nYour account details are:"<<endl;

cout<<"Name: ";

cout<<name;

cout<<"\nAccount no.: ";

cout<<acc\_no;

cout<<"\nAccount type: ";

cout<<type;

cout<<"\nTotal balance: ";

cout<<balance;

}

int main(){

bank\_ac obj;

obj.getdata();

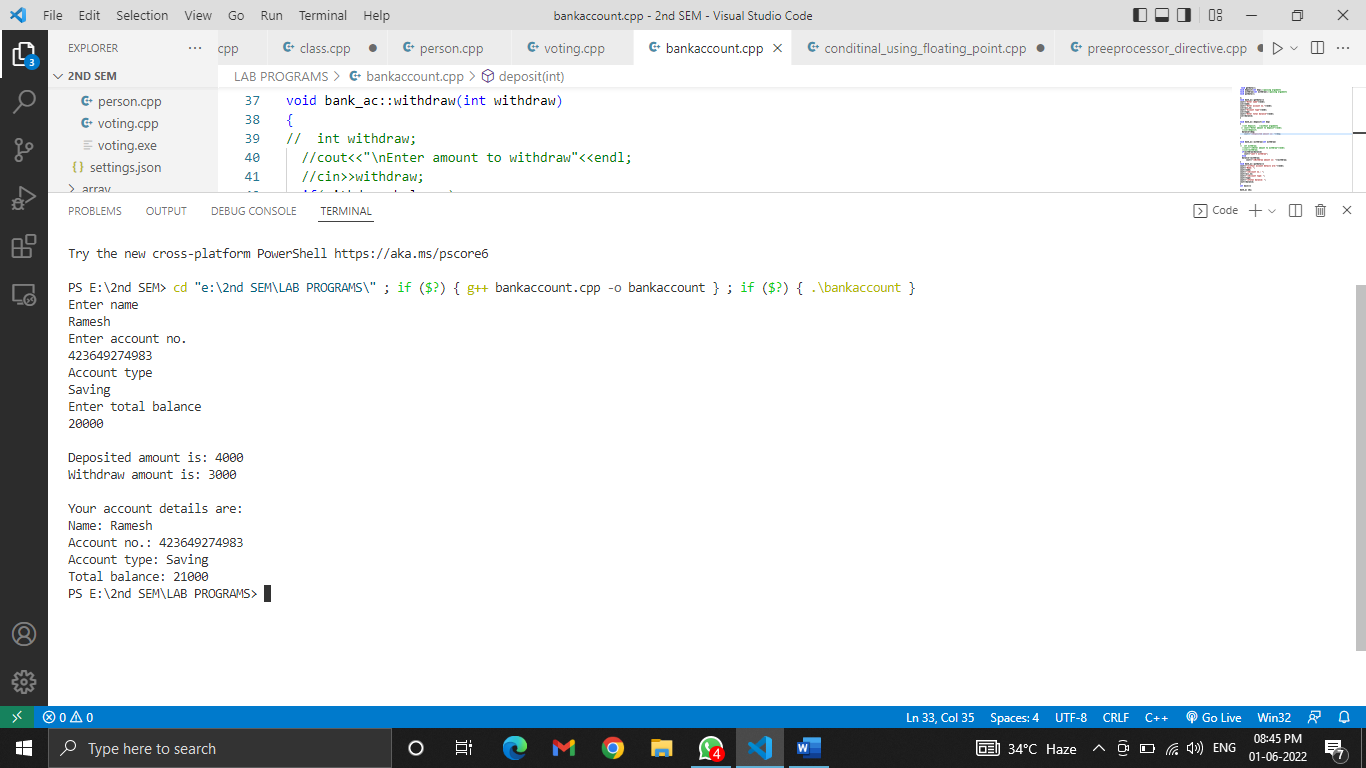
obj.deposit(4000);

obj.withdraw(3000);

obj.putdata();

}

**OUTPUT:**



**EXPERIMENT-16**

**AIM:** WAP to demonstrate a pre-processor directive calculating the sum of square of two numbers.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

#define square(n1,n2)((n1\*n1)+(n2\*n2))

int main(){

    int n1,n2;

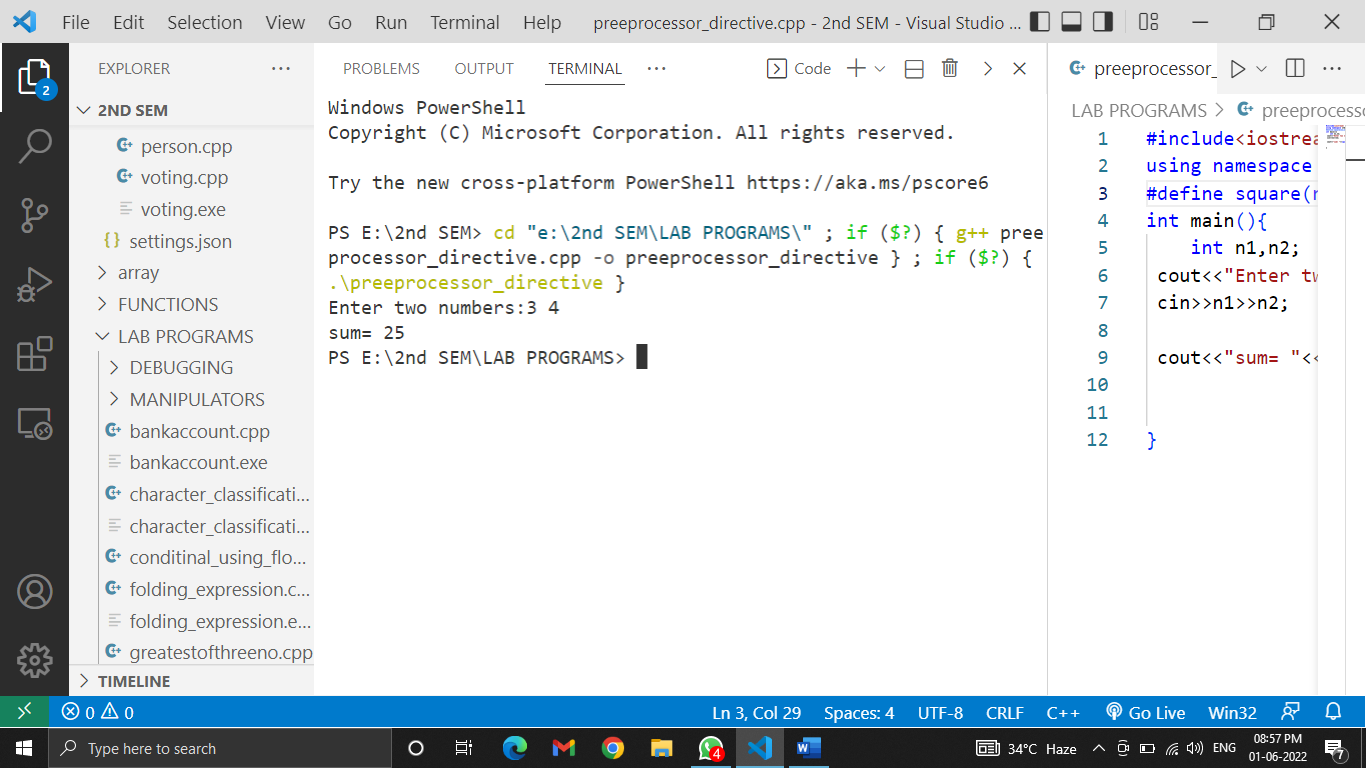
 cout<<"Enter two numbers:";

 cin>>n1>>n2;

 cout<<"sum= "<<square(n1,n2)<<endl;

}

**OUTPUT:**



**EXPERIMENT-17**

**AIM:** Five voters are contesting in three regions. Each voter has name, id, total no of votes obtained in respective regions. Determine using a C++ program, which voter received the maximum voting percentage combinedly in three regions using the concept of array of objects.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class election{

char name[20];

int id;

int reg1,reg2,reg3;

public:

    float per;

    void getdata();

    void calcdata();

    void display();

};

void election:: getdata(){

    cout<<"\nEnter details:\n";

    cout<<"Name: ";

    cin>>name;

    cout<<"Id: ";

    cin>>id;

    cout<<"Enter Votes in 3 regions:\n";

cin>>reg1>>reg2>>reg3;

}

void election :: calcdata(){

per = (reg1+reg2+reg3)/3.0;

}

void election :: display(){

cout<<"\nEntered details are:\n";

cout<<"Name: "<<name<<endl;

cout<<"Id: "<<id<<endl;

cout<<"Percentage is:  "<<per<<endl;

}

int main(){

    election cand[5];

    int arr[20];

    for(int i=0;i<5;i++){

        cand[i].getdata();

        arr[i]=cand[i].per;

    }

    for(int i=0;i<5;i++){

        cand[i].calcdata();

        arr[i]=cand[i].per;

        cand[i].display();

    }

    float high=arr[0];

    for(int i=0;i<5;i++){

        if(high<arr[i]){

            high=arr[i];

        }

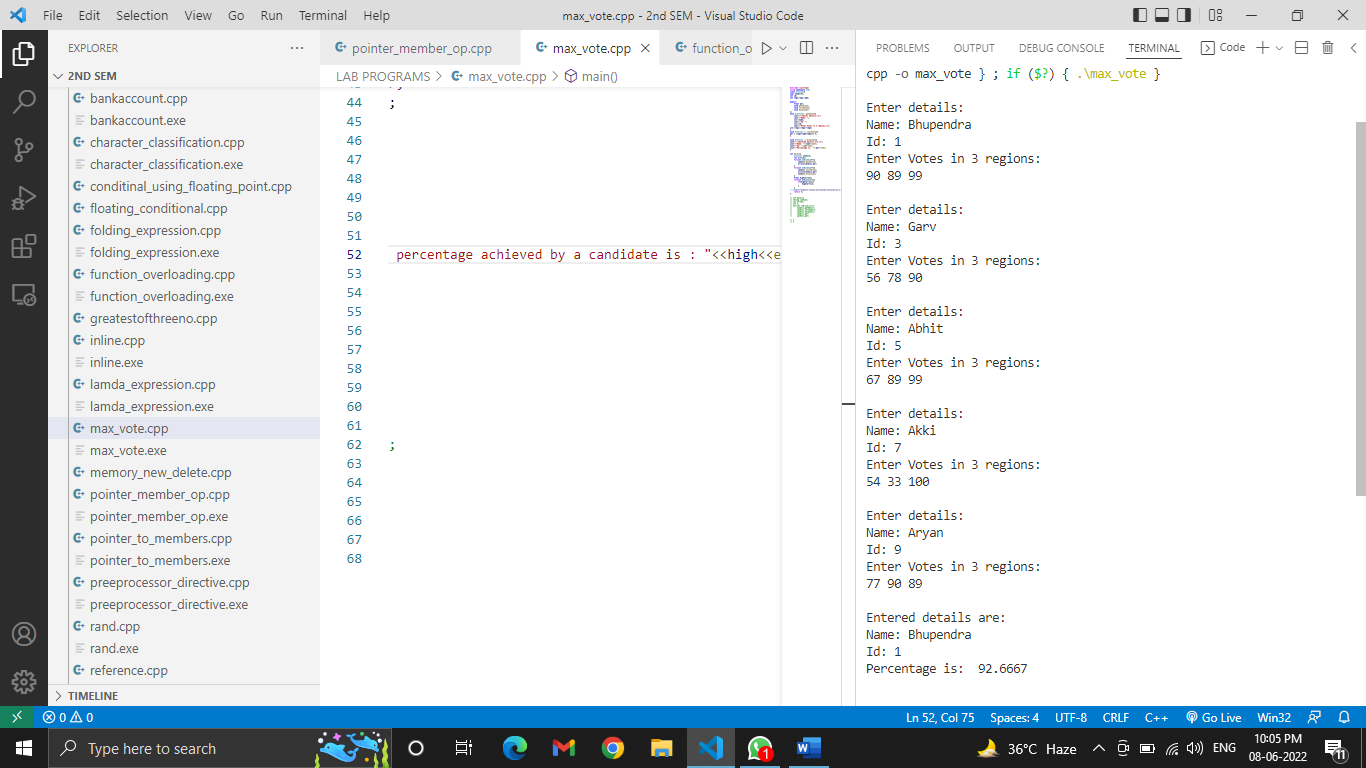
    }

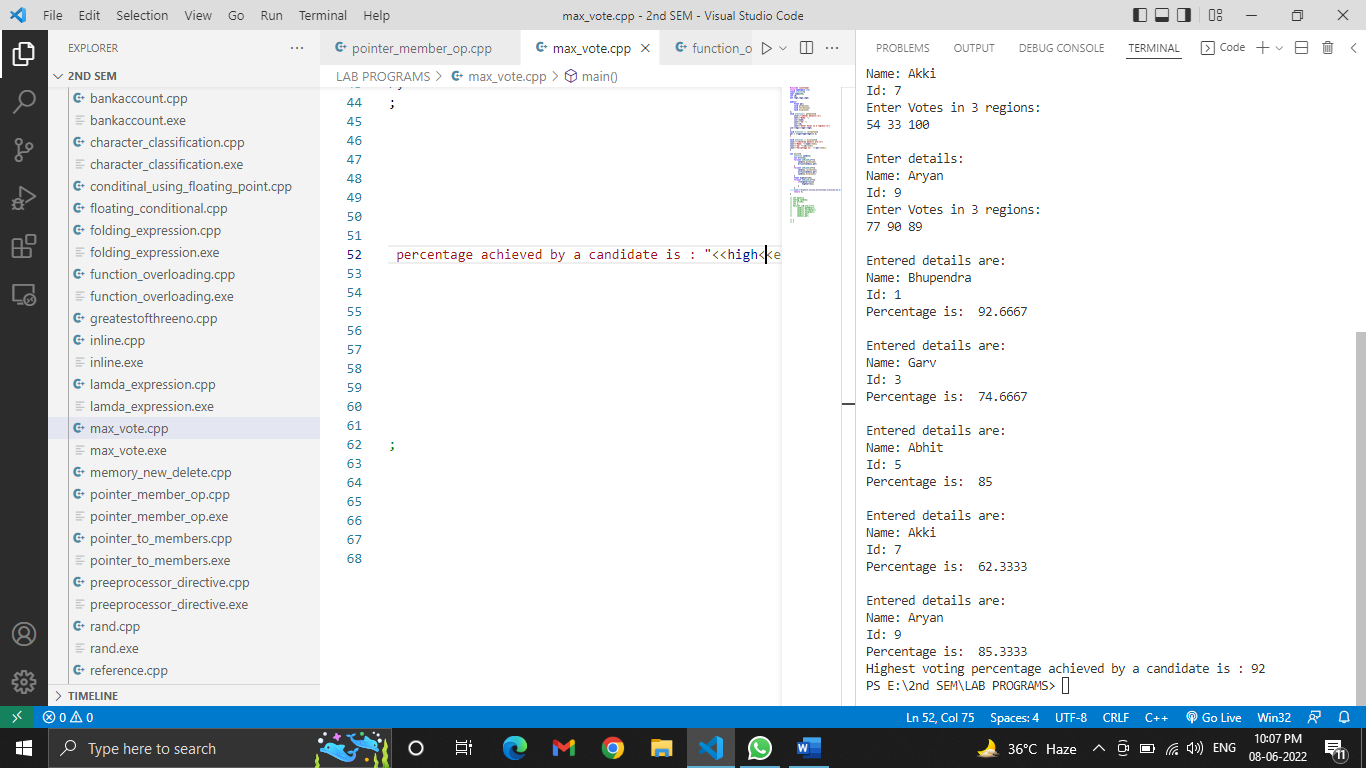
    cout<<"Highest voting percentage achieved by a candidate is : "<<high<<endl;

    return 0;

}

**OUTPUT:**



 **EXPERIMENT-18**

**AIM:** WAP to access the class members using pointer to member operators.

Access to be done using both .\* And ->\*

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class test{

int x,y;

public:

    int z=6;

    void setdata(int a, int b){

    x=a;

    y=b;

 cout<<x<<" "<<y<<endl;

}

void sum(int x,int y,int z){

cout<<(x+y+z);

}

};

int main(){

test obj;

int test::\*p1=&test::z;

test\*p2=&obj;

void (test::\*p3)(int a ,int b)=&test::setdata;

void (test::\*p4)(int ,int ,int)=&test::sum;

cout<<"Accessing member using object & pointer to the member z: ";

cout<<obj.\*p1<<endl;

cout<<"Accessing member using pointer to object & pointer to the member: ";

cout<<obj.\*p1<<endl;

cout<<"Accessing member using pointer to member function: ";

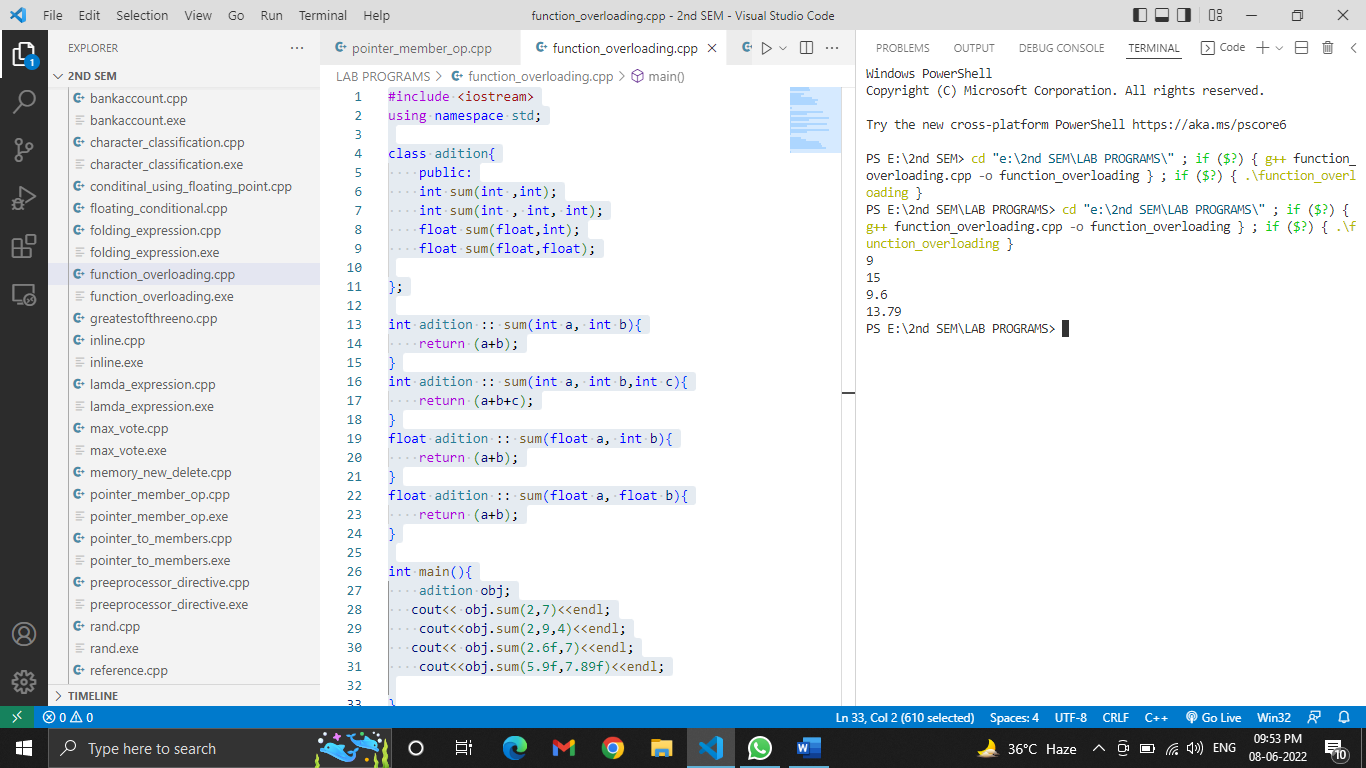
(obj.\*p3)(6,9);

cout<<"Accessing member using pointer to member function: ";

(obj.\*p4)(5,8,9);

}

**OUTPUT:**



**EXPERIMENT-19**

**AIM:** WAP to overload class member functions on number and type of arguments.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class adition{

    public:

    int sum(int ,int);

    int sum(int , int, int);

    float sum(float,int);

    float sum(float,float);

};

int adition :: sum(int a, int b){

    return (a+b);

}

int adition :: sum(int a, int b,int c){

    return (a+b+c);

}

float adition :: sum(float a, int b){

    return (a+b);

}

float adition :: sum(float a, float b){

    return (a+b);

}

int main(){

    adition obj;

   cout<< obj.sum(2,7)<<endl;

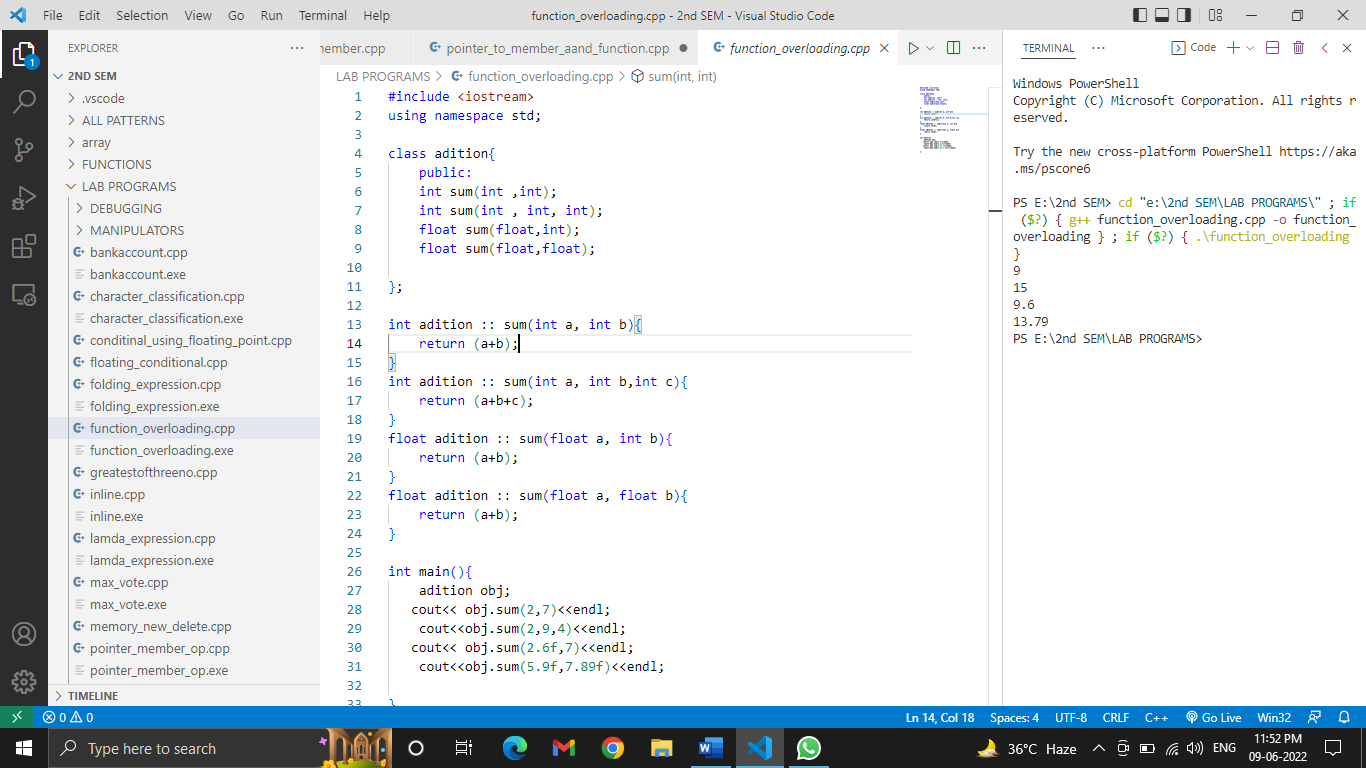
    cout<<obj.sum(2,9,4)<<endl;

   cout<< obj.sum(2.6f,7)<<endl;

    cout<<obj.sum(5.9f,7.89f)<<endl;

}

**OUTPUT:**



**EXPERIMENT-20**

**AIM:** WAP to demonstrate inline functions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class A {

    public:

    int add(int,int);

    int sub(int, int);

    float div(int,int);

    int mul(int,int);

};

inline int A:: add(int a, int b){

    return a+b;

}

inline int A:: sub(int a, int b){

    return a-b;

}

inline float A:: div(int a, int b){

    return a/b;

}

inline int A:: mul(int a, int b){

    return a\*b;

}

int main(){

A a;

   cout<<"Adition= "<<a.add(3,8)<<endl;

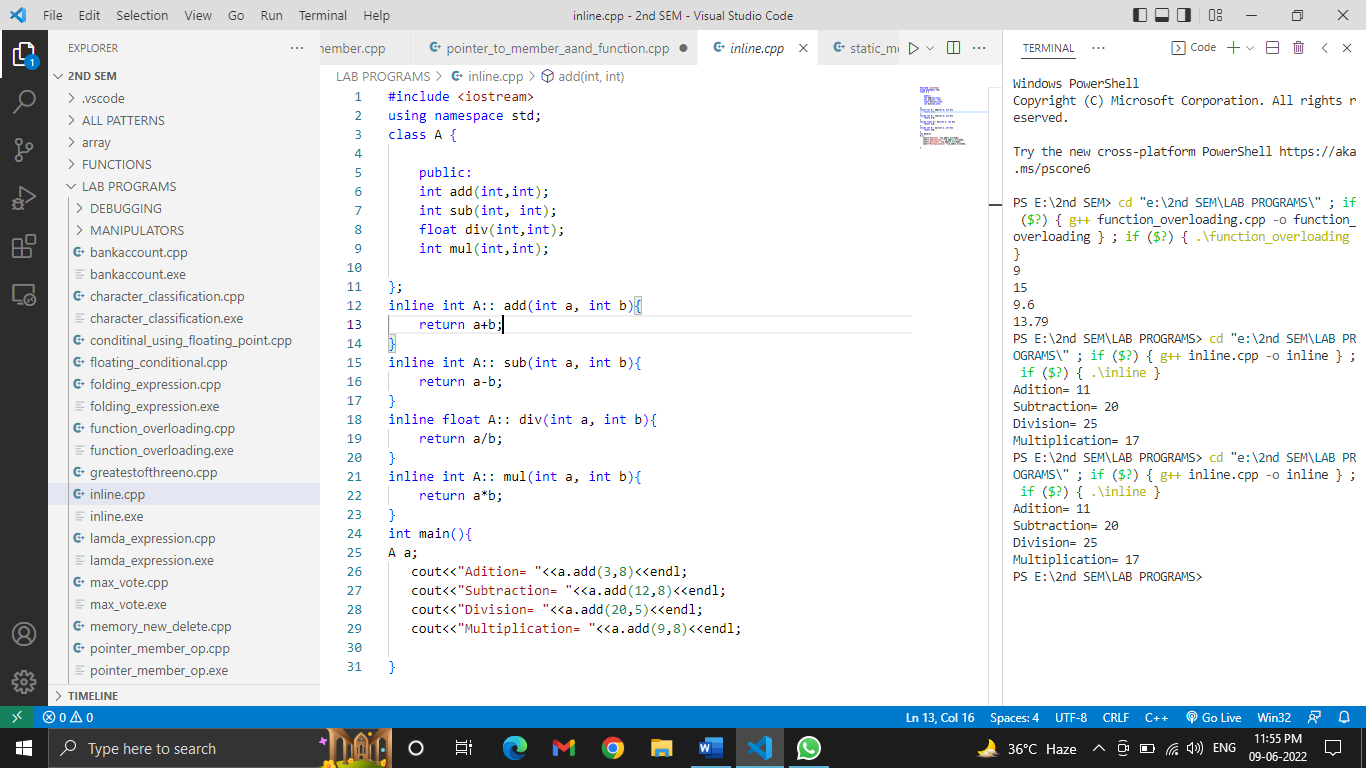
   cout<<"Subtraction= "<<a.add(12,8)<<endl;

   cout<<"Division= "<<a.add(20,5)<<endl;

   cout<<"Multiplication= "<<a.add(9,8)<<endl;

}

**OUTPUT:**



**EXPERIMENT-21**

**AIM:** WAP to demonstrate static data members & static member functions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class item{

    static int count;

    int no;

    public:

    void getdata(int x){

      int  no=x;

     count++;

     cout<<"Number is: "<<no<<endl;

    }

    static void showcount(){

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

    }

    void display(){

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

    }

};

int item :: count;

int main(){

item T1,T2,T3,obj1,obj2;

cout<<"Value through static data member:\n";

obj1.getdata(8);

obj1.display();

obj2.getdata(10);

obj2.display();

cout<<"\nValue through static member function:\n";

T1.getdata(20);

item :: showcount();//by using class name

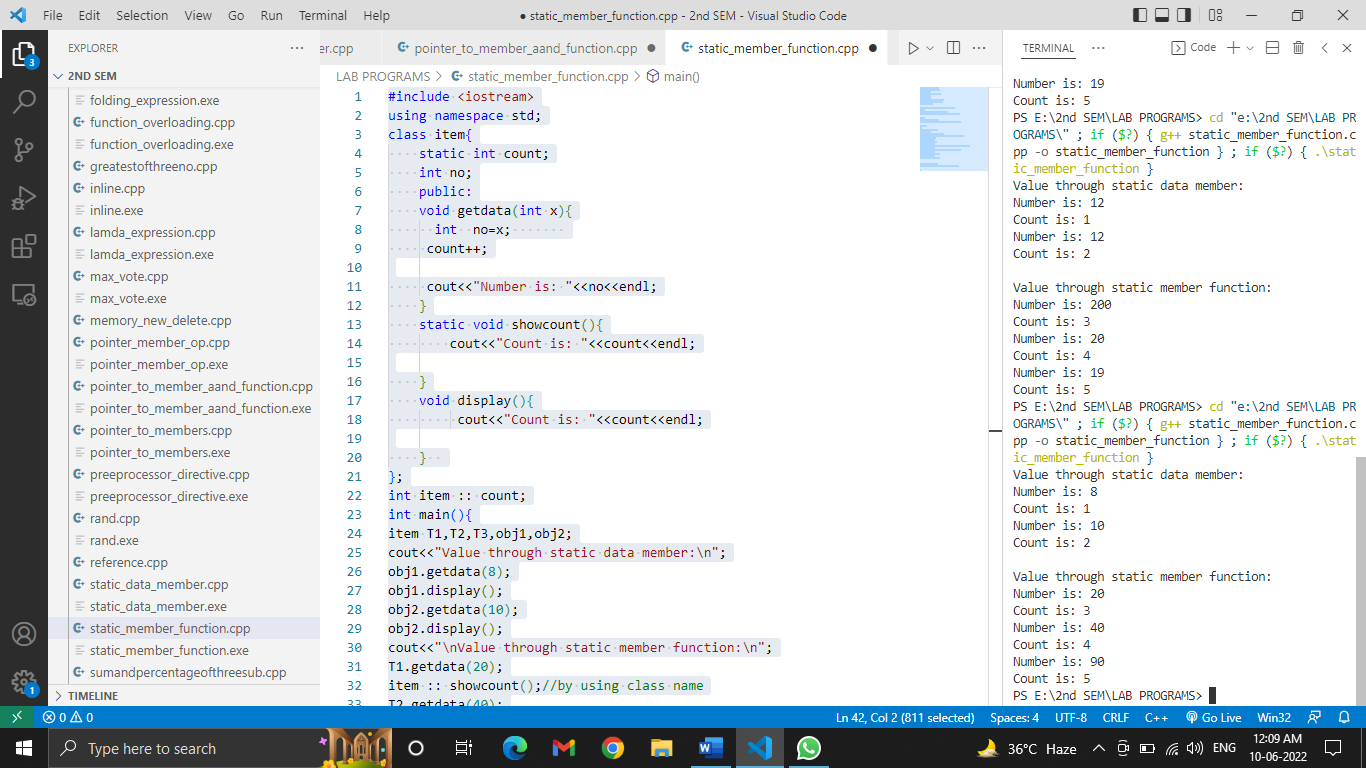
T2.getdata(40);

item :: showcount();

T3.getdata(90);

item :: showcount();

}

**OUTPUT:**

**EXPERIMENT-22**

**AIM:**  WAP to calculate the mean using friend functions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include<iostream>

using namespace std;

 class ClassB;

 class ClassA{

    int no1=8;

     friend float mean(ClassA,ClassB);//decleration

 };

  class ClassB{

    int no2=5;

     friend float mean(ClassA,ClassB);

 };

 float mean(ClassA objA,ClassB objB){    //definition

     return (objA.no1 + objB.no2)/2.0;

 }

 int main(){

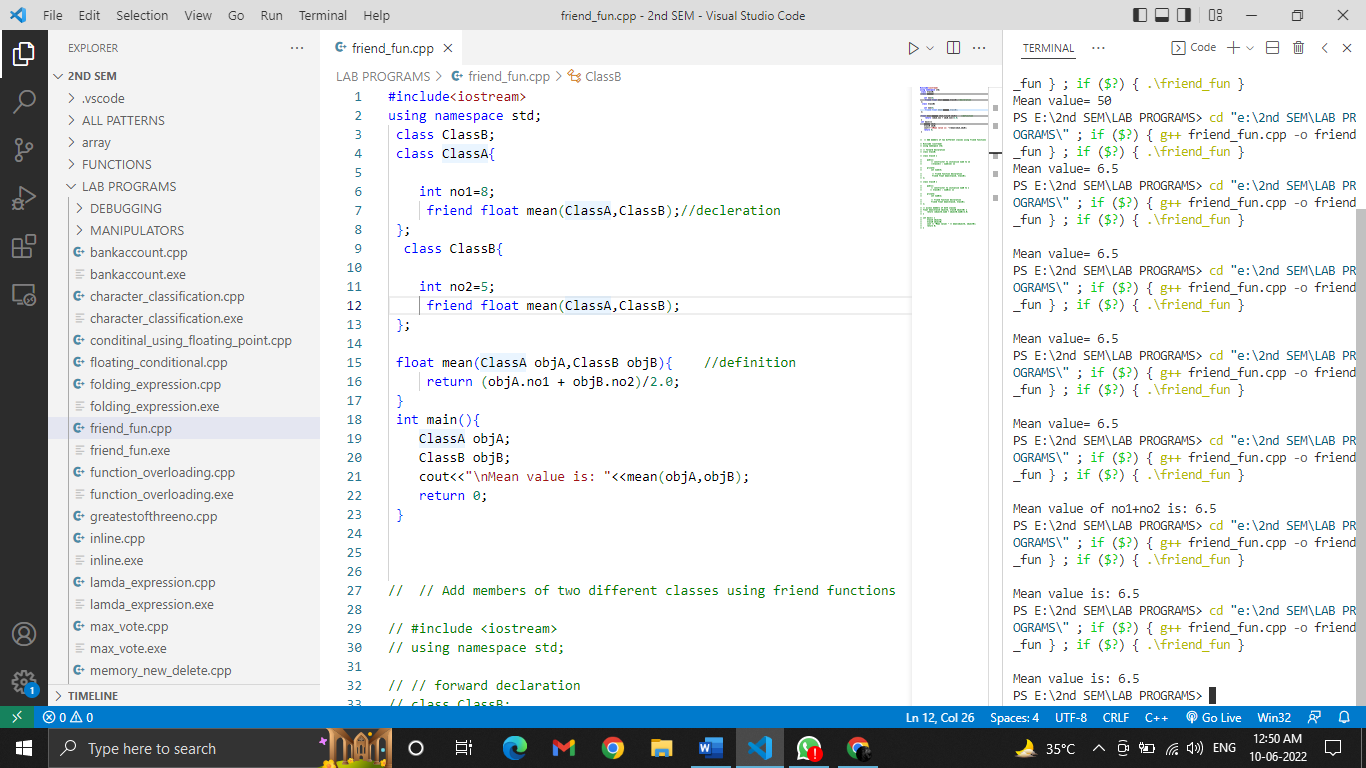
    ClassA objA;

    ClassB objB;

    cout<<"\nMean value is: "<<mean(objA,objB);

    return 0;

 }

**OUTPUT:**

**EXPERIMENT-23**

**AIM:** Write a friend function to calculate income tax for doctor, scientist, and teacher. Assume that tax for each the three professionals is 10% of total of hra, basic salary(bs) & dearness allowance(da).

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class scientist;

class teacher;

class doctor

{

    float hra,bs,da,total;

public:

    friend float incometax(doctor d,scientist s,teacher t);

    void getdata(){

        cout<<"Enter Doctor's hra,bs,da: ";

        cin>>hra>>bs>>da;

    }

};

class scientist

{

    float hra,bs,da,total;

public:

    friend float incometax(doctor d,scientist s,teacher t);

    void getdata(){

        cout<<"Enter Scientist hra,bs,da: ";

        cin>>hra>>bs>>da;

    }

};

class teacher

{

    float hra,bs,da,total;

public:

    friend float incometax(doctor d,scientist s,teacher t);

    void getdata(){

        cout<<"Enter Teacher hra,bs,da: ";

        cin>>hra>>bs>>da;

    }

};

float incometax(doctor d,scientist s,teacher t){

    d.total=(d.hra + d.bs + d.da);

    s.total=(s.hra + s.bs + s.da);

    t.total=(t.hra + t.bs + t.da);

    cout<<"Income tax(10% of total):- \n";

    cout<<"Doctor: "<<d.total\*0.1<<"\n";

    cout<<"Scientist: "<<s.total\*0.1<<"\n";

    cout<<"Teacher: "<<t.total\*0.1;

    return 0;

}

int main(){

    doctor d1; //declare diff class obj in multiple lines

    scientist s1;

    teacher t1;

    d1.getdata();

    s1.getdata();

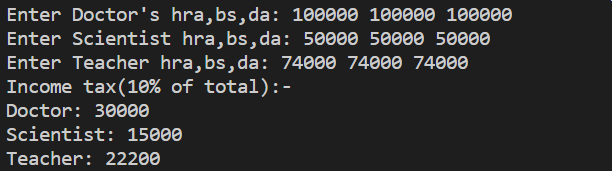
    t1.getdata();

    incometax(d1,s1,t1);

    return 0;

}

**OUTPUT:**



**EXPERIMENT-24**

**AIM:** Number Pattern

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

int main(){

    int i,j,k,l;

    for(i=0;i<4;i++){

        for(j=0;j<4-i;j++ ){

            cout<<" ";

        }

        int n=i+1;

        for(k=0;k<=i;k++){

            cout<<n;

            n++;

        }

        for(l=n-2;l>=i+1;l--){

            cout<<l;

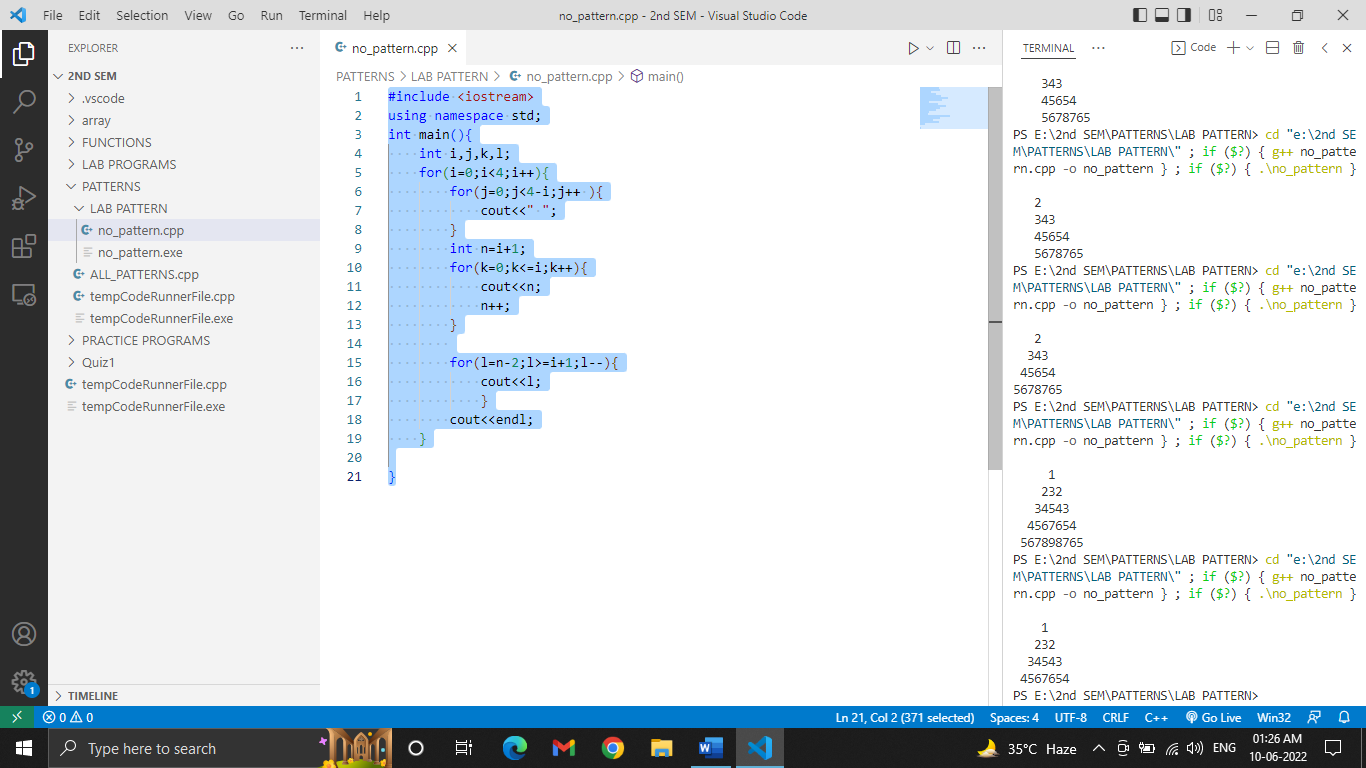
            }

        cout<<endl;

    }

return 0;

}

**OUTPUT:**

**EXPERIMENT-25**

**AIM:** Create a class COMPLEX with members as ‘real’ & ‘imag’. Write a program to add two complex numbers together. The function should receive two complex objects to be added. Set the real & imag part through a function setval () with default value as ‘2’ for the imaginary part.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class complexNo

{

    int real,img;

public:

    void add(complexNo c1, complexNo c2){

        cout<<"Added Complex no. is: "<<(c1.real+c2.real)<<"+"<<(c1.img+c2.img)<<"i";

    }

    void setval(int r,int i=2){

        real=r;

        img=i;

    }

};

int main(){

    complexNo c1,c2;

    c1.setval(1);

    c2.setval(3);

    c1.add(c1,c2);

}

**OUTPUT:**



**EXPERIMENT-26**

**AIM:** WAP to overload constructor complex int terms of number and type of arguments. Create a default constructor, constructor with one and two arguments respectively. Write a friend function that should return complex number containing the sum of any two numbers created by constructor.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class complexno{

    float real;

    float imaginary;

public:

    complexno(){

        real=2.0;

        imaginary=2.0;

        cout<<"i am default constructor \n";

    }

    complexno(float a){

        real=imaginary=a;

        cout<<"i am constructor with one arg. \n";

    }

    complexno(float a,float b){

        real=a;

        imaginary=b;

        cout<<"i am constructor with two arg.(float) \n";

    }

    complexno(int a, int b){

        real=a;

        imaginary=b;

        cout<<"i am constructor with two arg.(int) \n";

    }

    complexno(complexno &obj){

        real=obj.real;

        imaginary=obj.imaginary;

        cout<<"i am copy constructor\n";

    }

    friend complexno sum(complexno o1, complexno o2);

    void putdata(){

        cout<<"\n\nThe added complex no. is: "<<real<<"+"<<imaginary<<"i";

    }

};

complexno sum(complexno o1, complexno o2){

        complexno c;

        c.real= o1.real + o2.real;

        c.imaginary= o1.imaginary + o2.imaginary;

        return c;

}

int main()

{

    complexno c1(3),c2(3,1),c3;

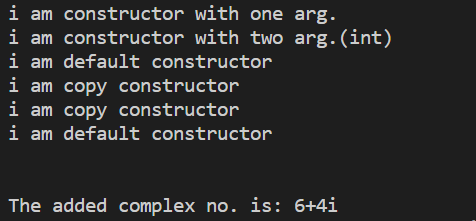
    c3=sum(c1,c2);

    c3.putdata();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-27**

**AIM:** Design a class string with attributes ‘name’ and ‘length’ which are private members and

method display() in public section. Create a dynamic constructor reading the string in

dynamically allocated memory. Design a destructor for the class which should release the

dynamically allocated memory.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <cstring>

using namespace std;

class String

{

    char \*name;

    int length;

public:

    String(const char \*s){

        int length= strlen(s);      //calc length of given string

        name= new char[length+1];  //dynamic memory

        strcpy(name,s);             //copy name to pointer

    }

    void display(){

        cout<<name;

    }

    ~String(){

        free(name);

    }

};

int main()

{

    const char\* name = "Sachin";

    String s1(name);

    s1.display();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-28**

**AIM:** An educational institution wishes to maintain a database of its employees. The database is

divided into a number of classes whose hierarchical relationship are shown in the fig. The

figure also show the minimum information required for each class. Specify all the classes and

define functions to create the database and retrieve individual information as and when

required.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <cstring>

#include <cstdio>

using namespace std;

class staff

{

protected:

    int code;

    char name[50];

};

class teacher : public staff

{

protected:

    char subject[20];

    char publication[50];

public:

    void read(){

        cin>>code;

        cin.getline (name,sizeof(name));

        cin.getline (subject,sizeof(subject));

        cin.getline (publication,sizeof(publication),'$');

    }

    void display(){

        cout<<"\n"<<code<<" "<<name<<" "<<subject<<" "<<publication<<"\n";

    }

};

class typist : public staff

{

protected:

    int speed;

};

class regular : public typist

{

    float salary;

public:

    void read(){

        cin>>code;

        gets(name);

        cin>>speed>>salary;

    }

    void display(){

        cout<<"\n"<<code<<" "<<name<<" "<<speed<<" "<<salary<<"\n";

    }

};

class casual : public typist

{

    float wages;

public:

    void read(){

        cin>>code;

        gets(name);

        cin>>speed>>wages;

    }

    void display(){

        cout<<"\n"<<code<<" "<<name<<" "<<speed<<" "<<wages<<"\n";

    }

};

class officer : public staff

{

protected:

    char grade;

public:

    void read(){

        cin>>code;

        gets(name);

        cin>>grade;

    }

    void display(){

        cout<<"\n"<<code<<" "<<name<<" "<<grade<<"\n";

    }

};

int main()

{

    int ch;

    teacher t1; regular r1; casual c1; officer o1;

    cout<<"Choose a option:\n1- Teacher\n2- Regular Typist\n3- Casual Typist\n4- Officer\n\n";

    cin>>ch;

    switch (ch)

    {

    case 1:cout<<"\n";

        t1.read();

        t1.display();

        break;

    case 2:cout<<"\n";

        r1.read();

        r1.display();

        break;

    case 3:cout<<"\n";

        c1.read();

        c1.display();

    case 4:cout<<"\n";

        o1.read();

        o1.display();

    default:

        cout<<"Invalid Choice";

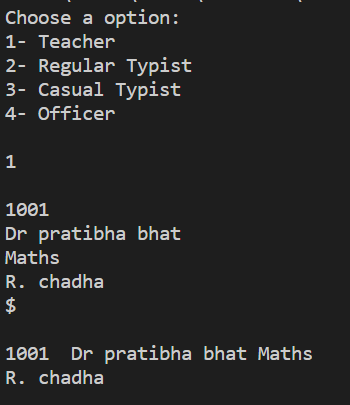
        break;

    }

    return 0;

}

**OUTPUT:**



**EXPERIMENT-29**

**AIM:** WAP to implement the following hybrid inheritance

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class student

{

    int rollno;

public:

    void readr(){

        cin>>rollno;

    }

    int get\_rollno(){

        return rollno;

    }

};

class test: public student

{

    float sub1,sub2;

public:

    void reads(){

        cin>>sub1>>sub2;

    }

    float get\_sub1(){

        return sub1;

    }

    float get\_sub2(){

        return sub2;

    }

};

class sports

{

    float sports\_score;

public:

    void readss(){

        cin>>sports\_score;

    }

    float get\_sports\_score(){

        return sports\_score;

    }

};

class result: public test, public sports

{

    float tot;

public:

    void read(){

        readr();

        reads();

        readss();

    }

    void display(){

        tot = get\_sub1()+get\_sub2()+get\_sports\_score();

        cout<<"\nTotal: "<<tot;

    }

};

int main()

{

    result r1;

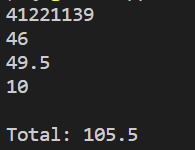
    r1.read();

    r1.display();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-30**

**AIM:** WAP to implement aggregation ‘Course has instructor’ and ‘Course has textbook’.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <string>

using namespace std;

class instructor

{

    string name;

    long int phone;

public:

    void setdata(string n, long int ph){

        name=n;

        phone=ph;

    }

    void putdata(){

        cout<<"Name: "<<name<<"\nPhone: "<<phone;

    }

};

class textbook

{

    string title;

    int book\_id;

public:

    void setdata(string n, int ph){

        title=n;

        book\_id=ph;

    }

    void putdata(){

        cout<<"Title: "<<title<<"\nBook id: "<<book\_id;

    }

};

class course

{

    textbook t;

    instructor i;

    string course\_name;

public:

    void setdata(string cn, string title, int book\_id, string name, long int phone){

    course\_name=cn;

    t.setdata(title, book\_id);

    i.setdata(name,phone);

    }

    void putdata(){

    cout<<"Course Name: "<<course\_name;

    cout<<"\n\nTextBook:- \n";

    t.putdata();

    cout<<"\n\nInstructor:- \n";

    i.putdata();

    }

};

int main(){

    course c1;

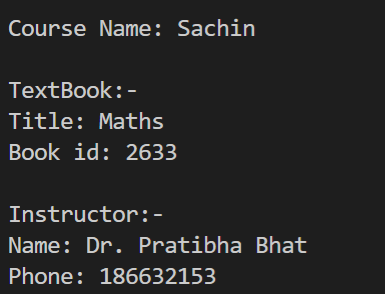
    c1.setdata("Sachin", "Maths", 2633, "Dr. Pratibha Bhat", 8776566745);

    c1.putdata();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-31**

**AIM:** WAP to implement function overriding.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class Base

{

public:

    void print(){

        cout<<"Base\n";

    }

};

class Derived : public Base

{

public:

    void print(){

        cout<<"Derived\n";

    }

};

int main(){

    Derived d;

    d.print();

    d.Base::print();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-32**

**AIM:** WAP to initialise base class constructor through derive class constructor in case of multiple inheritance.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class alpha

{

    int a;

public:

    alpha(int x)

    {

        a=x;

    }

    void show\_a(){

        cout<<"\na: "<<a;

    }

};

class beta

{

    float b;

public:

    beta(float y)

    {

        b=y;

    }

    void show\_b(){

        cout<<"\nb: "<<b;

    }

};

class gamma: public alpha, public beta

{

    int m;

    int n;

public:

    gamma(int a, float b, int c, int d): alpha(a), beta(b)

    {

        m=c;

        n=d;

    }

    void show\_mn(){

        cout<<"\nm: "<<m<<"\nn: "<<n;

    }

};

int main(){

    gamma g1(1,2.0,3,4);

    g1.show\_a();

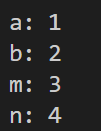
    g1.show\_b();

    g1.show\_mn();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-33**

**AIM:** WAP to display the student result in the following scenario using concept of virtual base class.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class student

{

protected:

    int rollno;

};

class test: virtual public student

{

protected:

    int sub1, sub2;

};

class sports: virtual public student

{

protected:

    int sports\_score;

};

class result: public test, public sports

{

    int tot;

public:

    void getdata(){

        cout<<"Enter details:- Rollno Sub1\_marks Sub2\_marks Sports\_score \n";

        cin>>rollno>>sub1>>sub2>>sports\_score;

    }

    void calTot(){

        tot=sub1+sub2+sports\_score;

    }

    void putdata(){

        cout<<"Total: "<<tot;

    }

};

int main(){

    result r1;

    r1.getdata();

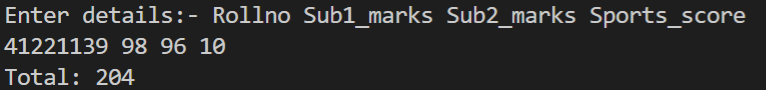
    r1.calTot();

    r1.putdata();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-34**

**AIM:** Write a program to overload binary operator ‘+’ to concatenate two character arrays together.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <string.h>

using namespace std;

// Class to implement operator overloading

// function for concatenating the strings

class AddString {

public:

    // Classes object of string

    char s1[25], s2[25];

    // Parameterized Constructor

    AddString(char str1[], char str2[])

    {

        // Initialize the string to class object

        strcpy(this->s1, str1);

        strcpy(this->s2, str2);

    }

    // Overload Operator+ to concat the string

    void operator+()

    {

        cout << "\nConcatenation: " << strcat(s1, s2);

    }

};

// Driver Code

int main()

{

    // Declaring two strings

    char str1[] = "Ms. Komal";

    char str2[] = " Dhingra";

    // Declaring and initializing the class

    // with above two strings

    AddString a1(str1, str2);

    // Call operator function

    +a1;

    return 0;

}

**OUTPUT:**



**EXPERIMENT-35**

**AIM:** Write a program to overload ‘>’ operator to determine if length of one character array is

greater than other.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <cstring>

using namespace std;

class charArr

{

    char arr[50];

public:

    void getdata(){

        cin.getline(arr,50);

    }

    void operator >(charArr b){

        if(strlen(arr)>strlen(b.arr))

        {

            cout<<arr<<" is greater!";

        }

        else

        {

            cout<<b.arr<<" is greater!";

        }

    }

};

int main(){

    charArr x, y;

    x.getdata();

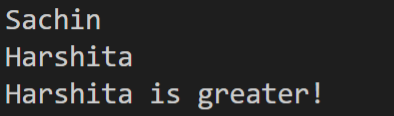
    y.getdata();

    x.operator >(y);

    return 0;

}

**OUTPUT:**



**EXPERIMENT-36**

**AIM:** Create a base class shape with two double type values to compute area of figures. Derive two classes Triangle and Rectangle from base class shape. Add two member functions in base class getdata() - to initialise base class members and displayarea() - to compute and display area of figures. The displayarea() is virtual function in base class. Redefine this function in derived class. Accept dimensions from user and display the area using run time polymorphism.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class shape{

protected:

    double dimen1;

    double dimen2;

public:

    void getdata(){

        cout<<"\nEnter two dimension : ";

        cin>>dimen1>>dimen2;

    }

    virtual void display\_area(){   //Significance in derived class

        cout<<"Inside Shape";

    }

};

class triangle: public shape

{

public:

    void display\_area(){

        cout<<"Area of triangle: "<<0.5\*dimen1\*dimen2;

    }

};

class rectangle: public shape

{

public:

    void display\_area(){

        cout<<"Area of rectangle: "<<dimen1\*dimen2;

    }

};

int main(){

    shape\* s;

    triangle t1;

    rectangle r1;

    s=&t1;

    s->getdata();

    s->display\_area();

    s=&r1;

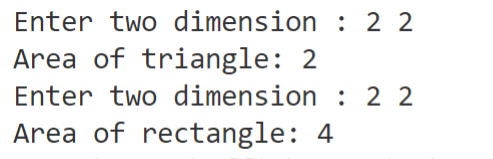
    s->getdata();

    s->display\_area();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-37**

**AIM:** WAP to overload ‘!’ operator to reverse the case of a string.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class String

{

    char arr[50];

public:

    void getdata(){

        cin.getline(arr,50);

    }

    void operator !(){

    for(int i=0;arr[i]!='\0';i++)

    {

        if(arr[i]>=65 && arr[i]<=90)

        {

            arr[i]+=32;

        }

        else if(arr[i]>=97 && arr[i]<=122)

        {

            arr[i]-=32;

        }

    }

    cout<<"Reverse string: "<<arr;

    }

};

int main()

{

    String s1;

    s1.getdata();

    s1.operator !();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-38**

**AIM:** WAP to overload ‘!’ operator to reverse the case of a string.Bubble Sort)

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

// A function to implement bubble sort

template<class T3>

void bubbleSort(T3 arr[], int n)

{

    int i, j;

    for (i = 0; i < n - 1; i++)

        // Last i elements are already

        // in place

        for (j = 0; j < n - i - 1; j++)

            if (arr[j] > arr[j + 1])

                swap(arr[j], arr[j + 1]);

}

// Function to print an array

template<class T1>

void printArray(T1 arr[], int size)

{

    int i;

    for (i = 0; i < size; i++)

        cout << arr[i] << " ";

    cout << endl;

}

// Driver code

int main()

{

    int arr[] = { 5, 1, 4, 2, 8};

    int N = sizeof(arr) / sizeof(arr[0]);

    cout << "Unsorted array: \n";

    printArray(arr, N);

    bubbleSort(arr, N);

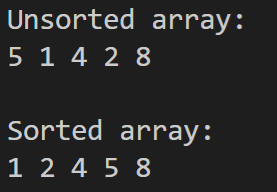
    cout << "\nSorted array: \n";

    printArray(arr, N);

    return 0;

}

**OUTPUT:**



**EXPERIMENT-39**

**AIM:** WAP to overload template functions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

template<class T>

void display(T a){

    cout<<"\n\nGeneric 1"<<endl;

    cout<<a;

}

template<class T1,class T2>

void display(T1 c,T2 d){

    cout<<"\n\nGeneric 2"<<endl;

    cout<<c<<" "<<d;

}

void display(int x){

    cout<<"ordinary function[int]"<<endl;

    cout<<x;

}

template<class T3, class T4, class T5>

void display(T3 x, T4 y, T5 z){

    cout<<"\n\n3 classes"<<endl;

    cout<<x<<" "<<y<<" "<<z;

}

int main()

{

    display(3);

    display(2.5);

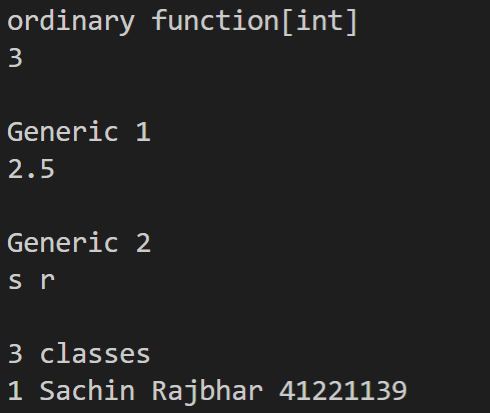
    display('s','r');

    display(1,"Sachin Rajbhar",41221139);

    return 0;

}

**OUTPUT:**



**EXPERIMENT-40**

**AIM:** WAP using class templates to obtain sum of two complex numbers. The class template should have two placeholders and a constructor to initialize the members.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

template<class T1, class T2>

class complex{

    T1 real;

    T2 img;

public:

    complex(T1 a, T2 b){

        real=a;

        img=b;

    }

    void display(){

        cout<<"\n"<<real<<"+"<<img<<"i";

    }

};

int main()

{

    complex <int,float>c1(2,3.0f);

    c1.display();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-41**

**AIM:** Create a class student with members as rollno and name, public member functions to read and display members.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <fstream>

#define N 3  //no. of students

using namespace std;

class student{

    char name [20];

    int rollno;

public:

    void getdata(){

        cin>>rollno;

        cin>>name;

    }

    void putdata(){

        cout<<rollno<<"\n"<<name<<"\n";

    }

};

int main(){

    student stu[N],temp;

    fstream file;

    file.open("student.dat",ios::in|ios::out|ios::trunc);

    if(file.fail())

    {

        cout<<"File cannot be created\n";

    }

    cout<<"Enter rollno, name of "<<N<<"students:\n";

    //writing obj to file

    for (int i=0;i<N;i++){

        stu[i].getdata();

        file.write((char\*)&stu[i],sizeof(stu[i]));

    }

    cout<<"File Content:\n";

    //reading obj from file

    file.seekg(0);

    for (int i=0;i<N;i++){

        file.read((char\*)&temp,sizeof(temp));

        temp.putdata();

    }

    //Modify

    int n;

    cout<<"\nEnter position to modify: ";

    cin>>n;

    file.seekp((n-1)\*sizeof(temp),ios::beg);

    temp.getdata();

    file.write((char\*)&temp,sizeof(temp));

    cout<<"File Content:\n";

    file.seekg(0);

    cout<<"\n";

    for (int i=0;i<N;i++){

        file.read((char\*)&temp,sizeof(temp));

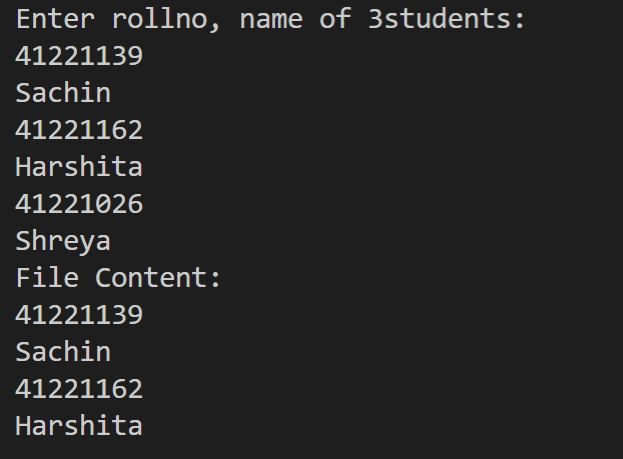
        temp.putdata();

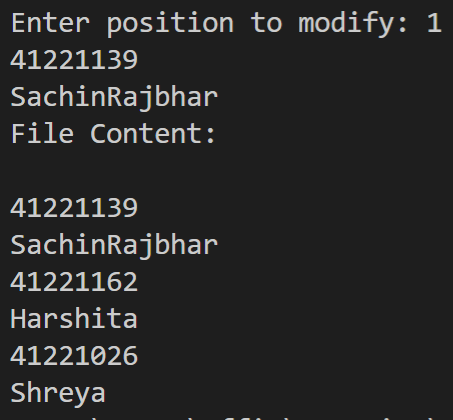
    }

    return 0;

}

**OUTPUT:**





**EXPERIMENT-42**

**AIM:** Write a program with a

a) Function to read two integer variables and an operator of type char.

b) Function to calculate the result of operator on operand.

c) try block to throw exception when wrong type of data for operand is entered

d) try block to throw exception when operator is other than ‘+’, ‘-’, ‘\*’, ‘/’.

e) catch block to handle the exceptions.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

using namespace std;

class Compute

{

    int a,b;

    char x;

public:

    void read(){

        cin>>a>>b;

    }

    void calculate(){

        try{

            if(cin.fail())

            throw "Invalid Input";

            cin>>x;

            if((x!='+')&&(x!='-')&&(x!='/')&&(x!='\*'))

                throw x;

            switch(x)

            {

                case '+': cout<<a+b; break;

                case '-': cout<<a-b; break;

                case '/': cout<<a/b; break;

                case '\*': cout<<a\*b; break;

            }

        }

        catch(char x)

        {

            cout<<"Invalid Operation\n";

        }

        catch(const char\*p)

        {

            cout<<p;

        }

    }

};

int main(){

    Compute c1;

    c1.read();

    c1.calculate();

    return 0;

}

**OUTPUT:**



**EXPERIMENT-43**

**AIM:** Write a program to perform following password validation(s):

a) Length of password should be minimum 6 characters.

b) It should contain atleast one digit.

Throw exceptions and handle them if validations are not met. In case of successful validation,

print “Password Validation Successful”.

**SOFTWARE USED:** VS code

**PROGRAM:**

#include <iostream>

#include <cstring>

using namespace std;

class Validate

{

    char pass[50];

public:

    void getdata(){

        cin.getline(pass,50);

    }

    void validation(){

        bool flag=false;

        try{

            if(strlen(pass)<6)

                throw "Invalid Length";

            for(int i=0; i<strlen(pass); ++i)

                if(isdigit(pass[i])){

                    flag=true;

                    break;

                }

            if(!flag)

                throw "Digit not present";

            else

                cout<<"Password Valid";

        }

        catch(const char\*p){

            cout<<p;

        }

    }

};

int main(){

    Validate v1;

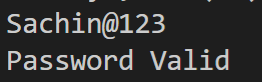
    v1.getdata();

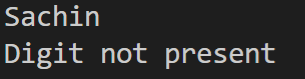
    v1.validation();

    return 0;

}

**OUTPUT:**





**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**

**EXPERIMENT-**

**AIM:**

**SOFTWARE USED:** VS code

**PROGRAM:**

**OUTPUT:**