**Q.1)** Write a C++ program to create a class Worker with data members as Worker\_Name, No\_of\_Hours\_worked, Pay\_Rate. Write necessary member functions to calculate and display the salary of worker. (Use default value for Pay\_Rate)

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class Worker

{

char wname[20];

int n;

float pr;

public:

void accept(char lw[20], int ln, float lpr = 10)

{

strcpy(wname, lw);

n = ln;

pr = lpr;

}

void calculate\_display()

{

cout<<"\nSalary of worker ="<<n \* pr;

}

};

int main()

{

clrscr();

char wname[20];

int n;

Worker wobj;

cout<<"\nEnter wname, hours ";

cin>>wname>>n;

wobj.accept(wname, n);

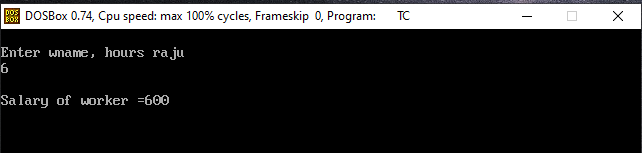
wobj.calculate\_display();

getch();

return 0;

}

**Output :**



**Q.2)** Create class Person which contains data member as Passport\_Id, Person\_name, Nationality, Gender, Date\_of\_Birth, Date\_of\_Issue, Date\_of\_expiry .

Write a c++ program to perform following member functions:

1. Enter details of all persons
2. Display passport details of one person
3. Display passport details of all persons

(Use Function overloading and Array of object).

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Person

{

int pid;

char pname[20], nat[20];

public:

void accept();

void display(int pid);

void display();

};

void Person :: accept()

{

cout<<"\nEnter pid, pname ,nationality";

cin>>pid>>pname>>nat;

}

void Person :: display()

{

cout<<"\n PID = "<<pid;

cout<<"\n PNAME = "<<pname;

cout<<"\n NATIONALITY = "<<nat;

}

void Person :: display(int temp\_pid)

{

if(temp\_pid == pid)

{

cout<<"\n PID = "<<pid;

cout<<"\n PNAME = "<<pname;

cout<<"\n NATIONALITY = "<<nat;

}

}

int main()

{

clrscr();

Person parr[2];//array of object

for(int i = 0; i < 2; i++)

{

cout<<"\n Enter details of person number "<<i + 1;

parr[i].accept();

}

cout<<"\n Details of all the persons are below ...";

for(i = 0; i < 2; i++)

{

cout<<"\n Details of person number "<<i + 1;

parr[i].display();

}

int id;

cout<<"\n Enter person id of one person.";

cin>>id;

for(i = 0; i < 2; i++)

{

parr[i].display(id);

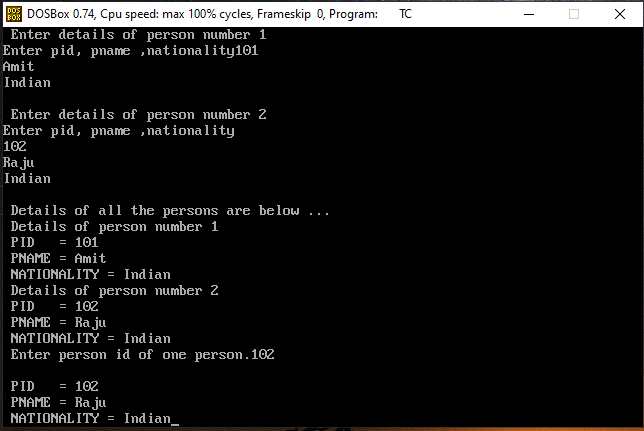
}

getch();

return 0;

}

**Output :**



**Q.3)** Write a C++ program using class which contains two data members of type integer. Create and initialize the object using default constructor, parameterized constructor and parameterized constructor with default value. Write a member function to display maximum from given two numbers for all objects.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Number

{

int no1;

int no2;

public:

Number()

{

no1 = 0;

no2 = 0;

cout<<"\nDefault Const. ";

}

Number(int n)

{

no1 = n;

no2 = n;

cout<<"\n Paramet. Const. :";

}

Number(int n1, int n2 = 20)

{

no1 = n1;

no2 = n2;

cout<<"\n Parame. with default argument.:";

}

void display\_max()

{

if( no1 > no2)

{

cout<<"\nThe number "<<no1<<" is > "<<no2<<endl;

}

else

{

cout<<"\nThe number "<<no2<<" is > "<<no1<<endl;

}

}

};

int main()

{

clrscr();

Number obj1;

obj1.display\_max();

Number obj2(11,22);

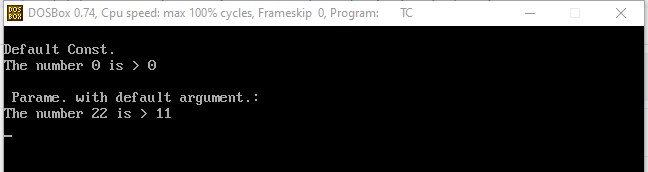
obj2.display\_max();

getch();

return 0;

}

**Output :**



**Q.4)** Create a base class Student(Roll\_No, Name) which derives two classes Academic\_Marks(Mark1, Mark2, Mark3) and Extra\_Activities\_Marks(Marks). Class Result(Total\_Marks, Grade) inherits both Academic\_Marks and Extra\_Activities\_Marks classes. (Use Virtual Base Class)

Write a C++ menu driven program to perform the following functions:

1.Build a master table

2.Calculate Total\_marks and grade

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class student

{

public:

int roll\_no;

char name[20];

student()

{

}

student(int rn,char na[20])

{

roll\_no=rn;

strcpy(name,na);

}

};

class academic\_marks:virtual public student

{

public:

int mark1,mark2,mark3;

academic\_marks()

{

}

academic\_marks(int m1,int m2,int m3)

{

mark1=m1;

mark2=m2;

mark3=m3;

}

};

class extra\_activities\_marks:virtual public student

{

public:

int mark4;

extra\_activities\_marks()

{

}

extra\_activities\_marks(int m4)

{

mark4=m4;

}

};

class result:public academic\_marks,public extra\_activities\_marks

{

int total\_marks;

float perc;

public:

result()

{

}

result(int rn,char na[20],int m1,int m2,int m3,int m4):student(rn,na),academic\_marks(m1,m2,m3),extra\_activities\_marks(m4)

{

}

void cal()

{

total\_marks=mark1+mark2+mark3+mark4;

perc=total\_marks/4;

cout<<"\ntotal marks:"<<total\_marks;

cout<<"\npercentage:"<<perc;

}

};

int main()

{

clrscr();

int rn, m1, m2, m3, m4;

rn = 10;

char na[20];

cout<<"\nenter name:";

cin>>na;

cout<<"enter marks:";

cin>>m1>>m2>>m3>>m4;

result r(rn,na,m1,m2,m3,m4);

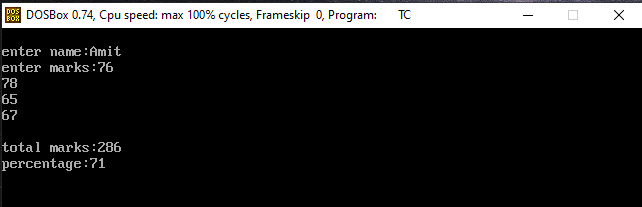
r.cal();

getch();

return 0;

}

**Output :**



**Q.5)** Write a C++ program to create a class Part which contains data members as Part\_Id, Part\_Name, Part\_Price. Create and Initialize all values of Part object by using parameterized constructor and copy constructor. Display the values of Part object. (Part\_price should be right justified with a precision of two digits)

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

#include<iomanip.h>

class part

{

int part\_id;

char part\_name[10];

float part\_price;

public:

part()

{

}

part(int pi,char pn[10],float pp)

{

part\_id=pi;

strcpy(part\_name,pn);

part\_price=pp;

}

part(part &p1)

{

part\_id=p1.part\_id;

strcpy(part\_name,p1.part\_name);

part\_price=p1.part\_price;

}

void putdata()

{

cout<<"\npartid:"<<part\_id;

cout<<"\npartname:"<<part\_name;

cout<<"\npartprice:"<<setw(10)<<setprecision(2)<<part\_price;

}

};

int main()

{

clrscr();

int n,pi;

char pn[10];

float pp;

part p1;

cout<<"\nenter partid:";

cin>>pi;

cout<<"\nenter partname:";

cin>>pn;

cout<<"\nenter partprice:";

cin>>pp;

cout<<"\nobject for parameterized constructor:";

p1=part(pi,pn,pp);

p1.putdata();

cout<<"\nobject for copy constructor:";

part p2(p1);

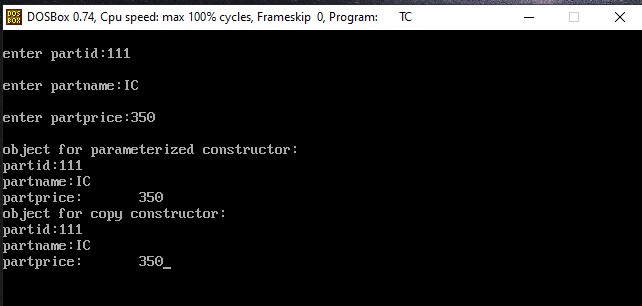
p2.putdata();

getch();

return 0;

}

**Output :**



**Q.6)** Write a C++ program to create a class Item with data members Item\_Code, Item\_Name, Item\_Price. Write member functions to accept and display Item information also display number of objects created for a class. (Use Static data member and Static member function)

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Item

{

int icode,iprice;

char name[20];

static int cnt;

public:

Item()

{

cnt++;

}

void accept();

void display();

static void display\_cnt()

{

cout<<"\nNumber of objects created=:"<<cnt;

}

};

void Item::accept()

{

cout<<"\nEnter ItemCode:";

cin>>icode;

cout<<"\nEnter ItemPrice andItemName:";

cin>>iprice>>name;

}

void Item::display()

{

cout<<"\nItemCode:"<<icode<<"\nItemName:"<<name<<"\nItemPrice:"<<iprice;

}

int Item::cnt;

void main()

{

clrscr();

Item i1,i2;

i1.accept();

i2.accept();

i1.display();

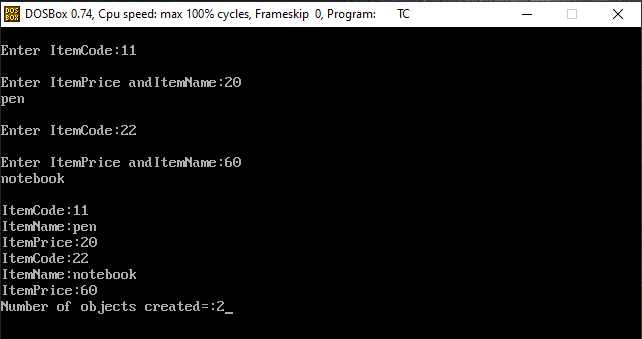
i2.display();

Item::display\_cnt();

getch();

}

**Output :**



**Q.7)** Create a Base class Train containing protected data members as Train\_no, Train\_Name. Derive a class Route (Route\_id, Source, Destination) from Train class. Also derive a class Reservation(Number\_Of\_Seats, Train\_Class, Fare, Travel\_Date) from Route. Write a C++ program to perform following necessary functions :

1. Enter details of ‘n’ reservations
2. Display details of all reservations

Display reservation details of a specified Train class

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class Train

{

protected:

char tname[20];

public:

void accept\_td()

{

cout<<"\nEnter train name : ";

cin>>tname;

}

};

class Route : public Train

{

int rid;

char source[20];

char dest[20];

public:

void acceptr()

{

accept\_td();

cout<<"\n enter route id, source,destination :";

cin>>rid>>source>>dest;

}

void displayr()

{

cout<<"\nroute id:"<<rid<<"\n source:"<<source<<"\n destination:"<<dest;

}

};

class Reservation: public Route

{

int ns,f;

char tclass[20];

public:

void accept\_resd()

{

cout<<"\n enter number of seats, fare:";

cin>>ns>>f;

cout<<"\nEnter train class :";

cin>>tclass;

}

void display\_trainclass\_wise(char temp\_class[20])

{

if(strcmp(temp\_class,tclass) == 0)

{

display\_res();

}

}

void display\_res()

{

cout<<"\n number of seats:"<<ns;

cout<<"\n fare:"<<f;

cout<<"\nT class = " <<tclass;

}

};

int main()

{

clrscr();

Route robj;

robj.acceptr();

Reservation rarr[2];

char ttclass[20];

for(int i = 0;i < 2;i++)

rarr[i].accept\_resd();

cout<<"\n\nEnter specified class ....";

cin>>ttclass;

for(i = 0; i<2;i++)

rarr[i].display\_trainclass\_wise(ttclass);

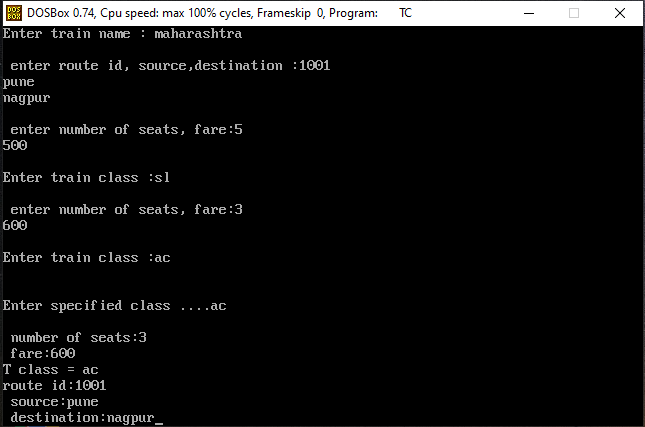
robj.displayr();

getch();

return 0;

}

**Output :**



**Q.8)** Create a class Date containing members as:

* dd
* mm
* yyyy

Write a C++ program for overloading operators >> and << to accept and display a Date also write a member function to validate a date.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class date

{

int dd,mm,yyyy;

public:

void validation();

friend istream &operator>>(istream &in, date &obj);

friend ostream &operator<<(ostream &out, date &obj);

};

istream &operator>>(istream &in, date &obj)

{

cout<<"\nenter date";

in>>obj.dd;

cout<<"\nenter month";

in>>obj.mm;

cout<<"\nenter year";

in>>obj.yyyy;

return in;

}

ostream &operator<<(ostream &out, date &obj)

{

out<<"\ndate is:"<<obj.dd<<"/"<<obj.mm<<"/"<<obj.yyyy;

return out;

}

void date::validation()

{

int d;

switch(mm)

{

case 1:d=31;

break;

case 2:if((yyyy%4==0 && yyyy%100!=0)||(yyyy%400==0))

d=29;

else

d=28;

break;

case 3:d=31;

break;

case 4:d=30;

break;

case 5:d=31;

break;

case 6:d=30;

break;

case 7:d=31;

break;

case 8:d=31;

break;

case 9:d=30;

break;

case 10:d=31;

break;

case 11:d=30;

break;

case 12:d=31;

break;

}

if(d!=dd || mm>12)

cout<<"\nInvalid date";

else

cout<<"\nValid date";

}

int main()

{

clrscr();

date a;

cin>>a ;

cout<<a ;

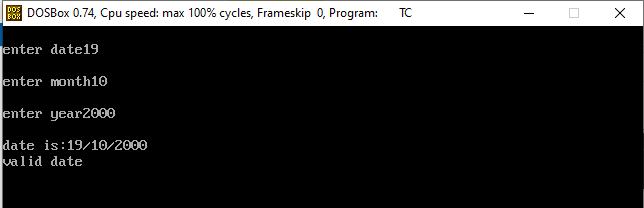
a.validation();

getch();

return 0;

}

**Output :**



**Q.9)** Create two base classes Learning\_Info( Roll\_No, Stud\_Name, Class, Percentage) and Earning\_Info(No\_of\_hours\_worked, Charges\_per\_hour). Derive a class Earn\_Learn\_info from above two classes. Write necessary member functions to accept and display Student information. Calculate total money earned by the student. (Use constructor in derived class)

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class learning\_info

{

public:

int rno;

char name[20],cls[10];

float per;

learning\_info(int roll,char sname[],char cl[],float perc)

{

rno=roll;

strcpy(name,sname);

strcpy(cls,cl);

per=perc;

}

void display()

{

cout<<"\n Roll no = "<<rno<<"\n Name="<<name<<"\n class = "<<cls;

cout<<"\n Percentage = "<<per;

}

};

class earning\_info

{

public:

int hrs,charges;

earning\_info(int hr,int chrgs)

{

hrs=hr;

charges=chrgs;

}

void display()

{

cout<<"\n Hours = "<<hrs;

cout<<"\t charges = "<<charges;

}

};

class earn\_learn\_info:public learning\_info,public earning\_info

{

public:

earn\_learn\_info(int roll,char sname[],char cl[],float perc,int hr,int chrgs):learning\_info(roll,sname,cl,perc),earning\_info(hr,chrgs)

{

}

void display()

{

learning\_info::display();

earning\_info::display();

}

void calculate()

{

int total;

total=charges\*hrs;

cout<<"\n Money earned by student:= "<<total;

}

};

void main()

{

clrscr();

int rno,ch,hr;

float p;

char nm[10],c[10];

earn\_learn\_info ob1(1,"Amit","sy",77,5,50);

ob1.display();

ob1.calculate();

cout<<"\n Enter rno : ";

cin>>rno;

cout<<"\n enter name : ";

cin>>nm;

cout<<"\n Enter class : ";

cin>>c;

cout<<"\n Enter percentage : ";

cin>>p;

cout<<"\n Enter working hr : ";

cin>>hr;

cout<<"\n Enter charges : ";

cin>>ch;

earn\_learn\_info ob2(rno,nm,c,p,hr,ch);

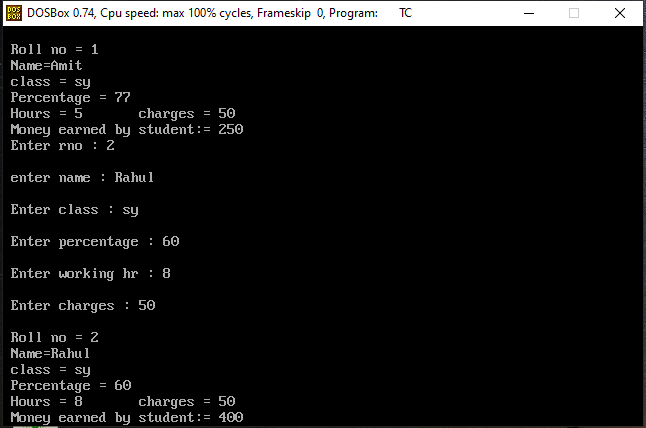
ob2.display();

ob2.calculate();

getch();

}

**Output :**



**Q.10)** Write a C++ program to find area and volume of cylinder using Inline function.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

define pi=3.14

inline void area(float r, float h)

{

float a

a = 2 \* pi \* r \* r + 2 \* pi \* r \* h;

cout<<"\n area of cylinder :"<<a<<endl;

}

inline void volume(float r, float h)

{

float v

v = pi \* r \* r \* h;

cout<<"\n volume of cylinder :"<<v<<endl;

}

int main()

{

clrscr();

float r,h;

cout<<"enter number r,h";

cin>>r>>h;

area(r,h);

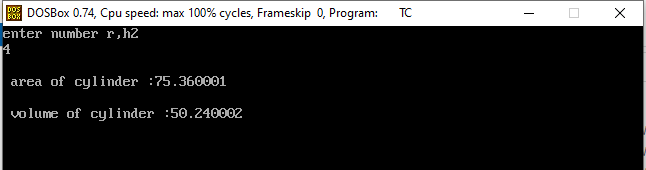
volume(r,h);

getch();

return 0;

}

**Output :**



**Q.11)** Create a class College containing data members as College\_Id, College\_Name,

Establishment\_year, University\_Name. Write a C++ program with following member functions:

1. To accept ‘n’ College details
2. To display College details of a specified University
3. To display College details according to a specified establishment year

(Use Array of Object and Function overloading)

**Ans :**

**Code :**

#include<conio.h>

#include<iostream.h>

#include<string.h>

class college

{

int id,n,yr;

char name[10],uname[10];

public:

void accept()

{

cout<<"\n Enter id : ";

cin>>id;

cout<<"\n Enter name : ";

cin>>name;

cout<<"\n Enter univercity name : ";

cin>>uname;

cout<<"\n Enter year : ";

cin>>yr;

}

void display()

{

cout<<"\n id = "<<id<<"\n name = "<<name<<"\n Univercity name = "<<uname<<"\n year = "<<yr<<endl;

}

int display(char unin[])

{

if(strcmp(uname,unin)==0)

{

display();

return 1;

}

else return 0;

}

int display(int year)

{

if(yr==year)

{

display();

return 1;

}

else return 0;

}

};

void main()

{

int n,yr,a;

college ob[10];

char uni[10];

clrscr();

cout<<"enter no of college";

cin>>n;

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

{

ob[i].accept();

}

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

{

ob[i].display();

}

cout<<"\n Enter uname to search : "<<endl;

cin>>uni;

int cnt=0;

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

{

a=ob[i].display(uni);

if(a==1)

cnt++;

}

if(cnt==0)

cout<<"\n Univercity name is NOT found "<<endl;

cout<<"\n Enter establishment year to search : "<<endl;

cin>>yr;

cnt=0;

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

{

a=ob[i].display(yr);

if(a==1)

cnt++;

}

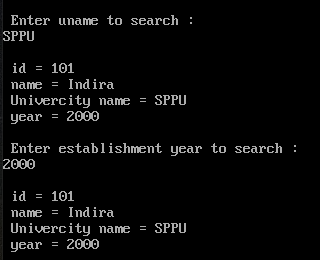
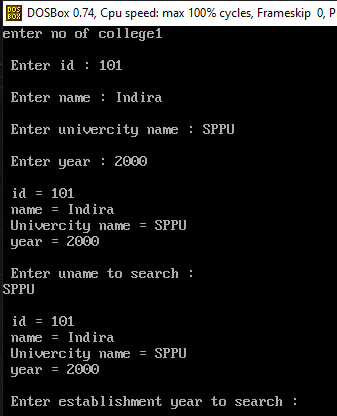
if(cnt==0)

cout<<"\n Year NOT found ";

getch();

}

**Output :**



**Q.12)** Create a class MyString which contains a character pointer (using new operator). Write a

C++ program to overload following operators:

> to compare length of two strings

!= to check equality of two strings

+ to concatenate two strings

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<string.h>

class mystring

{

char \*name;

public:

mystring()

{

name = NULL;

}

mystring(char \*n)

{

int c=strlen(n);

name=new char[c+1];

strcpy(name, n);

}

void putdata()

{

cout<<"\nstring:"<<name;

}

void operator>(mystring m2)

{

if(strlen(name) > strlen(m2.name))

cout<<"\nLength of string 1 > string 2";

else if(strlen(name) < strlen(m2.name))

cout<<"\nLength of string 2 > string 1";

else

cout<<"\nBoth the strings are equal";

}

void operator!=(mystring m2)

{

if(strcmp(name,m2.name)!=0)

cout<<"\nstrings are not equal";

else

cout<<"\nstrings are equal";

}

void operator+(mystring m2)

{

cout<<strcat(name,m2.name);

}

};

int main()

{

clrscr();

char n1[20],n2[20];

cout<<"\nenter 1 string: ";

cin>>n1;

cout<<"\nenter 2 string: ";

cin>>n2;

mystring m1(n1);

mystring m2(n2);

cout<<"\n\ncomparision: ";

m1>m2; //m1.oprator>(m2)

cout<<"\n\nequality: ";

m1!=m2;

cout<<"\n\nconcat: ";

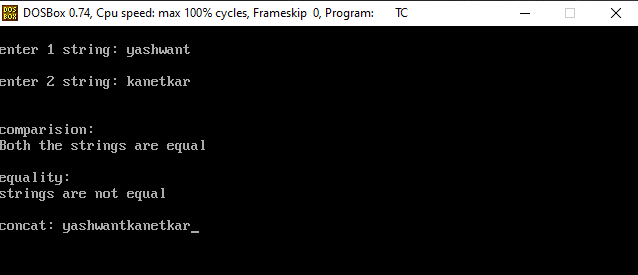
m1+m2;

getch();

return 0;

}

**Output :**



**Q.13)** Write a C++ program to calculate area of cone, sphere and circle by using function overloading.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#define PI 3.14

void area()//circle area

{

float r;

cout<<"\n Enter radius ..";

cin>>r;

cout<<"\n Circle Area = "<< PI \* r \* r;

}

void area(float r)// sphere area

{

cout<<"\nSphere area = "<< 4 \* PI \* r \* r;

}

void area(float r, float l)

{

cout<<"\n Cone area = "<< (PI \* r \* r) + (PI \* r \* l);

}

int main()

{

clrscr();

area(); //circle area will call

area(2.3);// sphere

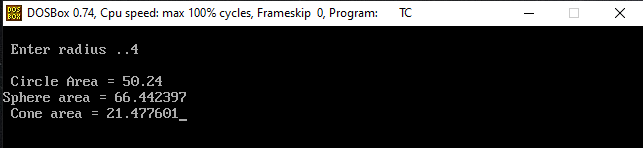
area(1.2,4.5);//cone

getch();

return 0;

}

**Output :**



**Q.14)** Create a base class Shape. Derive three different classes Circle, Rectangle and Triangle from Shape class. Write a C++ program to calculate area of Circle, Rectangle and Triangle. (Use pure virtual function).

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Shape

{

public:

virtual void accept() = 0;

virtual void cal\_display() = 0;

};

class Circle: public Shape

{

float r;

public:

void accept()

{

cout<<"\nEnter radius : ";

cin>>r;

}

void cal\_display()

{

cout<<"\nArea of Cirlcle : "<<3.14\* r \* r;

}

};

class Rectangle: public Shape

{

float len, br;

public:

void accept()

{

cout<<"\nEnter length & breath : ";

cin>>len>>br;

}

void cal\_display()

{

cout<<"\nArea of Rectangle : "<<len \* br;

}

};

class Triangle: public Shape

{

float base, he;

public:

void accept()

{

cout<<"\nEnter base & heigth : ";

cin>>base>>he;

}

void cal\_display()

{

cout<<"\nArea of Triangle : "<<(0.5) \* base \* he;

}

};

int main()

{

clrscr();

Shape \*sptr;

Circle cobj;

Rectangle robj;

Triangle tobj;

//Circle

sptr = &cobj;

sptr->accept();

sptr->cal\_display();

//Rectangle

sptr = &robj;

sptr->accept();

sptr->cal\_display();

//Triangle

sptr = &tobj;

sptr->accept();

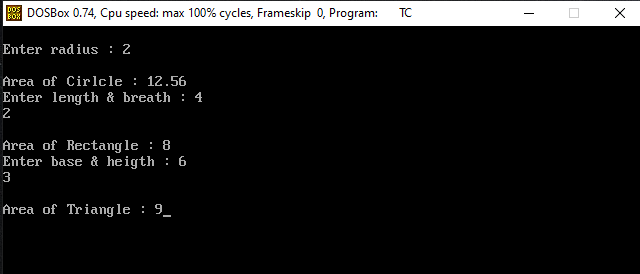
sptr->cal\_display();

getch();

return 0;

}

**Output :**



**Q.15)** Write a C++ program to accept ‘n’ numbers from user through Command Line Argument. Store all positive and negative numbers in two different arrays. Display contents both arrays.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

void main(int argc,char \*argv[])

{

int pos[20],neg[20],n,j=0,k=0;

clrscr();

for(int i=1;i<argc;i++)

{

n=atoi(argv[i]);

if(n>=0)

{

pos[k]=n;

k++;

}

else

{

neg[j]=n;

j++;

}

}

cout<<"\n positive no array is : ";

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

{

cout<<"\t"<<pos[i];

}

cout<<"\n Negative no array is : ";

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

{

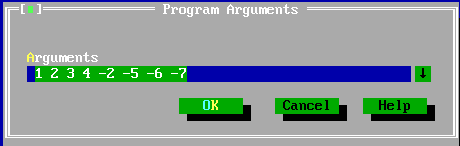
cout<<"\t"<<neg[i];

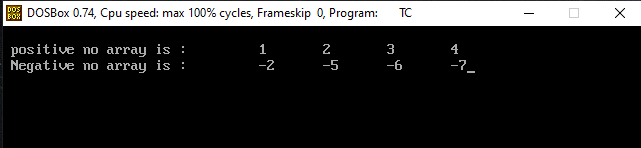
}

getch();

}

**Output :**





**Q.16)** Write a C++ program to create a class Integer. Write necessary member functions to overload the operator unary pre and post increment ‘++’ for an integer number.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Integer

{

int number;

public:

void accept()

{

cout<<"\nEnter number :";

cin>>number;

}

void display()

{

cout<<"\nNumber = "<<number;

}

void operator++() //preincrement

{

++number;

}

void operator++(int k) //postincrement

{

number++;

}

};

int main()

{

clrscr();

Integer iobj;

iobj.accept();

++iobj; // iobj.operator++(); // preincrement

iobj.display();

iobj++; // internaly compiler will convert iobj.operator++(0); // post increment

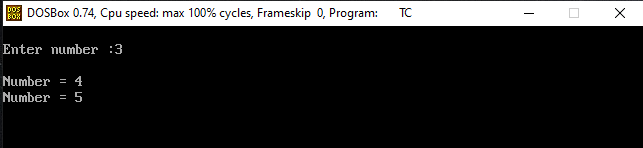
iobj.display();

getch();

return 0;

}

**Output :**



**Q.17)** Write a C++ program to read the contents of a text file. Count and display number of

characters, words and lines from a file. Find the number of occurrences of a given word present in a file.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<stdlib.h>

#include<ctype.h>

#include<stdio.h>

#include<string.h>

int main()

{

clrscr();

ifstream in1;

int cnt1=0,cnt2=0,cnt3=0,cnt4=0,i=0;

char ch,str[20],temp[20],\*ch1;

in1.open("sample.txt");

cout<<"Enter a word to search\n";

scanf("%s",str);

do

{

in1.get(ch);

temp[i++]=ch;

if(ch==' '||ch=='\n'||ch==EOF)

{

cnt1++;

i=0;

memset(temp,0,sizeof(temp));

}

if(!(strcmp(temp,str)))

{

cnt4++;

}

if(isalpha(ch))

{

cnt3++;

}

if(ch=='\n'||ch==EOF)

{

cnt2++;

}

}while(in1);

cout<<"Word Count :"<<cnt1<<"\nLine Count :"<<cnt2<<"\nCharatcter Count"<<cnt3<<"\n\nGiven Word "<<str<<" found "<<cnt4<<" times in the file"<<endl;

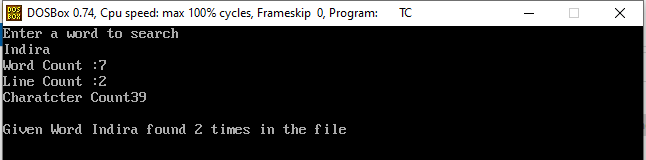
in1.close();

getch();

return 0;

}

**Output :**



**Q.18)** Write a C++ program to read the contents of a “Sample.txt” file. Store all the uppercase characters in ”Upper.txt”, lowercase characters in ”Lower.txt” and digits in “Digit.txt” files. Change the case of each character from “Sample.txt” and store it in “Convert.txt” file.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

#include<fstream.h>

#include<stdlib.h>

#include<ctype.h>

int main()

{

clrscr();

ifstream in1;

char ch,ch2;

ofstream out1,out2,out3,out4;

in1.open("sample.txt");

out1.open("digit.txt");

out2.open("upper.txt");

out3.open("lower.txt");

out4.open("convert.txt");

do

{

in1.get(ch);

if(isalpha(ch))

{

if(isupper(ch))

{

out2<<ch;

ch2=tolower(ch);

out4<<ch2;

}

if(islower(ch))

{

out3<<ch;

ch2=toupper(ch);

out4<<ch2;

}

}

if(isdigit(ch))

{

out1<<ch;

out4<<ch;

}

}while(in1);

cout<<"Program executed"<<endl;

out1.close();

out2.close();

in1.close();

ifstream in2,in3,in4,in5;

in1.open("sample.txt");

in2.open("digit.txt");

in3.open("upper.txt");

cout<<"\nCharacters are: "<<endl;

do

{

in1.get(ch);

cout<<ch;

}while(in1);

cout<<endl<<endl;

in1.close();

cout<<"\nDigts are: "<<endl;

do

{

in2.get(ch);

cout<<ch;

}while(in2);

cout<<endl<<endl;

in2.close();

cout<<"\n upper Characters are: "<<endl;

do

{

in3.get(ch);

cout<<ch;

}while(in3);

cout<<endl<<endl;

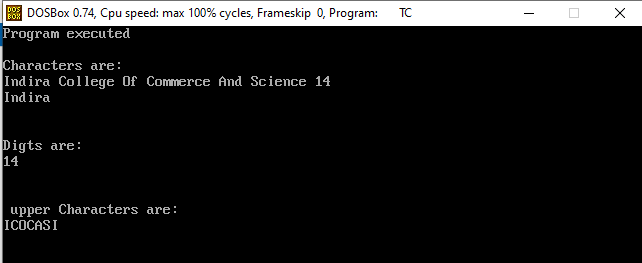
in3.close();

getch();

return 0;

}

**Output :**



**Q.19)** Write a C++ program to calculate maximum of two integer numbers of two different classes using friend function.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

class Integer

{

int a;

public:

Integer()

{

a=0;

}

Integer(int x)

{

a=x;

}

friend int max(Integer,Integer);

};

int max(Integer I1,Integer I2)

{

if(I1.a>I2.a)

{

return I1.a;

}

else

{

return I2.a;

}

}

void main()

{

clrscr();

Integer I1(15);

Integer I2(7);

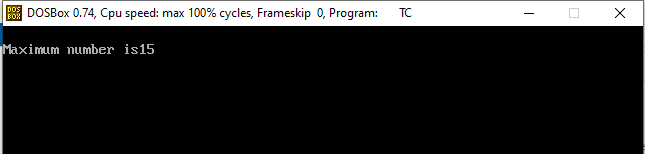
int ans=max(I1,I2);

cout<<"\nMaximum number is"<<ans;

getch();

}

**Output :**



**Q.20)** Write a C++ program to swap two integer values and two float values by using function template.

**Ans :**

**Code :**

#include<iostream.h>

#include<conio.h>

template <class T>

void swap(T &num1, T &num2)

{

T temp;

temp = num1;

num1 = num2;

num2 = temp;

}

int main()

{

clrscr();

int i1 = 10, i2 = 20;

float f1 = 100.11, f2 = 200.22;

cout<<"Before swapping integer values are :";

cout<<"\ni1 = "<<i1<<"\ni2 = "<<i2;

swap(i1, i2);

cout<<"\n\nAfter swapping integer values are :";

cout<<"\ni1 = "<<i1<<"\ni2 = "<<i2;

cout<<"\n\n\nBefore swapping float values are :";

cout<<"\nf1 = "<<f1<<"\nf2 = "<<f2;

swap(f1, f2);

cout<<"\n\nAfter swapping float values are :";

cout<<"\nf1 = "<<f1<<"\nf2 = "<<f2;

getch();

return 0;

}

**Output :**

