**ASSIGNMENT 3**

NAME: SHUBHAM V. TAKANKHAR

CLASS SY-MCA

ROLL NO. 54

GR. NO. 119C0046

Q1: Write a class cString and overload +,<,>,== operator for various string operations.

🡺

**<SOURCE\_CODE>**

#include<iostream>

#include<string.h>

using namespace std;

class cStrings

{

    char str[100];

    int len;

    public:

    void getdata()

    {

        cout<<"\nEnter string:";

        cin>>str;

    }

    void showdata(){

        cout<<str<<endl;

    }

    void operator+(cStrings str1);

    void operator <(cStrings);

    void operator >(cStrings);

    void operator ==(cStrings);

};

void cStrings::operator==(cStrings str1)

{

    if(strcmp(str,str1.str)==0){

        cout<<"strings are equal"<<endl;

    }

    else{

        cout<<"strings are not equat"<<endl;

    }

}

void cStrings::operator+(cStrings str1)

{

    strcat(str,str1.str);

    cout<<"\nString After Concatination:"<<str<<endl;

}

void cStrings::operator>(cStrings str1)

{

    int tstr1=strlen(str1.str);

    int tstr=strlen(str);

    if(tstr1>tstr){

        cout<<str1.str<<" is greater than "<<str;

    }

    else if (tstr==tstr1)

    {

        cout<<"both are equal of size";

    }

    else{

       cout<<str<<" is greater than "<<str1.str;

    }

}

void cStrings::operator<(cStrings str1)

{

    int tstr1=strlen(str1.str);

    int tstr=strlen(str);

    if(tstr1<tstr){

        cout<<str1.str<<" is smaller than"<<str;

    }

    else if (tstr==tstr1)

    {

        cout<<"both are equal of size";

    }

    else{

       cout<<str<<" is smaller than"<<str1.str;

    }

}

int main(){

    int opt,c;

    cStrings a,b;

    while(opt!=5)

    {

        cout<<"\n\t\t---Main Menu---\n\n\t1.Concat\n\t2.Equality";

        cout<<"\n\t3.Greater Length \n\t4.Smaller Length\n\t";

        cout<<"\n\t5.Exit\n\n\t--Enter your choice-->";

        cin>>opt;

        switch(opt)

      {

             case 1:

                      cout<<"\nEnter the 1st string-\n";

                      a.getdata();

                      cout<<"\nEnter the 2nd string-\n";

                      b.getdata();

                    a+b;

                    break;

            case 2:

                 cout<<"\nEnter the 1st string-\n";

                      a.getdata();

                cout<<"\nEnter the 2nd string-\n";

                      b.getdata();

                a==b;

                break;

            case 3:

                 cout<<"\nEnter the 1st string-\n";

                      a.getdata();

                      cout<<"\nEnter the 2nd string-\n";

                      b.getdata();

                    a>b;

                    break;

            case 4:

                 cout<<"\nEnter the 1st string-\n";

                      a.getdata();

                      cout<<"\nEnter the 2nd string-\n";

                      b.getdata();

                      a<b;

                      break;

    case 5: return 0;

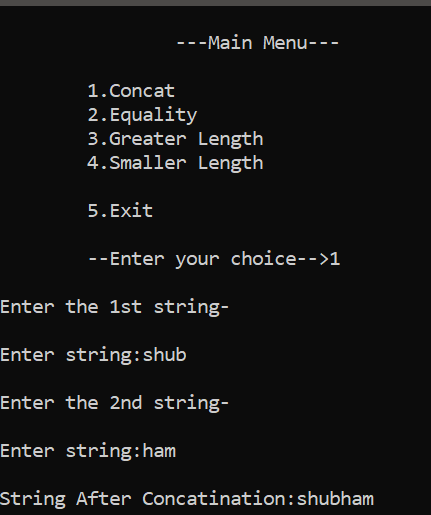
             default: cout<<"Invalid choice..try again\n";

      }

}

}

**<<OUTPUT\_SCREENS>>**



------------------------------------------------------------------------------

Q2. Add additional operators to class complex , overload ++,+=,\*= operators for complex class.

🡺

**<SOURCE\_CODE>**

#include<iostream>

using namespace std;

class Complex{

private:

  int m\_real;

  int m\_imag;

public:

  Complex(int i=0,int r=0){

    m\_real=r;

    m\_imag=i;

  }

  void inputValues(){

    cout<<"\nEnter Real Number: ";

    cin>>m\_real;

    cout<<"Enter Imaginary Number: ";

    cin>>m\_imag;

  }

  Complex operator+(Complex obj){

    Complex temp;

    temp.m\_real=m\_real+obj.m\_real;

    temp.m\_imag=m\_imag+obj.m\_imag;

    return temp;

  }

  void printValues(){

    cout<<"\nOUTPUT: "<<m\_real<<"+"<<m\_imag;

  }

  Complex operator++(){

    m\_real=m\_real+1;

    m\_imag=m\_imag+1;

    cout<<"\nReal no:"<<m\_real<<" Imaginary no:"<<m\_imag<<" after increment.";

  }

  Complex operator+=(Complex obj){

    m\_real=m\_real+obj.m\_real;

    m\_imag=m\_imag+obj.m\_imag;

    cout<<"\nReal no:"<<m\_real<<" Imaginary no:"<<m\_imag<<" after adding.";

  }

  Complex operator\*=(Complex obj){

    Complex temp;

    m\_real=m\_real\*obj.m\_real;

    m\_imag=m\_imag\*obj.m\_imag;

    cout<<"\nReal no:"<<m\_real<<" Imaginary no:"<<m\_imag<<" after multiplication ";

  }

};

int main(){

  while(1){

  Complex a,b,c;

  a.inputValues();

  b.inputValues();

  c=a+b;

  c.printValues();

  ++a;

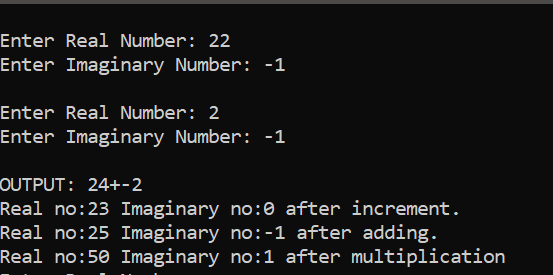
  a+=b;

  a\*=b;

  }

}

**<<OUTPUT\_SCREENS>>**

****