Assignment:1

Problem Statement: Stack operation like push , pop and display using c++.

Source Code:

```
#include <iostream>
//#include <stack>
using namespace std;

int stack[100],n=100,top=-1;
class stack_operation {
int top;
public:
void push(void);
void pop(void);
void show(void);
};
void stack_operation ::push(void)
{
int x;
    if (top >=n - 1)
    {
        cout<<"\nOverflow!!";
    }
    else
    {
        cout<<"\nEnter the element to be added onto the stack: ";
        cin>>x;
        top = top + 1;
        stack[top] = x;
    }
}
void stack_operation:: pop(void)
{
    if (top <= -1)
    {
        cout<<"\nUnderflow!!";
 }
    else
    {
        cout<<stack[top];
        top = top - 1;
    }
}
void stack_operation:: show(void)
{
    if (top <=0)
    {
        cout<<"\nUnderflow!!";
    }
    else
    {
        cout<<"\nElements present in the stack: \n";
        //stack_operation::push();
        for (int i = top; i > 0; i--)
            cout<<stack[i];
            cout<<endl;
    }
}
int main() {
 int choice;
 stack_operation sp;
cout<<"\nPerform operations on the stack:";

    while (1)
    {
         cout<<"\n1.Push the element\n2.Pop the element\n3.Show\n4.End";
        cout<<"\n\nEnter the choice: ";
        cin>>choice;

        switch (choice)
        {
        case 1:
            sp.push();
            break;
        case 2:
           sp.pop();
            break;
        case 3:
            sp.show();
            break;
        case 4:
            exit(0);

        default:
            printf("\nInvalid choice!!");
        }
    }

  return 0;
}
```

Output:

```
Perform operations on the stack:
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 1

Enter the element to be added onto the stack: 12

1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 1

Enter the element to be added onto the stack: 34
```

```
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 3

Elements present in the stack:
34
12



1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 2
34
1.Push the element
2.Pop the element
3.Show
4.End

Enter the choice: 3

Elements present in the stack:
12
```

Assignment:2

Problem Statement: Matrix operation(addition, subtraction,Multiplication) using member function.

Source Code:

```
#include<iostream>
using namespace std;
class Matrix
{
private:
int i,j,k,M1[3][3],M2[3][3],r,result[3][3];
public:
void create()
{
cout<<"Enter the values of First matrix"<< endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<" Enter the number of "<<i+1<<" Row and "<<j+1<<" Column\t";
cin>>M1[i][j];
}
}
cout<<"\tEnter the values of Second Matrix"<<endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<"Enter the number of "<<i+1<<"Row and "<<j+1<<"Column\t";
cin>>M2[i][j];
}
}
cout<<"The values of First Matirx is"<<endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<M1[i][j]<<"\t";
}
cout<<endl;
}
cout<<"The values of Second Matirx is"<<endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<M2[i][j]<<"\t";
}
cout<<endl;
}
}
void ADD()
{
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
result[i][j]=M1[i][j]+M2[i][j];
}
}
cout<<endl<<"The Addtion of these two Matrix are as follows…"<<endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<"\t"<<result[i][j]<<"\t";
}
cout<<endl;
}
}
void Subtraction()
{
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
```

```
{
result[i][j]=M1[i][j]-M2[i][j];
}
}
cout<<endl<<"The Subtraction of these two Matrix are as follows…"<<endl;
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
cout<<"\t"<<result[i][j]<<"\t";
}
cout<<endl;
}
}
void Multiplication()
{
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
result[i][j]=0;
}
}
for( i=0;i<3;i++)
{
for(j=0;j<3;j++)
{
for(k=0;k<3;k++)
{
r=M1[i][k]*M2[k][j];
result[i][j]=result[i][j]+r;
}
}
}
cout<<endl<<"The Multiplication of these two Matrix are as follow…"<<endl;
for( i=0;i<3;i++)
{
```

```
for(j=0;j<3;j++)
{
cout<<"\t"<<result[i][j]<<"\t";
}
cout<<endl;
}
}
};
int main()
{
Matrix a;
a.create();
a.ADD();
a.Subtraction();
a.Multiplication();
}
```

Output:

```
Enter the values of First matrix
 Enter the number of 1 Row and 1 Column: 12
 Enter the number of 1 Row and 2 Column :23
 Enter the number of 1 Row and 3 Column: 2
 Enter the number of 2 Row and 1 Column:3
 Enter the number of 2 Row and 2 Column :3
 Enter the number of 2 Row and 3 Column :4
 Enter the number of 3 Row and 1 Column: 44
 Enter the number of 3 Row and 2 Column :5
 Enter the number of 3 Row and 3 Column :6
 Enter the values of Second Matrix
Enter the number of 1Row and 1Column    :7
Enter the number of 1Row and 2Column   : 8
Enter the number of 1Row and 3Column  :  90
Enter the number of 2Row and 1Column  :  2
Enter the number of 2Row and 2Column  :  3
Enter the number of 2Row and 3Column   : 4
Enter the number of 3Row and 1Column   : 56
Enter the number of 3Row and 2Column    :6
Enter the number of 3Row and 3Column  :  7
The values of First Matirx is
12      23      2
3       3       4
44      5       6
The values of Second Matirx is
7       8       90
2       3       4
56      6       7

The Addtion of these two Matrix are as followsà
        19              31              92
        5               6               8
        100             11              13

The Subtraction of these two Matrix are as followsà
        5               15              -88
        1               0               0
        -12             -1              -1

The Multiplication of these two Matrix are as followà
        242             177             1186
        251             57              310
        654             403             402
```

Assignment:3

Problem Statement: Matrix operation(addition, subtraction, Multiplication) using operator overloading.

Source Code:

```
#include<iostream>

using namespace std;

class Matrix{

        int a[3][3];

        public:

        void display();

        void accept();

        void operator +(Matrix x);

        void operator *(Matrix x);

};

void Matrix::accept()

{

        cout<<"Enter row and column elements:";

        for(int i=0;i<3;i++)

        {

                for(int j=0;j<3;j++)

                {

                        cin>>a[i][j];

                }

cout<<endl;


        }

}


void Matrix::display()

{

        cout<<"The matrix is"<<endl;


        for(int i=0;i<3;i++)

        {

        for(int j=0;j<3;j++)

                        {
```

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

            }
        cout<<endl;
    }
}

//Addition
void Matrix::operator +(Matrix x)
{

    int mat[3][3];

    for(int i=0;i<3;i++)

    {

        for(int j=0;j<3;j++)
        {
         mat[i][j]=a[i][j] + x.a[i][j];

        }
    }

    cout<<"\n Addition of matrix:\n\n";

        for(int i=0;i<3;i++)
        {

            for(int j=0;j<3;j++)
                {
                cout<<"  ";
                cout<<mat[i][j]<<"  ";

                }
        cout<<endl;
```

```
             }
}
//mltiplication
void Matrix::operator *(Matrix x)
{
int mat[3][3];
       for(int i=0;i<3;i++)
       {
       for(int j=0;j<3;j++)
             {
              mat[i][j]=a[i][j] * x.a[i][j];
             }
       }
       cout<<"\n Multiplication of matrix:\n\n";
             for(int i=0;i<3;i++)
             {
                    for(int j=0;j<3;j++)
                           {
                           cout<<"  ";
                           cout<<mat[i][j]<<"  ";

                           }
             cout<<endl;
             }
}
int main()
{
       Matrix m,n;
       m.accept();//////////// accepting row
       n.accept();/////////// accepting column
       cout<<"\n First matrix:";
       m.display();
       cout<<"\n Second matrix:";
       n.display();
       m +n;    // operator overloaded
       m*n;
return 0;
```

```
}
```

Output:

```
Enter row and column elements:3

3

2


4

5

6


6

8

8


Enter row and column elements:1

2

4



5

5

7


8

90

1




 First matrix:The matrix is

  3    3    2

  4    5    6

  6    8    8


 Second matrix:The matrix is

  1    2    4

  5    5    7
  8    90    1


 Addition of matrix:


  4    5    6

  9    10    13

  14    98    9


 Multiplication of matrix:


  3    6    8

  20    25    42

  48    720    8
```

Assignment:4

Problem Statement: Matrix operation(addition, subtraction, Multiplication) using Friend Function.

Source Code:

```
#include<iostream>
using namespace std;
 class matrix2;
 class matrix1

{
            int a[5][5];
            int m,n;
  public:
            void getmatrix1();
            void putmatrix1();
            friend void matrixsum(matrix1,matrix2);
            friend void matrixsub(matrix1,matrix2);
            friend void matrixmul(matrix1,matrix2);
};

void matrix1::getmatrix1()

{
            int i,j;
            cout<<"MATRIX 1:\n";
            cout<<"Rows and Columns: ";
            cin>>m>>n;
            cout<<"Enter elements:\n";
            for(i=0;i<m;i++)
            for(j=0;j<n;j++)
            cin>>a[i][j];
}

void matrix1::putmatrix1()

{
            int i,j;
            cout<<"MATRIX 1:\n";
            for(i=0;i<m;i++)
            {
```

```
                        for(j=0;j<n;j++)
                         cout<<a[i][j]<<" ";
                         cout<<endl;
            }
}

class matrix2
{
            int b[5][5];
            int m,n;
public:
            void getmatrix2();
            void putmatrix2();
            friend void matrixsum(matrix1,matrix2);
             friend void matrixsub(matrix1,matrix2);
              friend void matrixmul(matrix1,matrix2);
};

void matrix2::getmatrix2()
{
            int i,j;
            cout<<"MATRIX 2:\n";
            cout<<"Rows and Colums: ";
            cin>>m>>n;
            cout<<"Enter elements:\n";
            for(i=0;i<m;i++)
            for(j=0;j<n;j++)
                        cin>>b[i][j];
}

void matrix2::putmatrix2()
{
            int i,j;
            cout<<"MATRIX 2:\n";
            for(i=0;i<m;i++)
            {
                        for(j=0;j<n;j++)
```

```
                                    cout<<b[i][j]<<" ";
                        cout<<endl;
            }
}

void matrixsum(matrix1 g,matrix2 h)
{

            int i,j;
            if(g.m==h.m && g.n==h.n)
            {
                         cout<<"SUM OF MATRICES:\n";
                           for(i=0;i<g.m;i++)
                        {

                                    for(j=0;j<g.n;j++)
                                                cout<<g.a[i][j]+h.b[i][j]<<" ";
                                    cout<<endl;
                        }
            }
            else
                        cout<<"Dimensions are not same...Addition not possible";
}
void matrixsub(matrix1 g,matrix2 h)
{

            int i,j;
            if(g.m==h.m && g.n==h.n)
            {
                         cout<<"SUBTRACTION OF MATRICES:\n";
                           for(i=0;i<g.m;i++)
                        {

                                    for(j=0;j<g.n;j++)
                                                cout<<g.a[i][j] - h.b[i][j]<<" ";
                                    cout<<endl;
                        }
            }
            else
                cout<<"Dimensions are not same...Addition not possible";
}
void matrixmul(matrix1 g,matrix2 h)
{

            int i,j;
double mul[i][j];
            if(g.m==h.m && g.n==h.n)
            {
                          cout<<"MULTIPLICATION OF MATRICES:\n";
                           for(i=0;i<g.m;i++)
                         {

                                     for(j=0;j<g.n;j++)
                                     {

                                          mul[i][j]=0;
                                                 for(int k =0; k<g.n ; k++)
                                                 {
                                                 mul[i][j]+= g.a[i][k] * h.b[k][j];

                                                }
                                     cout<<endl;
                                   }
                         }
                         for(int i =0 ; i<g.m;i++)
                         {
                         for(int j =0; j<h.n;j++)
                         {
                         cout<<mul[i][j]<<"  ";
                         }
        cout<<endl;

                      }
        }
```

```
       else
        cout<<"Dimensions are not same...Addition not possible";
}

int main()
{
            matrix1 m1;
            matrix2 m2;
            m1.getmatrix1();
            m2.getmatrix2();
            m1.putmatrix1();
            m2.putmatrix2();
            matrixsum(m1,m2);
            matrixsub(m1,m2);
            matrixmul(m1,m2);
       return 0;
}
```

Output:

```
MATRIX 1:
```

```
Rows and Columns: 2

2

Enter elements:

12

12

13

13

MATRIX 2:

Rows and Colums: 2

2

Enter elements:

12

12

13

13

MATRIX 1:

12 12

13 13

MATRIX 2:

12 12

13 13

SUM OF MATRICES:

24 24

26 26

SUBTRACTION OF MATRICES:

0 0

0 0

MULTIPLICATION OF MATRICES:

300 300

325 325
```

Assignment:5

Problem Statement: Matrix operation(addition, subtraction, Multiplication) using Constructor.

Source Code:

```
#include<iostream>
using namespace std;
class MATRIX{
    private:
    int a[10][10];
    int row, col;
    public:
    MATRIX(){}
    MATRIX(int m[10][10], int r, int c);
    void display();
    MATRIX addition(MATRIX M2);
    MATRIX subtraction(MATRIX M2);
    MATRIX multiplication(MATRIX M2);
};

MATRIX:: MATRIX(int m[10][10], int r, int c)
{
    row = r, col = c;
    for(int i = 0; i < r; i++)
    {
        for(int j = 0; j < c; j++)
        {
            a[i][j] = m[i][j];
        }
    }
}

MATRIX MATRIX :: addition(MATRIX M2)
{
    MATRIX t;
    t.row = M2.row;
    t.col = M2.col;
    for(int i = 0; i < t.row; i++)
    {
        for(int j = 0; j < t.col; j++)
        {
            t.a[i][j] = a[i][j] + M2.a[i][j];
        }
    }
    return t;
}

MATRIX MATRIX :: subtraction(MATRIX M2)
{
    MATRIX t;
    t.row = M2.row;
    t.col = M2.col;
    for(int i = 0; i < t.row; i++)
    {
        for(int j = 0; j < t.col; j++)
        {
            t.a[i][j] = a[i][j] - M2.a[i][j];
        }
    }

    return t;
}
void MATRIX :: display()
    {
        for(int i = 0; i < row; i++)
        {
            for(int j = 0; j < col; j++)
            {
                cout << a[i][j] << "\t";
            }
            cout << "\n";
        }
    }
MATRIX MATRIX :: multiplication(MATRIX M2)
{
    if(col == M2.row)
    {
        MATRIX t;
        t.row = row;
        t.col = M2.col;
        for(int i = 0; i < t.row; i++)
        {
            for(int j = 0; j < t.col; j++)
            {
                t.a[i][j] = 0;
                for(int k = 0; k < M2.col; k++)
                {
                    t.a[i][j] += a[i][k] * M2.a[k][j];
                }
            }
        }

        return t;
    }
    else
    {
        cout << "\ncannot multiply\n";
    }
}

int main()
{
    int r, c, i, j;
    int m[10][10];
    cout << ("Enter the size of the row: \n");
    cin >> r;
    cout << ("Enter the size of the column: \n");
    cin >> c;
    cout << "\nInput for first matrix:\n";
    for(i = 0; i < r; i++)
    {
        for(j = 0; j < c; j++)
        {
            cin >> m[i][j];
        }
    }
    MATRIX m1(m, r, c);
    cout << "\nInput for second matrix:\n";
    for(i = 0; i < r; i++)
    {
        for(j = 0; j < c; j++)
        {
            cin >> m[i][j];
        }
    }
    MATRIX m2(m, r, c), m3,m5, m4;
    cout << "matrix 1:\n";
    m1.display();
    cout << "matrix 2:\n";
    m2.display();

    m3 = m1.addition(m2);
    cout << " Addition of two matrices :\n";
    m3.display();

        m5 = m1.subtraction(m2);
    cout << "Subtraction of two matrices:\n";
    m5.display();

    m4 = m1.multiplication(m2);
    cout << "Multiplication of two matrices:\n";
    m4.display();
    return 0;
}
```

Output:

```
Enter the size of the row:
2
Enter the size of the column:
```

```
2

Input for first matrix:
12
13
12
13

Input for second matrix:
12
34
23
12
matrix 1:
12      13
12      13
matrix 2:
12      34
23      12
 Addition of two matrices :
24      47
35      25
Subtraction of two matrices:
0       -21
-11     1
Multiplication of two matrices:
443     564
443     564
```

Assignment:6

Problem Statement: Space Removal from a file.

Source Code:

```
#include<iostream>
using namespace std;
```

```
class Blank{
    char str[1000];
    public:
        void readdata()
        {
            int i=0;
            cout<<"Enter any string:";
            cin.getline(str,1000);
        }
        void count(){
            int i=0, countblank=0;
                while(str[i]!='\0')
                {
                    if(str[i]== ' ')
                    {
                        countblank++;
                        i++;
                        continue;
                    }
                    if(countblank>=1)
                    {
                        cout<<" ";
                        cout<<str[i];
                        countblank=0;
                    }
                    else
                    {
                        cout<<str[i];
                    }
                    i++;
                }
        }
};

int main()
{
    Blank bk;
```

```
       bk.readdata();
       bk.count();
       return 0;
}
```

Output:

```
Enter any string: My name is                        Ammrisha
My name is Ammrisha
```

Assignment:7

Problem Statement: Implement Inheritance for the given problem .

Source Code:

```
#include<iostream>
using namespace std;
class student{
protected:
string name;
int Roll;
public:
void getdata(void);
void putdata(void);
};
//student class's getdata() and putdata()
void student::getdata(void)
{
 cout<<"Enter your name: "<<endl;
 cin>>name;
 cout<<"Enter your roll: "<<endl;
 cin>>Roll;
}
void student::putdata(void)
{
cout<<name <<" "<<Roll;
}
//inherits test class
class test : virtual public student
{
protected:
float marks1 , marks2;
public:
void getdata()
{
cout<<name<<" "<<"Enter your first Marks: "<<endl;
cin>>marks1;
cout<<"Enter your second Marks: "<<endl;
cin>>marks2;
}
// inherits the student class as sports
void putdata()
{
 cout<<"Your first marks: " <<marks1<<endl;
cout<<"Your Second marks: "<<marks2<<endl;
}
};
class sports:virtual public student
{
protected :
float score;
public:
void getdata()
{
cout<<"Your score:";
cin>>score;
}
void putdata()
{
cout<<"Score is:"<<score;
}
};
// inherits the student class as result
class result:public test, public sports
{
protected:
float Total;
public:
void getdata()
{
student::getdata();
test::getdata();
sports::getdata();
Total = marks1 +marks2+score;
}
```

```
void putdata(void)
{
student::putdata();
test::putdata();
sports::putdata();
cout<<"Your Total marks is:"<<Total<<endl;
}
};
int main()
{
result R;
R.getdata();
R.putdata();
return 0;
}
```

Output:

```
Enter your name:
Ammrisha
Enter your roll:
26
Ammrisha Enter your first Marks:
80
Enter your second Marks:
70
Your score:13
Ammrisha 26Your first marks: 80
Your Second marks: 70
Score is:13Your Total marks is:163
```

Assignment:8

Problem Statement: Implement Inheritance for the given problem .

There is a Publication class whose data member is title ,price and member functions are getdata() & putdata(). There are other classes who are Book & Video both are derived from Publication class and both has same member functions who are getdata() & putdata().But in Book class there is a data member which is No of pages and in Video class there is a data member which is duration of video.

Source Code:

```
#include<iostream>

using namespace std;

class Publication{

	protected:

		string Title_video;

		string Title_book;

		int book_price, video_price;

	public:

	void getdata()

{

	cout<<"Enter the Title of the Book:";

	cin>>Title_book;

	cout<<"Enter the book price:";

	cin>>book_price;

	cout<<"Enter the Title of the Video:";

	cin>>Title_video;

	cout<<"Enter the video price:";

	cin>>video_price;

}

void putdata()

{

	cout<<"The "<<Title_book<<" "<<"price is: "<<book_price<<endl;

	cout<<"The "<<Title_video<<" "<<"price is: "<<video_price<<endl;


}

};

class video: public Publication{

	protected:

			int duration_video;
```

```
    public:

        void getdata(){
            Publication::getdata();

        cout<<"Enter the duration of the video:";
        cin>>duration_video;
        }
        void putdata(){
            Publication::putdata();
            cout<<"And the duration is"<<" "<<duration_video<<"hours";

        }
};

class Book:  virtual public Publication
{
    protected:
        int page_no;
    public:
        void getdata(){
            cout<<endl<<"Enter the number of pages in the book:"<<"\t";
            cin>>page_no;
        }
        void putdata(){
            cout<<endl<<"page number is:"<<"\t"<<page_no;
        }
};
int main()
{
    video vo;
    Book bk;
    video vb;
    vb.getdata();
    vb.putdata();
    bk.getdata();
    bk.putdata();
```

```
       return 0;
}
```

Output:

```
Enter the Title of the Book:    C++

Enter the book price:   1200

Enter the Title of the Video:   CPP

Enter the video price:  700

Enter the duration of the video:        2

The C++ price is:1200

The CPP price is:700

And the duration is 2

Enter the number of pages in the book:  1290


page number is: 1290
```

Assignment:9

Problem: Write a C++ program using a class Array to implement Array operations(search, display, sort, reverse).

Source Code:

```
#include<iostream>

 using namespace std;

  class Array {

       int size;

       int ar[10];

        public:

       void initialize(){

                                 cout<<"Enter the no of elements in array: ";

                                 cin>>size;

                                 int i;

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

                                        cout<<"Enter the element";

                                        cin>>ar[i];

                                 }

                }


       void display()

        {

              int d;

              cout<<"The array elements are: ";

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

                       {

                            cout<<ar[d]<<" ";

                            }

              cout<<endl;

       }


       void search()

        {

                     int s,b;

                     cout<<"Enter the element to be searched: ";

                     cin>>s;

              for(b=0;b<size;b++)
```

```
         {
                     if(ar[b] == s)
                        {
                            cout<<"Element is found "<<endl;
                            break;
                            }
              }
       if(b==size)
         {
              cout<<"Element not found "<<endl;
       }
}
void sort()
 {
       int i,j,swap;
              for(i=0;i<size;i++)
               {
                     for(j=i;j<size;j++)
                      {
                            if(ar[j]<ar[i])
                             {
                            swap=ar[j];
                            ar[j]=ar[i];
                            ar[i]=swap;
                             }
                      }
            }
       cout<<"Elements are sorted "<<endl;
}
void reverse(){
       int i,j,r;
       i=size-1;
       j=0;
       while(j<i)
         {
              r=ar[j];
              ar[j]=ar[i];
            ar[i]=r;

            j++;
            i--;
        }
        cout<<"Elements are reversed "<<endl;
    }
};
int main(){
    int n;  Array a;
    a.initialize();
     while(1){
        cout<<"---------------------"<<endl;
        cout<<"Enter 1 to display"<<endl;
        cout<<"Enter 2 to search"<<endl;
        cout<<"Enter 3 to sort"<<endl;
        cout<<"Enter 4 to reverse"<<endl;
        cout<<"0 to exit"<<endl;
        cin>>n;

switch(n){
        case 1:a.display();
            break;
            case 2:a.search();
                break;
            case 3:a.sort();
                break;
            case 4:a.reverse();
                break;
            case 0:exit(0);
            default:cout<<"Wrong option"<<endl;
        }
    }

    return 0;
}
```

Output:

```
Enter the no of elements in array: 2

Enter the element1

Enter the element2

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

Enter 1 to display

Enter 2 to search

Enter 3 to sort

Enter 4 to reverse

0 to exit

1

The array elements are: 1 2

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

Enter 1 to display

Enter 2 to search

Enter 3 to sort

Enter 4 to reverse

0 to exit

2

Enter the element to be searched: 3

Element not found

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

Enter 1 to display

Enter 2 to search

Enter 3 to sort

Enter 4 to reverse

0 to exit

3

Elements are sorted

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

Enter 1 to display

Enter 2 to search

Enter 3 to sort

Enter 4 to reverse

0 to exit

3

Elements are sorted
```

```
---------------------
Enter 1 to display
Enter 2 to search
Enter 3 to sort
Enter 4 to reverse
0 to exit
1
The array elements are: 1 2
---------------------
Enter 1 to display
Enter 2 to search
Enter 3 to sort
Enter 4 to reverse
0 to exit
4
Elements are reversed
---------------------
Enter 1 to display
Enter 2 to search
Enter 3 to sort
Enter 4 to reverse
0 to exit
1
The array elements are: 2 1
---------------------
Enter 1 to display
Enter 2 to search
Enter 3 to sort
Enter 4 to reverse
0 to exit
```

Assignmnet:10

Problem : Write a C++ program using a class AREA to find the area of circle, rectangle, and scalene triangle with the help of function overloading concept

Source Code:

```
#include<iostream>

#include<cstdlib>

 #include<cmath>

using namespace std; class AREA{

       public:

       void area(int);

       void area(int,int);

       void area(int,int,int);

};

void AREA::area(int r){

  float a;

  a= 3.14159*r*r;

       cout<<"Area of the circle of radius "<<r<<" is:: "<<a;

}

void AREA::area(int x, int y){

       int a;  a= x*y;

       cout<<"Area of the rectangle of length "<<x<<"and width "<<y<<" is:: "<<a;

}

void AREA::area(int a, int b, int c){

       float s,ar,z;

       s=(a+b+c)/2;

       z=s*(s-a)*(s-b)*(s-c);

        ar= sqrt(z);

        cout<<"Area of the scalene triangle of edges "<<a<<", "<<b<<", "<<c<<" is:: 
"<<ar;

}

int main(){

  AREA area;

  int a,b,c,n;

  while(1){

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

       cout<<"Enter '0' to exit\nEnter '1' to find area of a circle\nEnter '2' to 
find area of a rectangle\nEnter '3' to find area of a scalene triangle\n";
```

```
        cout<<"Enter your choice:: ";
        cin>>n;
         switch(n){
                case 1:
                        cout<<"Enter radius: ";
                        cin>>a;
                        area.area(a);
                break;
                case 2:
                        cout<<"Enter length: ";
                        cin>>a;
                        cout<<"Enter width: ";
                        cin>>b;
                        area.area(a,b);
                break;
                case 3:
                        cout<<"Enter values of three edges: ";
                        cin>>a>>b>>c;
                        area.area(a,b,c);
                        break;
                case 0: exit(0);
                default: cout<<"Invalid Choice!! try again";
        }
        }
}
```

Output:

```
Enter '0' to exit
Enter '1' to find area of a circle
Enter '2' to find area of a rectangle
Enter '3' to find area of a scalene triangle
Enter your choice:: 1
Enter radius: 12
Area of the circle of radius 12 is:: 452.389
-----------------------------------------
Enter '0' to exit
Enter '1' to find area of a circle

Enter '2' to find area of a rectangle

Enter '3' to find area of a scalene triangle

Enter your choice:: 3

Enter values of three edges: 12

12

12

Area of the scalene triangle of edges 12, 12, 12 is:: 62.3538

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

Enter '0' to exit

Enter '1' to find area of a circle

Enter '2' to find area of a rectangle

Enter '3' to find area of a scalene triangle

Enter your choice:: 0
```

Assignment:11

Problem: Write a program in C++ to compare and concatenate two string.

Source Code:

```
#include<iostream>

 #include<string>

 using namespace std;

 class String{

                    string s;

      public:

                    string getstring();

             string concate(string s1,string s2);

             int comp(string s1,string s2);

             void display();

};


string String::getstring(){

      cout<<"Enter a String: ";

 getline(cin,s);

 return s;

}

string String::concate(string s1,string s2){

      s=s1+s2;

 return s;

}

int String::comp(string s1,string s2)

{

cout<<"String comparison result"<<endl;

int x= s1.compare(s2);

if(x==0){

             cout<<"Strings are equal"<<endl;

      }

      else if(x>0){
```

```
 cout<<s1<<"greater than "<<s2<<endl;

        }

        else

        {

                cout<<s1<<"less than"<<s2<<endl;

        }

}

void String::display(){

        cout<<"String concatenation result:"<<endl;


        cout<<s;

}

int main(){

        String a,b,c;

 string s1,s2,s3;

 s1=a.getstring();

 s2=b.getstring();

        c.comp(s1,s2);

        c.concate(s1,s2);

        c.display();

 return 0;

}
```

Output:

```
Enter a String: Hello World
Enter a String: Ammrisha
String comparison result
Hello Worldgreater than Ammrisha
String concatenation result:
Hello WorldAmmrisha
```

Assignment:12

Problem :Write a program in C++ using class to allow the statement S!+=S2 using += operator; where S2 is added(concatenated) to S1 and the result left in S1. The operator should also permit the result of the operation to be used in other calculation as in S3=S1+=S2.

Source Code:

```
#include<iostream>

#include<string>

using namespace std;

class String{

       string s;

       public:


     void getstring();

        String operator +=(String);

        void display();

};


void String::getstring(){

       cout<<"Enter a String: ";

        getline(cin,s);

}

String String::operator +=(String s2){

       String tmp;

        tmp.s= s= s+" "+s2.s;

        return tmp;

}

void String::display(){

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

cout<<s;

}

int main(){

       String a,b,c;

       a.getstring();

       b.getstring();

        c= a+=b;

       a.display();

       c.display();
```

```
        return 0;
}
```

Output:

```
Enter a String: My name is
Enter a String: Ammrisha Chowdhury

-------------------------
My name is Ammrisha Chowdhury
-------------------------
My name is Ammrisha Chowdhury
```

Assignment:13

Problem: Write a program in C++ to compare and concatenate two string using dynamic initialization, constructor and operator overloading.

Source code:

```
#include <iostream>

 #include <cstring>

 using namespace std;

class String {

 private:

    char* str;

 public:

    String(const char* s) {

int length = strlen(s);

str = new char[length + 1];

strcpy(str, s);

    }

    int compare(const String& other) const {

return strcmp(str, other.str);

    }

    String operator+(const String& other) const {

int length1 = std::strlen(str);

int length2 = std::strlen(other.str);

char* concatenatedString = new char[length1 + length2 + 1];

strcpy(concatenatedString, str);

strcat(concatenatedString, other.str);

return String(concatenatedString);

    }

    const char* getString() const {

return str;

    }

};


int main() {
```

```
char input1[100],
input2[100];
cout << "Enter string 1: ";
cin.getline(input1, sizeof(input1));
cout << "Enter string 2: ";
cin.getline(input2, sizeof(input2));
    String S1(input1);
    String S2(input2);
String S3 = S1 + S2;
 int comparisonResult = S1.compare(S2);
 if (comparisonResult == 0) {
 cout << "The strings are equal." <<endl;
 }
 else if (comparisonResult < 0)
  {
  cout << "String 1 is less than string 2." << endl;

      }
 else {
   cout << "String 1 is greater than string 2." <<endl;

         }
    cout << "Concatenated string: " << S3.getString() <<endl;
       return 0;
}
```

Output:

```
Enter string 1: My name is
Enter string 2: Ammrisha
String 1 is greater than string 2.
Concatenated string: My name isAmmrisha
```

Assignment:14

Problem-: Write a C++ program to add two given Time objects using operator overloading.

Source Code:

```
#include<iostream>

 using namespace std;

 class Time

{

       int h,m,s;

 public:

       Time ()

       {

 h=0;

 m=0;

 s=0;

 }

       Time(int x,int y,int z)

       {

 h=x;

 m=y;

 s=z;

       }

       void display(void);

       Time operator+(Time);

};

void Time::display()

{

       cout<<"\n The Time is:";

       if(h<10)

       {

              cout<<"0";

       }

              cout<<h<<":";
              if(m<10)
       {
              cout<<"0";
       }
              cout<<m<<":";
       if(s<10)
       {
              cout<<"0";
       }
              cout<<s<<endl;
}
Time Time::operator+(Time t)
{
       Time k;
       k.h=h+t.h;
       k.m=m+t.m;
       k.s=s+t.s;
 if(k.s>59)
       {
              k.m+=(k.s/60);
              k.s=k.s%60;
       }
       if(k.m>59)
       {
              k.h+=(k.m/60);
       k.m=k.m%60;
       }
       return(k);
}
int main()
{
 int h,m ,s;
 Time add;

 cout<<"Enter the Time1:";

 cin>>h>>m>>s;

 Time t1(h,m,s);

 t1.display();

 cout<<"Enter the Time2:";

 cin>>h>>m>>s;

 Time t2(h,m,s);

 t2.display();

 add=t1+t2;

 add.display();

 return 0;

}
```

Output:

```
Enter the Time1:12

48

35


 The Time is:12:48:35

Enter the Time2:13

23

45


 The Time is:13:23:45


 The Time is:26:12:20
```

Assignment:15

Problem-: In an office all the staffs get Basic and HRA, but managers get additional allowance. The office has branches in Kolkata, Delhi, and Darjeeling. In Kolkata office all the staffs get special allowance, in Delhi city allowance and in Darjeeling hill allowance.
Write a program in C++ to get data and show the data for each branch.

Source Code:

```
#include<iostream>

using namespace std;

class Office

{

	protected:

	int basic,hra,a=0;

	public:

 void getdata();

 void putdata();

};

void Office::getdata()

{

	bool k;

	cout<<"\nEnter the basic salary of employee:\n ";

	cin>>basic;

	cout<<"\nEnter the HRA of employee:\n";

	cin>>hra;

	cout<<"if the employee is manager then enter'1' :\n";

	cin>>k;

	 if(k==true)

	{

	cout<<"\nEnter additional allowance:\n";

	cin>>a;

	}

}

void Office::putdata()

{

	cout<<"\nThe basic salary of the employee is:\n ";

	cout<<basic;

	cout<<"\nThe HRA of the employee is:\n ";

	cout<<hra;
```

```
        cout<<"\nThe additional allowance of the employee is:\n ";
        cout<<a;
}
class kolkata:public Office
{
        public:
         float k;
         void getdatak()
        {
        cout<<"\nEnter the city allowance:\n";
        cin>>k;
        }
        void putdatak()
        {
        cout<<"\nThe city allowance is:\n"<<k;         cout<<"\nTotal Salary of the employee:\n"<<basic+hra+a+k<<endl;
        }
};
class delhi:public Office
{
        public:
         float d;
        void getdata_d()
        {
        cout<<"\nEnter the city allowance:\n";
        cin>>d;
        }
        void putdata_d()
        {
        cout<<"\nThe city allowance is:\n"<<d;
         cout<<"\nTotal Salary of the employee:\n"<<basic+hra+a+d<<endl;
        }
};
class dargeeling:public Office
{
        public:
         float l;
         void getdata_l()
        {
        cout<<"\nEnter the city allowance:\n";
        cin>>l;
        }
        void putdata_l()
        {
        cout<<"\nThe city allowance is:\n"<<l;
                cout<<"\nTotal Salary of the employee:\n"<<basic+hra+a+l<<endl;
        }
};
int main()
{
        kolkata a;
        delhi m;
        dargeeling n;
        cout<<"Enter the employee information who work in Kolkata:\n";
        a.getdata();
        a.getdatak();
        cout<<"Enter the employee information who work in Delhi:\n";
        m.getdata();
        m.getdata_d();
        cout<<"Enter the employee information who work in Darjeeling:\n";
        n.getdata();
        n.getdata_l();
         cout<<" Employee information who work in Kolkata:\n";
        a.putdata();
        a.putdatak();
        cout<<"Employee information who work in Delhi:\n";
        m.putdata();
        m.putdata_d();
        cout<<"Employee information who work in Darjeeling:\n";
        n.putdata();
        n.putdata_l();
         return 0;
}
```

Output:

```
Enter the employee information who work in Kolkata:

Enter the basic salary of employee:
 10000

Enter the HRA of employee:
2000
if the employee is manager then enter'1' :
1

Enter additional allowance:
1000

Enter the city allowance:
1500
Enter the employee information who work in Delhi:

Enter the basic salary of employee:
 10000

Enter the HRA of employee:
2000
if the employee is manager then enter'1' :
0

Enter the city allowance:
2000
Enter the employee information who work in Darjeeling:

Enter the basic salary of employee:
 10000

Enter the HRA of employee:
2000
if the employee is manager then enter'1' :
1
```

```
Enter additional allowance:
2000

Enter the city allowance:
500
 Employee information who work in Kolkata:

The basic salary of the employee is:
 10000
The HRA of the employee is:
 2000
The additional allowance of the employee is:
 1000
The city allowance is:
1500
Total Salary of the employee:
14500
Employee information who work in Delhi:

The basic salary of the employee is:
 10000
The HRA of the employee is:
 2000
The additional allowance of the employee is:
 0
The city allowance is:
2000
Total Salary of the employee:
14000
Employee information who work in Darjeeling:

The basic salary of the employee is:
 10000
The HRA of the employee is:
 2000
The additional allowance of the employee is:
 2000

The city allowance is:

500

Total Salary of the employee:

14500
```

Assignment:16

Problem: Write a C++ program using class to read a text file and replace two or more consecutive blanks with a single blank and first letter of each word in uppercase to another file.

Source Code:

```
#include<iostream>

#include<fstream>

using namespace std;

class FIL1

{

        char ch;

        public:

        void createfile(string fn);

        void displayfile(string fn);

        void updatefile(string fn, string fn1);

};

void FIL1::createfile(string fn)

       {

              ofstream fout;

              fout.open(fn);

              cout<<"Enter text";

              while((ch=cin.get())!='\n')

              fout<<ch;

              fout.close();

       }

void FIL1::displayfile(string fn)

       {

             ifstream fin;

                fin.open(fn);

                cout<<"Displayig: "<<endl;

                ch=fin.get();

                while(fin)

                {

                     cout<<ch;

                     ch=fin.get();

                }
             fin.close();
               cout<<endl;
       }
void FIL1::updatefile(string fn1,string fn2)
{
        ifstream fin(fn1);
        ofstream fout(fn2);
        if(fin.fail()!=0)
              {
               cout<<"\nFile open error\n";
              }
        else
        {
         int flag=0,flag1=0;   fin.get(ch);
         do
         {
          if(ch!=' ')
           {
               if(flag1==0)
               {
                   fout<<char(int(ch)-32);
                           flag1=1;
               } else

               {
                   fout<<ch;
               }
               flag=1;
           }
             else
                 {
                     if(ch==' ' && flag==1)
                     {
                         fout<<"  ";
                                 flag=0;
                                 flag1=0;
                     }
```

```
                }
          fin.get(ch);
          }
          while(fin);
        }
        fin.close();  fout.close();
}
int main(){

             FIL1 f1;
             f1.createfile("abc.txt");
             cout<<"Content of input file"<<endl;
             f1.displayfile("abc.txt");
             f1.updatefile("abc.txt","pqr.txt");
             cout<<"content of output file"<<endl;
             f1.displayfile("pqr.txt");
        return 0;
}
```

Output:

```
Enter textLorem Ipsum is simply dummy     text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown            printer took a galley of       type and
scrambled it to make a type specimen book Content of input file Displayig:

Lorem Ipsum is simply dummy     text of the printing and typesetting
industry. Lorem Ipsum has been the industry's standard dummy text ever since the
1500s, when an unknown            printer took a galley of       type and
scrambled it to make a type specimen book content of output file Displayig:

,orem  )psum  Is  Simply  Dummy  Text  Of  The  Printing  And  Typesetting
Industry.  ,orem  )psum

Has  Been  The  Industry's  Standard  Dummy  Text  Ever  Since  The  500s,  When
An  Unknown

Printer  Took  A  Galley  Of  Type  And  Scrambled  It  To  Make  A  Type  Specimen
Book
```

Assignment:17

Problem: Create 2 text files COUNTRY and CAPITAL . COUNTRY file contains name of atleast 5 countries and CAPITAL file contains corresponding capital names. Write a C++ program using class to get the output of the following patterns.

The Capital of India is New Delhi

Source Code:

```
#include<iostream>

#include<fstream>

using namespace std;

class cl1

{

       string ch1,ch2;

 public:

        void createfile(string fn1,string fn2);

 void displayfile(string fn1,string fn2);

};

void cl1::createfile(string fn1,string fn2)

       {

             ofstream fout1;

         ofstream fout2;

         fout1.open(fn1);

         fout2.open(fn2);

         int i,c1;

             cout<<"Enter number of country:";

             cin>>c1;

         cin.get();

             for(i=0;i<c1;i++)

             {

                    cout<<"Enter the country name";

                    getline(cin,ch1);

               fout1<<ch1<<endl;

             cout<<"Enter the capital name";

                    getline(cin,ch2);
```

```
                fout2<<ch2<<endl;


             }

             fout1.close();

        fout2.close();

       }

void cl1::displayfile(string fn1,string fn2)

       {

             ifstream Fin1,Fin2;


             Fin1.open(fn1);

             Fin2.open(fn2);

             if(Fin1.fail()!=0 && Fin2.fail()!=0)

             {

                    cout<<"File opening error\n";

             }

             else {

             cout<<"Displayig: "<<endl;

        getline(Fin1,ch1);

        getline(Fin2,ch2);

        while(Fin1.eof()==0 && Fin2.eof()==0){

        cout<<"The capital of "<<ch1<<" is "<<ch2<<endl;

                      getline(Fin1,ch1);

                getline(Fin2,ch2);


             }

             }

             Fin1.close();

             Fin2.close();

             cout<<endl;

       }
```

```
int main(){
 cl1 f1;
f1.createfile("country.txt","capital.txt");
f1.displayfile("country.txt","capital.txt");
    return 0;
}
```

Output:

```
Enter number of country:6
Enter the country nameIndia
Enter the capital nameNew Delhi
Enter the country nameChina
Enter the capital nameBeijing
Enter the country nameFrance
Enter the capital nameParis
Enter the country nameGermany
Enter the capital nameBerlin
Enter the country nameIreland
Enter the capital nameDublin
Enter the country namePortugal
Enter the capital nameLisbon Displayig:
The capital of India is New Delhi
The capital of China is Beijing
The capital of France is Paris
The capital of Germany is Berlin
The capital of Ireland is Dublin
The capital of Portugal is Lisbon
```