**NAME: CHANDRIMA BHATTACHARYA**

**EXAM ROLL NO: 111308009**

**CLASS ROLL NO: HX-05**

**DEPARTMENT: I.T.**

**SEMESTER: 5**

**DATE OF ASSIGNMENT: 07/08/2015 & 14/08/2015**

**1.8. Re-write the time addition program using friend function.**

**Source code:**

#include <iostream>

#include <string.h>

using namespace std;

class time\_c

{

int hour;

int minute;

public:

void gettime(int hr, int min)

{

hour=hr;

minute=min;

}

void display()

{

cout<<"\n Time= "<<hour<<"::"<<minute;

}

friend time\_c sum(time\_c t1,time\_c t2);

};

time\_c sum(time\_c t1, time\_c t2)

{

int hour,minute;

hour=t1.hour+t2.hour;

minute=t1.minute+t2.minute;

if(minute>59)

{

hour+=1;

minute%=60;

}

if(hour>=24)

hour%=24;

cout<<"\n Time= "<<hour<<"::"<<minute;

}

int main()

{

int h1,h2,m1,m2;

time\_c t1;

time\_c t2;

cout<<"\n Enter time as hour and minute";

cin>>h1>>m1;

cout<<"\n Enter second time as hour and minute ";

cin>>h2>>m2;

t1.gettime(h1,m1);

t2.gettime(h2,m2);

t1.display();

t2.display();

sum(t1,t2);

return 0;

}

**Output:**

Enter time as hour and minute 23 40

Enter second time as hour and minute 21 30

Time= 23::40

Time= 21::30

Time= 21::10

**1.9. Write C++ program to define matrix and vector class, to use function with default argument and to do matrix -vector multiplication using friend function.**

**Declare vector Class**

**Define matrix Class**

**Declare friend function multiply() inside the matrix class**

**Define vector Class**

**Declare friend function multiply(matrix &, vector &) inside the vector class**

**Define getvector() function with for loop to get the elements for vector**

**Define disvector() function with for loop to display the contents of vector**

**Define getmatrix() function with nested for loops to get the matrix elements**

**Define dismatrix() function with nested for loops to display the matrix**

**Define the multiply() to multiply matrix and vector**

**• a. No of columns in the matrix should be equal to no. of elements in the vector • b. Apply the matrix-vector multiplication mechanism: • For simplicity take the matrix as 3X3 and the vector as 1X3.**

**1,2,3,6, and 9 are basic category and 4,5,7 and 8 are intellectual category.**

**Source Code:**

#include <iostream>

using namespace std;

class vctr;

class matrix

{

int a[10][10],r,c;

public:

void getmatrix()

{

int i,j;

cout<<"\n enter no of row and columns";

cin>>r>>c;

for(i=1;i<=r;i++)

for(j=1;j<=c;j++)

{

cout<<"\n Enter the "<<i<<" row and"<<j<<" column value";

cin>>a[i][j];

}

}

void dismatrix()

{

int i,j;

for(i=1;i<=r;i++)

{ cout<<"\n";

for(j=1;j<=c;j++)

cout<<a[i][j]<<"\t";

}

}

friend void multiply(matrix &,vctr &);

};

class vctr

{

int a[10];

int c;

public:

void getvector()

{

int i;

cout<<"\n Enter no of columns such that no of columns in vector=no of rows in matrix";

cin>>c;

for(i=1;i<=c;i++)

{

cout<<"\n Enter the "<<i<<"column value";

cin>>a[i];

}

}

void disvector()

{

int i;

for(i=1;i<=c;i++)

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

}

friend void multiply(matrix &,vctr &);

};

void multiply(matrix &m,vctr &v)

{

int i,j,a[10]={0};

if(m.r!=v.c)

cout<<"\n Wrong value- cant proceed";

else

{

for(i=1;i<=m.r;i++)

{

cout<<"\n";

for(j=1;j<=v.c;j++)

//i]+=m.a[i][j]\*v.a[j];

a[i]+=(m.a[i][j])\*(v.a[j]);

}

for(i=1;i<=m.r;i++)

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

}

}

int main()

{

matrix m;

vctr v;

m.getmatrix();

m.dismatrix();

v.getvector();

v.disvector();

multiply(m,v);

}

**Output:**

enter no of row and columns 3 3

Enter the 1 row and1 column value1

Enter the 1 row and2 column value2

Enter the 1 row and3 column value3

Enter the 2 row and1 column value1

Enter the 2 row and2 column value2

Enter the 2 row and3 column value3

Enter the 3 row and1 column value1

Enter the 3 row and2 column value2

Enter the 3 row and3 column value3

1 2 3

1 2 3

1 2 3

Enter no of columns such that no of columns in vector=no of rows in matrix3

Enter the 1column value1

Enter the 2column value2

Enter the 3column value3

1 2 3

14 14 14

**1.10. Write a program using friend function to swap the contents of private data members of two classes.[a) With and a) without using third variable . C) Without using addition or substraction ]**

**Source code (a)**

#include <iostream>

using namespace std;

class data

{

int a;

public:

void getval()

{

cout<<"\n Enter value of no";

cin>>a;

}

void disp()

{

cout<<a<<"\t";

}

friend void swapping(data &, data &);

};

void swapping(data &s,data &p)

{

int c;

c=s.a;

s.a=p.a;

p.a=c;

cout<<"\n New swapped values are "<<s.a<<"\t"<<p.a;

}

int main()

{

data s;

data p;

s.getval();

p.getval();

s.disp();

p.disp();

swapping(s,p);

return 0;

}

**Output(a):**

Enter value of no3

Enter value of no5

3 5

New swapped values are 5 3

**Source code (b):**

#include <iostream>

using namespace std;

class data

{

int a;

public:

void getval()

{

cout<<"\n Enter value of no";

cin>>a;

}

void disp()

{

cout<<a<<"\t";

}

friend void swapping(data &, data &);

};

void swapping(data &s,data &p)

{

s.a=s.a+p.a;

p.a=s.a-p.a;

s.a=s.a-p.a;

cout<<"\n New swapped values are "<<s.a<<"\t"<<p.a;

}

int main()

{

data s;

data p;

s.getval();

p.getval();

s.disp();

p.disp();

swapping(s,p);

return 0;

}

**Output (b):**

Enter value of no24

Enter value of no21

24 21

New swapped values are 21 24

**Source code (c) :**

#include <iostream>

using namespace std;

class data

{

int a;

public:

void getval()

{

cout<<"\n Enter value of no";

cin>>a;

}

void disp()

{

cout<<a<<"\t";

}

friend void swapping(data &, data &);

};

void swapping(data &s,data &p)

{

s.a=s.a^p.a;

p.a=s.a^p.a;

s.a=s.a^p.a;

cout<<"\n New swapped values are "<<s.a<<"\t"<<p.a;

}

int main()

{

data s;

data p;

s.getval();

p.getval();

s.disp();

p.disp();

swapping(s,p);

return 0;

}

**Output (c) :**

Enter value of no33

Enter value of no53

33 53

New swapped values are 53 33

**ASSIGNMENT 2**

**2.1.** **1. Write a C++ program to add two complex numbers.**

**i) The class Complex contains three constructors.**

**a) One with no parameter. (Used for the object for storing result.)**

**b) With one parameter(Same value for real and imaginary part)**

**c) With two parameters.**

**and**

**ii) Two friend functions**

**a) One to add two complex number by taking two reference variables of class complex and returning another reference.**

**b) To display the result.**

**Source Code:**

#include <iostream>

#include <cstring>

using namespace std;

class complex\_c

{

int real;

int comp;

public:

complex\_c()

{

real=0;

comp=0;

}

complex\_c(int no)

{

real=no;

comp=no;

}

complex\_c(int r,int i)

{

real=r;

comp=i;

}

friend void display(complex\_c c);

friend complex\_c sum(complex\_c c1, complex\_c c2);

};

void display(complex\_c c)

{

char sign;

if(c.comp>=0)

sign='+';

else

sign='-';

cout<<"\n complex no =\t"<<c.real<<sign<<c.comp<<"i \n";

}

complex\_c sum(complex\_c c1,complex\_c c2)

{

complex\_c co;

char sign;

co.real=c1.real+c2.real;

co.comp=c1.comp+c2.comp;

return co;

}

int main()

{

int r1,r2,i2;

complex\_c c;

cout<<"\n Enter value in format no=x+ix";

cin>>r1;

cout<<"\n Enter value in format no=x+iy ";

cin>>r2>>i2;

complex\_c c1=complex\_c(r1);

complex\_c c2=complex\_c(r2,i2);

display(c1);

display(c2);

c=sum(c1,c2);

display(c);

return 0;

}

**Output:**

Enter value in format no=x+ix

>>3

Enter value in format no=x+iy

>>5

>>-1

Complex no=3+3i

Complex no=5-1i

Complex no=8+2i

**2.2 A Bank gives 4% interest on current account and 6% interest on savings account. An additional 3% interest is provided for savings duration of 5 years and above. Using dynamic initialization of constructor write banking program using C++.**

**Source Code:**

#include <iostream>

#include <cstring>

using namespace std;

class banking

{

char account;

int rate;

int addrate;

int time;

float amount;

public:

banking(char acc,float a,int t)

{

account=acc;

if(account=='s')

rate=6;

else

rate=4;

time=t;

if(time>=5 & account=='s')

addrate=3+rate;

else

addrate=rate;

amount=a;

amount=amount+(amount\*rate\*time/100);

}

void display()

{

cout<<"\n Customer is getting amount=Rs "<<amount<<" is getting "<<rate<<"% interest rate and as additional total of”<<addrate<<"% interest rate for account type "<<account<<" keeping in the bank for "<<time<<" years";

}

};

int main()

{

char acc;

int t;

float a;

cout<<"\n Enter s for savings and c for current type";

cin>>acc;

cout<<"Enter time and amount in Rs and paisa";

cin>>t;

cin>>a;

banking acnt(acc,a,t);

acnt.display();

return 0;

}

**Output:**

**CASE 1: type ‘s’ and time>=5**

Enter s for savings and c for current type

>>s

Enter time and amount in Rs and paisa

>>5

>>20000.00

Customer is getting Rs 29000.00 is getting rate 6% interest rate and as additional total of 9% rate of interest for account type s and keeping in bank for 5 years.

**CASE 2: type ‘s’ and time<5**

Enter s for savings and c for current type

>>s

Enter time and amount in Rs and paisa

>>1

>>20000.00

Customer is getting Rs 21200.00 is getting rate 6% interest rate and as additional total of 6% rate of interest for account type s and keeping in bank for 1 years.

**CASE 3: type ‘c’**

Enter s for savings and c for current type

>>c

Enter time and amount in Rs and paisa

>>1

>>20000.00

Customer is getting Rs 20800.00 is getting rate 4% interest rate and as additional total of 4% rate of interest for account type s and keeping in bank for 1 years.