**SYLLABUS FOR CAE-I UNIT I and few topics of UNIT II**

**Unit-I: Principles of Object Oriented Programming**

Differences between C and C++. A look at procedure Oriented programming, object oriented programming paradigm, basic concepts of OOP, Benefits of OOP, OO languages, A sample program, structure of C++ program. Introduction to OOPS: The origins of C++, What is Object Oriented Programming?, Some C++ fundamentals, Headers & Name Spaces, Introducing C++ Classes, Function overloading, Operator overloading, Inheritance, Constructors & Destructors, Function & Operator Overloading.

**Unit-II: Overloading**

Constructor functions, Localizing variables, Function overloading & Ambiguity

**PROGRAMS**

**1.** #include<iostream>

using namespace std;

class student

{

char Name[10];

int Age;

public:

void getdata()

{

cout<<"Name:";

cin>>Name;

cout<<"Age:";

cin>>Age;

}

void putdata()

{

cout<<"Name:\t"<<Name;

cout<<"\tAge:\t"<<Age;

}

};

int main()

{

student s1;

s1.getdata();

s1.putdata();

return(0);

}

**2.Largest number**

#include<iostream>

using namespace std;

class large

{

int a,b;

public:

void getdata()

{

cout<<"a:";

cin>>a;

cout<<"b:";

cin>>b;

}

void calculate()

{

if(a>b)

cout<<"a is max";

else

cout<<"b is max";

}

};

int main()

{

large m1;

m1.getdata();

m1.calculate();

return(0);

}

**3.function call**

#include<iostream>

using namespace std;

int add(int num1,int num2)

{

int sum=0;

sum=num1+num2;

return(sum);

}

int main()

{

int a,b,c;

cout<<"a:";

cin>>a;

cout<<"b:";

cin>>b;

c=add(a,b);

cout<<"a:"<<a<<endl;

cout<<"b:"<<b<<endl;

cout<<"c:"<<c<<endl;

return(0);

}

**4. All types of constructors(default,parameterized,copy)**

#include<iostream>

using namespace std;

class complex

{

int r,i;

public:

complex()

{

r=i=0;

}

complex(int a)

{

r=i=a;

}

complex(int a,int b)

{

r=a;

i=b;

}

complex(complex &x)

{

r=x.r;

i=x.i;

}

void show()

{

cout<<"r:\t"<<r<<endl;

cout<<"i:\t"<<i<<endl;

}

};

int main()

{

complex c1;

complex c2(12);

complex c3(10,15);

complex c4(c3);

c1.show();

c2.show();

c3.show();

c4.show();

return(0);

}

**SUB OF TWO ARRAYS**

#include<iostream>

using namespace std;

class marray

{

private:

int a1[2][2],a2[2][2],a3[2][2],r,c;

public:

void get();

void print();

void cal();

};

void marray::get()

{

cout<<"Enter First Array Elements\n";

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

{

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

{

cin>>a1[r][c];

}

}

cout<<"Enter Second Array Elements\n";

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

{

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

{

cin>>a2[r][c];

}

}

}

void marray::print()

{

cout<<"\nFirst Array\n";

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

{

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

{

cout<<" "<<a1[r][c];

}

cout<<"\n";

}

cout<<"\nSecond Array\n";

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

{

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

{

cout<<" "<<a2[r][c];

}

cout<<"\n";

}

}

void marray::cal()

{

cout<<"Subtraction Of Two Matrices\n";

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

{

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

{

a3[r][c]=a1[r][c]-a2[r][c];

}

}

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

{

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

{

cout<<" "<<a3[r][c];

}

cout<<"\n";

}

}

int main()

{

marray sub;

sub.get();

sub.print();

sub.cal();

}

**BINARY\_OPERATOR\_OVERLOADING (subtraction of two complex nos.)**

#include<iostream>

using namespace std;

class complex

{

private:

int real,img;

public:

complex()

{

real=0,img=0;

}

void input()

{

cout<<"\n\nEnter real and imaginary values";

cin>>real>>img;

}

void display()

{

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

}

complex operator -(complex c2)

{

complex temp;

temp.real=real-c2.real;

temp.img=img=img-c2.img;

return temp;

}

};

int main()

{

complex c1;

cout<<"\nEnter first complex";

c1.input();

c1.display();

complex c2;

c2.input();

c2.display();

complex result;

result=c1-c2;

result.display();

}

**SUM OF DIAGONAL\_ELEMENTS OF MATRIX**

#include<iostream>

using namespace std;

class marray

{

private:

int a[3][3],r,c,s=0;

public:

void get();

void print();

void cal();

};

void marray::get()

{

cout<<"Enter Array Elements:";

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

cin>>a[r][c];

}

}

}

void marray::print()

{

cout<<"\nMatrix\n";

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

cout<<" "<<a[r][c];

}

cout<<"\n";

}

}

void marray::cal()

{

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

if(r==c)

{

s=s+a[r][c];

}

}

}

cout<<"Sum of Diagonal Elements is "<<s;

}

int main()

{

marray sum;

sum.get();

sum.print();

sum.cal();

}

**MATRIX**

#include<iostream>

using namespace std;

class marray

{

private:

int a[2][2],r,c;

public:

void get();

void print();

};

void marray::get()

{

cout<<"Enter elements of matrix";

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

{

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

{

cin>>a[r][c];

}

}

}

void marray::print()

{

cout<<"Matrix:\n";

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

{

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

{

cout<<" "<<a[r][c];

}

cout<<"\n";

}

}

int main()

{

marray ob;

ob.get();

ob.print();

}

**SEARCHING OF ELEMENT IN ARRAY**

#include<iostream>

using namespace std;

class search

{

private:

int a[5],i,key;

public:

void getdata();

void print();

void find();

};

void search::getdata()

{

cout<<"Enter Array Elements";

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

{

cin>>a[i];

}

}

void search::print()

{

cout<<"Array Elements:";

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

{

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

}

}

void search::find()

{

int flag=0;

cout<<"\nEnter the Element to serach";

cin>>key;

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

{

if(a[i]==key)

{

flag=1;

break;

}

}

if (flag==1)

{

cout<<"\nElement found in the array";

}

else

{

cout<<"\nElement not found in the array";

}

}

int main()

{

search ob;

ob.getdata();

ob.print();

ob.find();

}

**ARRAY**

#include<iostream>

using namespace std;

class array

{

public:

void get();

void print();

};

void array::get()

{

int i,a[5];

cout<<"Enter array Elements";

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

{

cin>>a[i];

}

}

void array::print()

{

int i,a[5];

cout<<"Array Elements";

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

{

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

}

}

int main()

{

array ar;

ar.get();

ar.print();

}

**\*\*BI\_OP\_OVER**

#include<iostream>

using namespace std;

class box

{

private:

int v,l,b,h;

public:

box()

{

l=b=h=v=0;

}

void input()

{

cout<<"\nEnter the length , breadth and height of box";

cin>>l>>b>>h;

}

void display()

{

v=l\*b\*h;

cout<<"\nvolume is "<<v;

}

box operator +(box b2)

{

box temp;

temp.l=l+b2.l;

temp.b=b+b2.b;

temp.h=h+b2.h;

return(temp);

}

};

int main()

{

box b1;

b1.input();

b1.display();

box b2;

b2.input();

b2.display();

box r;

r=b1+b2;

r.display();

}**\*\***

**SUM OF ARRAY ELEMENTS**

#include<iostream>

using namespace std;

class array

{

private:

int i,a[5],s=0;

public:

void get();

void print();

void sum();

};

void array::get()

{

cout<<"Enter Array Elements\n";

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

{

cin>>a[i];

}

}

void array::print()

{

cout<<"\nArray Elements\n";

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

{

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

}

}

void array::sum()

{

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

{

s=s+a[i];

}

cout<<"\nSum Of Array Elements is "<<s;

}

int main()

{

array ob;

ob.get();

ob.print();

ob.sum();

}

**ADD OF MATRICES**

#include<iostream>

using namespace std;

class marray

{

private:

int a1[2][2],a2[2][2],a3[2][2],r,c;

public:

void get();

void print();

void cal();

};

void marray::get()

{

cout<<"Enter First Array elements";

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

{

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

{

cin>>a1[r][c];

}

}

cout<<"Enter Second Array elements";

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

{

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

{

cin>>a2[r][c];

}

}

}

void marray::print()

{

cout<<"\nFirst Array\n";

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

{

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

{

cout<<" "<<a1[r][c];

}

cout<<"\n";

}

cout<<"\nSecond Array\n";

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

{

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

{

cout<<" "<<a2[r][c];

}

cout<<"\n";

}

}

void marray::cal()

{

cout<<"\nAddition of two matrices\n";

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

{

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

{

a3[r][c]=a1[r][c]+a2[r][c];

}

}

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

{

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

{

cout<<" "<<a3[r][c];

}

cout<<"\n";

}

}

int main()

{

marray add;

add.get();

add.print();

add.cal();

}

**IDENTITY MATRIX**

#include<iostream>

using namespace std;

class identitymatrix

{

private:

int a[3][3],r,c,flag;

public:

void input();

void check();

void display();

};

void identitymatrix::input()

{

cout<<"\nEnter matrix elements";

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

cin>>a[r][c];

}

}

}

void identitymatrix::check()

{

flag=0;

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

if(r==c)

{

{

if(a[r][c]!=1)

{

flag=1;

break;

}

}

}

if(r!=c)

{

if(a[r][c]!=0)

{

flag=1;

break;

}

}

}

}

}

void identitymatrix::display()

{

if(flag==0)

{

cout<<"\nMatix is Identity Mtrix";

}

else

{

cout<<"\nMtrix is not identity matrix\n\n";

}

cout<<"\nMatrix\n\n";

for(r=0;r<3;r++)

{

for(c=0;c<3;c++)

{

cout<<" "<<a[r][c];

}

cout<<"\n";

}

}

int main()

{

identitymatrix ob;

ob.input();

ob.check();

ob.display();

}

**RESULT**

#include<iostream>

using namespace std;

class result

{

private:

int i,std[3],total=0,rollno;

float per;

public:

void get();

void print();

};

void result::get()

{

cout<<"Enter Roll No.";

cin>>rollno;

cout<<"Enter marks of Physics,Maths,Chemistry";

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

{

cin>>std[i];

}

}

void result::print()

{

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

{

total=total+std[i];

}

per=(total/3);

cout<<"Total Marks:"<<total<<"\n";

cout<<"Percentage:"<<per<<"%\n";

}

int main()

{

result r1,r2,r3;

r1.get();

r1.print();

r2.get();

r2.print();

r3.get();

r3.print();

}

**OPERATOR\_OVERLOADING**

#include<iostream>

using namespace std;

class distance

{

private:

int feet,inches;

public:

distance (int f,int i)

{

feet=f;

inches=i;

}

void displaydistance ()

{

cout<<"\nfeet="<<feet;

cout<<"\ninches="<<inches;

}

distance operator -()

{

feet=-feet;

inches=-inches;

return distance(feet,inches);

}

};

int main()

{

::distance d1(10,20);

-d1;

d1.displaydistance();

::distance d2(-30,10);

-d2;

d2.displaydistance();

return 0;

}

**\*\*MAXIMUM**

#include<iostream>

using namespace std;

class maximum

{

public:

void getdata();

void print();

void max();

};

void maximum::getdata()

{

int i,a[5];

cout<<"Enter Array Elements";

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

{

cin>>a[i];

}

}

void maximum::print()

{

int i,a[5];

cout<<"Array elements";

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

{

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

}

}

void maximum::max()

{

int m,a[5],i;

m=a[0];

for(i=1;i<5;i++)

{

if(a[i]>m)

{

m=a[i];

}

}

cout<<"\nMaximum element is:"<<m;

}

int main()

{

maximum ob;

ob.getdata();

ob.print();

ob.max();

}

**\*\***

**CHEF & SEQUENCES**

#include<iostream>

using namespace std;

int main()

{

int t,n,k,count,i=10,j;

int a[i];

cout<<"enter the no. of test cases:";

cin>>t;

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

{

cout<<"enter the no. of order:";

cin>>n;

cout<<"enter the no. of sequences:";

cin>>k;

cout<<"the order:";

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

{

cin>>a[i];

}

}

if(a[i]>=1)

{

count++;

}

if(k==n)

{

cout<<"yes"<<endl;

}

else

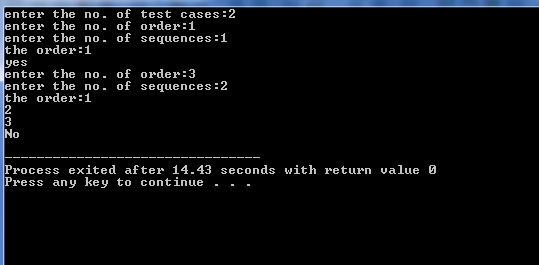
{

cout<<"No"<<endl;

}

return (0);

}



**ADDITION**

#include<iostream>

using namespace std;

class addition

{

public:

add(int a,int b)

{

int sum;

sum=a+b;

cout<<sum<<"\n";

}

add(float c,float d)

{

float addi;

addi=c+d;

cout<<addi;

}

};

int main()

{

addition a;

a.add(11,22);

a.add(11.1f,22.0f);

return (0);

}

**ADDITION**

#include <iostream>

using namespace std;

int sum(int, int);

float sum(float, float);

float sum(int, float);

int main()

{

int num1, num2,x;

float num3, num4,y;

cout<<"Enter two integer numbers: ";

cin>>num1>>num2;

cout<<"Result: "<<sum(num1, num2)<< endl;

cout<<"Enter two float numbers: ";

cin>>num3>>num4;

cout<<"Result: " <<sum(num3, num4)<< endl;

cout<<"Enter one int and one float number: ";

cin>>x>>y;

cout<<"Result: " <<sum(x, y)<< endl;

return 0;

}

int sum(int a, int b)

{

return a+b;

}

float sum(float a, float b)

{

return a+b;

}

float sum(int a, float b)

{ return a+b;}

**SHAPE**

#include<iostream>

using namespace std;

class shape

{

public:

vol(float a,float b,float c)

{

float vol;

vol=a\*b\*c;

cout<<vol<<"\n";

}

vol(int d)

{

int volcube;

volcube=d\*d\*d;

cout<<volcube<<"\n";

}

vol(float r,float h)

{

float volcyl;

volcyl=3.14\*r\*r\*h;

cout<<volcyl;

}

};

int main()

{

shape a;

a.vol(1.0f,2.0f,3.0f);

a.vol(2);

a.vol(2.0f,3.0f);

return (0);

}

**RECTANGLE**

#include <iostream>

using namespace std;

class rectangle

{

int l,b;

public:

void getvalue()

{

cout<<"enter value of length and breadth";

cin>>l>>b;

}

rectangle()

{

l=5;

b=2;

}

float calculatearea()

{

cout<<"area of rectangle is"<<endl;

return(l\*b);

}

};

int main()

{

rectangle r1,r2;

r1.getvalue();

float area=r1.calculatearea();

cout<<"area of rectangle"<<area<<endl;

float area1=r2.calculatearea();

cout<<"area of rectangle without calling getvalue"<<area1<<endl;

return(0);

}

**\*\*EMPLOYEE**

#include <iostream>

#include<string>

#include<stdio.h>

using namespace std;

class employee

{

char name[30];

float salary;

int emp\_id;

public:

employee()

{

emp\_id=0;

salary=0;

strcpy(name,null);

}

employee(int e,char[30],float=s)

{

emp\_id=e;

strcpy(name,n);

salary=s;

}

employee(int e1)

{

emp\_id=e1;

}

void display()

{

cout<<"employee details are:";

cout<<"id"<<emp\_id<<endl;

cout<<"name"<<name<<endl;

cout<<"salary"<<salary<<endl;

}

};

int main()

{

employee obj1;

employee obj2(210);

employee obj3(110,"nikhil",2000);

obj1.display();

obj2.display();

obj3.display();

}

}

**\*\***

**FACTORIAL**

#include<iostream>

using namespace std;

class Factorial

{

private:

int n,i,fact;

public:

void getdata()

{

cout<<"enter the number:"<<endl;

cin>>n;

}

void cal()

{

fact=1;

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

{

fact=fact\*i;

}

}

void display()

{

cout<<"factorial of a number:"<<fact;

}

};

int main()

{

Factorial f;

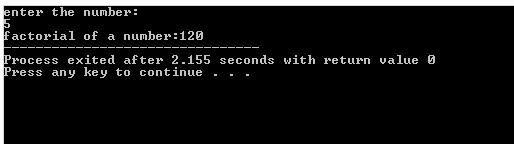
f.getdata();

f.cal();

f.display();

return (0);

}



**BANK USING CLASS**

// for 1 member

#include<iostream>

using namespace std;

class Bank\_Account

{

public:

int an1,deposite,withdrawl,balance,amount;

char nd1[20],ta1[30];

void getdata()

{

cout<<"enter the name of the depositor:"<<endl;

cin>>nd1;

cout<<"enter the account number:"<<endl;

cin>>an1;

cout<<"enter the type of account:"<<endl;

cin>>ta1;

cout<<"enter the deposite amount in the account:"<<endl;

cin>>deposite;

cout<<"enter the initial amount:"<<endl;

cin>>amount;

}

void bank\_info()

{

balance=deposite+amount;

if(balance>=amount)

{

withdrawl=balance-amount;

}

else

{

cout<<"the balance is zero:";

}

}

void display\_info()

{

cout<<"Depositor's name : "<<nd1<<endl;

cout<<"account no. : "<<an1<<endl;

cout<<"account type : "<<ta1<<endl;

cout<<"balance amount : "<<deposite<<endl;

cout<<"initial amount : "<<balance<<endl;

cout<<"withdrawl amount : "<<withdrawl<<endl;

}

};

int main()

{

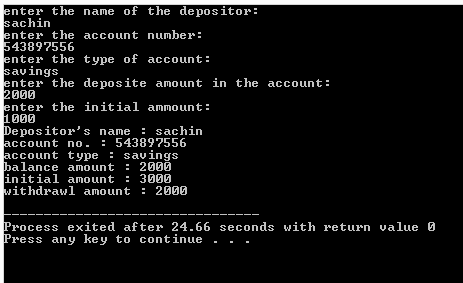
Bank\_Account b;

b.getdata();

b.bank\_info();

b.display\_info();

}



**BANK USING CLASS**

//for 10 members

#include<iostream>

using namespace std;

class Bank\_Account

{

public:

int an1,deposite,withdrawl,balance,amount;

char nd1[20],ta1[30];

void getdata()

{

cout<<"enter the name of the depositor:"<<endl;

cin>>nd1;

cout<<"enter the account number:"<<endl;

cin>>an1;

cout<<"enter the type of account:"<<endl;

cin>>ta1;

cout<<"enter the deposite amount in the account:"<<endl;

cin>>deposite;

cout<<"enter the initial ammount:"<<endl;

cin>>amount;

}

void bank\_info()

{

balance=deposite+amount;

if(balance>=amount)

{

withdrawl=balance-amount;

}

else

{

cout<<"the balance is zero:";

}

}

void display\_info()

{

cout<<"Depositor's name : "<<nd1<<endl;

cout<<"account no. : "<<an1<<endl;

cout<<"account type : "<<ta1<<endl;

cout<<"balance amount : "<<deposite<<endl;

cout<<"initial amount : "<<balance<<endl;

cout<<"withdrawl amount : "<<withdrawl<<endl;

}

};

int main()

{

int i;

Bank\_Account b;

for(i=1;i<10;i++)

{

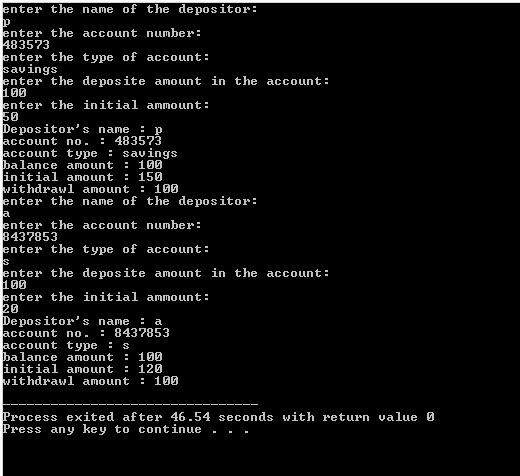
b.getdata();

b.bank\_info();

b.display\_info();

}

}



**TRUTH & DARE**

#include<iostream>  
using namespace std;  
int main()  
{  
    int i,tr[10],dr[10],ts[10],ds[10],nr,ns,ps,pr,j,flag=0;  
        cout<<"Enter the number of truth task ram can perform:";  
        cin>>nr;  
        cout<<"Enter the integers:";  
        for(i=0;i<nr;i++)  
        {  
            cin>>tr[i];  
        }  
        cout<<"Enter the number of dare task ram can perform:";  
        cin>>pr;  
        cout<<"Enter the integer:";  
        for(i=0;i<pr;i++)  
        {  
            cin>>dr[i];  
        }  
        cout<<"Enter the number of truth task shyaam can ask:";  
        cin>>ns;  
        cout<<"Enter the integer:";  
        for(i=0;i<ns;i++)  
        {  
            cin>>ts[i];  
        }  
        cout<<"Enter the number of dare task shyaam can ask:";  
        cin>>ps;  
        cout<<"Enter the integer:";  
        for(i=0;i<ps;i++)  
        {  
            cin>>ds[i];  
        }  
      
        for(i=0;i<ns;i++)  
        {  
            for(j=0;j<nr;j++)  
            {

if(ts[ns]==tr[nr])  
                flag++;  
            }  
              
        }  
        for(i=0;i<ps;i++)  
        {  
            for(j=0;j<pr;j++)  
            {

if(ds[ps]==dr[pr])  
                flag++;  
            }  
        }  
          
        if(flag==0)  
        {  
            cout<<"ram wins";  
        }  
        else  
        {  
            cout<<"shyaam wins";  
        }  
return(0);  
}

**OR**

#include<iostream>

using namespace std;

int main()

{

int tr,dr,ts,ds,tr1[20],dr1[20],ts1[20],ds1[20],i;

cout<<" Enter the no of tasks ram can perform"<<endl;

cin>>tr;

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

{

cout<<"Enter the task no ram can perform"<<endl;

cin>>tr1[i];

}

cout<<"The nos of task are "<<endl;

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

{

cout<<tr1[i];

}

cout<<" Enter the no of dare ram can perform"<<endl;

cin>>dr;

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

{

cout<<"Enter the dare nos ram can perform"<<endl;

cin>>dr1[i];

}

cout<<"The nos of dare are "<<endl;

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

{

cout<<dr1[i];

}

cout<<" Enter the no of tasks shyam can perform"<<endl;

cin>>ts;

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

{

cout<<"Enter the task no shaym can perform"<<endl;

cin>>ts1[i];

}

cout<<"The nos of task are "<<endl;

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

{

cout<<ts1[i];

}

cout<<" Enter the no of dare shyam can perform"<<endl;

cin>>ds;

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

{

cout<<"Enter the taskdare no shyam can perform"<<endl;

cin>>ds1[i];

}

cout<<"The nos of dare are "<<endl;

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

{

cout<<ds1[i];

}

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

{

if(tr1[i]==ts1[i])

{

cout<<"Ram wins";

}

else

{

cout<<"shyam wins";

}}

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

{

if(dr1[i]==ds1[i])

{

cout<<"Ram wins";

}

else

{

cout<<"shyam wins";

}

}

}

**BANK**

#include<iostream>

using namespace std;

class bank

{

private:

int account\_Number;

char depositor\_Name[50],toa;

float balance;

public:

float dep,with;

void getdata()

{

cout<<"enter the name acc no ,toa,balence ";

cin>>depositor\_Name>>account\_Number>>toa>>balance;

}

void dep\_amt()

{

cout<<"ehter the amount to b edeposited";

cin>>dep;

balance=balance+dep;

}

void wid()

{

cout<<"the baalnec is;"<<balance;

cout<<"Enter he amount to be withdrawed";

cin>>with;

balance=balance-with;

}

void display()

{

cout<<"name"<< depositor\_Name;

cout<<"<balance"<<balance;

}

};

int main()

{

bank e;

e.getdata();

e.dep\_amt();

e.wid();

e.display();

return (0);

}

**LARGEST**

#include<iostream>

using namespace std;

int main()

{

int x,y,z;

cout<<"Enter 3 interger nos: ";

cin>>x>>y>>z;

if(x>y&&x>z)

{

cout<<"The largest no is:"<<x;

}

else if(y>z&&y>x)

{

cout<<"The greatest no is "<<y;

}

else

{

cout<<"The greatest no is "<<z;

}

return 0;

}

**ODD & EVEN**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int n;

cout<<"Enter the integer no to be checked";

cin>>n;

if(n%2==0)

{

cout<<"The no is even";

}

else

{

cout<<"The no is odd";

}

return 0;

}

**CALCULATOR**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int x,y,a,s,m;

float d;

cout<<"Enter two integer nos";

cin>>x>>y;

a=x+y;

s=x-y;

m=x\*y;

d=x/y;

cout<<"Addition:"<<a;

cout<<"sub:"<<s;

cout<<"mul:"<<m;

cout<<"div:"<<d;

return 0;

}

**SIZE OF DATA TYPES**

#include<iostream>

using namespace std;

int main()

{

cout<<"Size of char :\t"<<sizeof(char)<<endl;

cout<<"Size of int:\t "<<sizeof(int)<<endl;

cout<<"Size of float:\t"<<sizeof(float)<<endl;

cout<<"Size of long int:\t"<<sizeof(long int)<<endl;

cout<<"Size of signed char:\t"<<sizeof( signed char)<<endl;

cout<<"Size of unsigned char:\t"<<sizeof( unsigned char)<<endl;

cout<<"Size of short int: \t "<<sizeof(short int)<<endl;

return 0;

}

**LEAP YEAR**

#include<iostream>

#include<math.h>

using namespace std;

int main()

{

int x;

cout<<"Enter the year to be checked\t";

cin>>x;

if(x%100==0 && x%4==0)

{

cout<<"The year is century as well as leap year";

}

else

{

cout<<" The year is not a leap year";

}

return 0;

}

**VOWEL**

#include<iostream>

#include<string.h>

using namespace std;

int main()

{

char x;

cout<<"Enter a character: \t";

cin>>x;

if(x=='a'|| x=='e'||x=='i'||x=='o'||x=='u')

cout<<"the char is vowel";

else

cout<<"The char is a consonant";

return 0;

}

**SWAP**

#include<iostream>

using namespace std;

class temp

{

public:

int a,b,temp;

void getdata()

{

cout<<"enter a:"<<endl;

cin>>a;

cout<<"enter b:"<<endl;

cin>>b;

}

void swap()

{

temp=a;

a=b;

b=temp;

}

void display()

{

cout<<"after swapping a is:"<<a<<endl;

cout<<"after swapping b is:"<<b;

}

};

int main()

{

temp t;

t.getdata();

t.swap();

t.display();

return(0);

}

**FARENHEIT TO CELSIUS**

#include<iostream>

using namespace std;

int main()

{

int t1,t2;

cout<<"enter temperature t1 in fahrenheit:"<<endl;

cin>>t1;

t2=(t1-32)\*5/9;

cout<<"temperature t2 in celcius:"<<t2;

return (0);

}

**GAS AGENCY SYSTEM**

#include<iostream>

#include<string.h>

using namespace std;

class Employee

{

public:

char f\_name[20],l\_name[20],a[20]="Manager",b[20]="Salesman",c[20]="Technician";

int sal;

void getFirstname()

{

cout<<"Enter the first name of employee: ";

cin>>f\_name;

}

void getLastname()

{

cout<<"Enter the last name of employee: ";

cin>>l\_name;

}

};

class ContractEmployee:public Employee

{

public:

char dep[20],des[20];

void getcDepartment(void)

{

cout<<"Enter the department: ";

cin>>dep;

}

void getcDesignation(void)

{

cout<<"Enter the designation assigned: ";

cin>>des;

}

void cdisplay(void)

{

cout<<"The Information of the Employee is:-----"<<endl;

cout<<"Full Name: "<<f\_name<<l\_name<<endl;

cout<<"Department: "<<dep<<endl;

cout<<"Designation: "<<des<<endl;

cout<<"Salary: "<<sal;

}

void getcSalary(void)

{

if(strcmp(des,a)==0)

{

sal=40000;

}else if(strcmp(des,b)==0)

{

sal=15000;

}else if(strcmp(des,c)==0)

{

sal=25000;

}else{

cout<<"NO Information Found!!!!";

}

}

};

class RegularEmployee:public Employee

{

public:

char dep[20],des[20];

void getrDepartment(void)

{

cout<<"Enter the department: ";

cin>>dep;

}

void getrDesignation(void)

{

cout<<"Enter the designation assigned: ";

cin>>des;

}

void rdisplay(void)

{

cout<<"The Information of the Employee is:-----"<<endl;

cout<<"Full Name: "<<f\_name<<l\_name<<endl;

cout<<"Department: "<<dep<<endl;

cout<<"Designation: "<<des<<endl;

cout<<"Salary: "<<sal;

}

void getrSalary(void)

{

if(strcmp(des,a)==0)

{

sal=50000;

}else if(strcmp(des,b)==0)

{

sal=20000;

}else if(strcmp(des,c)==0)

{

sal=30000;

}else{

cout<<"NO Information Found!!!!";

}

}

};

int main()

{

int i,ch;

cout<<"Welcome to the EMPLOYEEE MANAGEMENT SYSTEM"<<endl;

cout<<"1)Regular Employee"<<endl;

cout<<"2)Contract Employee"<<endl;

cout<<"Enter your choice: ";

cin>>ch;

if(ch==1)

{

RegularEmployee r;

r.getFirstname();

r.getLastname();

r.getrDepartment();

r.getrDesignation();

r.getrSalary();

r.rdisplay();

}else if(ch==2)

{

ContractEmployee c;

c.getFirstname();

c.getLastname();

c.getcDepartment();

c.getcDesignation();

c.getcSalary();

c.cdisplay();

}else{

cout<<"WRONG ATTEMPT!!!!";

}

}