

UNIT 2(Program Solution)

1. Declare a C++ structure (Program) to contain the following piece of information about cars on used car lot. [PU:2013 spring]

- i. Manufacturer of the car
- ii. Model name of the car
- iii. The asking price of car
- iv. The number of miles on odometer

```
#include<iostream>
using namespace std;
struct car
{
char name[50];
char model[50];
long int price;
int miles;
}
int main()
{
    car c;
    cout<<"Enter manufacturer of car"<<endl;
    cin.getline(c.name,50);
    cout<<"Enter the model name of car"<<endl;
    cin.getline(c.model,50);
    cout<<"Enter the price of car"<<endl;
    cin>>c.price;
    cout<<"Enter number of miles on odometer"<<endl;
    cin>>c.miles;
    cout<<"Details of car:"<<endl;
    cout<<"Manufacturer name:"<<c.name<<endl;
    cout<<"Model name:"<<c.model<<endl;
    cout<<"Price : "<<c.price<<endl;
    cout<<"Number of miles on odometer:"<<c.miles<<endl;
    return 0;
}
```

2. Create a class called Employee with three data members (empno , name, address), a function called readdata() to take in the details of the employee from the user, and a function called displaydata() to display the details of the employee. In main, create two objects of the class Employee and for each object call the readdata() and the displaydata() functions. **[PU:2005 fall]**

```
#include<iostream>
using namespace std;
class Employee
{
private:
int empno;
char name[20];
char address[20];
public:
void readdata();
void displaydata();
};
void Employee::readdata()
{
cout<<"Enter the Employee number"<<endl;
cin>>empno;
cout<<"Enter the Employee name"<<endl;
cin>>name;
cout<<"Enter the address"<<endl;
cin>>address;
}
void Employee::displaydata()
{
cout<<"Employee number="<<empno<<endl;
cout<<"Employee name="<<name<<endl;
cout<<"Employee address="<<address<<endl;
}
int main()
{
Employee e1,e2;
cout<<"Enter the information of first employee"<<endl;
e1.readdata();
cout<<"Enter the information of second employee"<<endl;
e2.readdata();
cout<<"Information of first employee is"<<endl;
e1.displaydata();
cout<<"Information of second employee is"<<endl;
e2.displaydata();
return 0;
}
```

3. Create a class called Student with three data members(stdnt_name[20],faculty[20],roll_no), a function called readdata() to take the details of the students from the user and a function called displaydata() to display the details the of the students. In main, create two objects of the class student and for each object call both of the functions. **[PU:2010 fall]**

```
#include<iostream>
using namespace std;
class Student
{
private:
char stdnt_name[20];
char faculty[20];
int roll_no;
public:
void readdata();
void displaydata();
};
void Student::readdata()
{
cout<<"Enter student name"<<endl;
cin>>stdnt_name;
cout<<"Enter student faculty"<<endl;
cin>>faculty;
cout<<"Enter student roll number"<<endl;
cin>>roll_no;
}
void Student::displaydata()
{
cout<<"Name="<<stdnt_name<<endl;
cout<<"Roll no="<<roll_no<<endl;
cout<<"Faculty="<<faculty<<endl;
}
int main()
{
Student st1,st2;
cout<<"Enter the information of first student"<<endl;
st1.readdata();
cout<<"Enter the information of second student"<<endl;
st2.readdata();
cout<<"Informaton of first student is:"<<endl;
st1.displaydata();
cout<<"Information of second student is:"<<endl;
st2.displaydata();
return 0;
}
```

4. Modify the Que.no 3 for 20 students using array of object.

```
#include<iostream>
using namespace std;
class Student
{
private:
char stdnt_name[20];
char faculty[20];
int roll_no;
public:
void readdata();
void displaydata();
};
void Student::readdata()
{
cout<<"Enter student name"<<endl;
cin>>stdnt_name;
cout<<"Enter student faculty"<<endl;
cin>>faculty;
cout<<"Enter student roll number"<<endl;
cin>>roll_no;
}
void Student::displaydata()
{
cout<<"Name="<<stdnt_name<<endl;
cout<<"Roll no="<<roll_no<<endl;
cout<<"Faculty="<<faculty<<endl;
}
int main()
{
Student st[20];
int i;
for(i=0;i<20;i++)
{
cout<<"Enter the information of student:"<<i+1<<endl;
st[i].readdata();
}
for(i=0;i<20;i++)
{
cout<<"Information of student:"<<i+1<<endl;
st[i].displaydata();
}
return 0;
}
```

5. WAP to perform the addition of time in hours, minutes and seconds by passing object as an argument.

```
#include<iostream>
using namespace std;
class Time
{
private:
int hours,minutes,seconds;
public:
void gettime(int h,int m,int s)
{
hours=h;
minutes=m;
seconds=s;
}
void display()
{
cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
}
void sum(Time t1,Time t2)
{
seconds=t1.seconds+t2.seconds;
minutes=seconds/60;
seconds=seconds%60;
minutes=minutes+t1.minutes+t2.minutes;
hours=minutes/60;
minutes=minutes%60;
hours=hours+t1.hours+t2.hours;
}
};
int main()
{
Time t1,t2,t3;
t1.gettime(2,45,35);
t2.gettime(3,30,40);
t3.sum(t1,t2);
cout<<"First time=";
t1.display();
cout<<"Second time=";
t2.display();
cout<<"sum of two times=";
t3.display();
return 0;
}
```

6. WAP to perform the addition of time by passing object as an argument and returning object.

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

class Time
{
private:
int hours,minutes,seconds;
public:
void gettime(int h,int m,int s)
{
hours=h;
minutes=m;
seconds=s;
}
void display()
{
cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
}
Time sum(Time t1,Time t2)
{
Time temp;
temp.seconds=t1.seconds+t2.seconds;
temp.minutes=temp.seconds/60;
temp.seconds=temp.seconds%60;
temp.minutes=temp.minutes+t1.minutes+t2.minutes;
temp.hours=temp.minutes/60;
temp.minutes=temp.minutes%60;
temp.hours=temp.hours+t1.hours+t2.hours;
return temp;
}
};

int main()
{
Time t1,t2,t3,result;
t1.gettime(2,45,35);
t2.gettime(3,30,40);
result=t3.sum(t1,t2);
cout<<"First time=";
t1.display();
cout<<"Second time=";
t2.display();
cout<<"Sum of two times=";
result.display();
return 0;
}
```

7. WAP to create two time objects with data members hours, minutes and seconds a function call by one object passing second object as function argument and return third object adding two objects. *Hint: t3=t1.adddistance(t2);*

```
#include<iostream>
using namespace std;
class Time
{
private:
int hours,minutes,seconds;
public:
void gettime(int h,int m,int s)
{
hours=h;
minutes=m;
seconds=s;
}
void display()
{
cout<<hours<<":"<<minutes<<":"<<seconds<<endl;
}
Time sum(Time t2)
{
Time temp;
temp.seconds=seconds+t2.seconds;
temp.minutes=temp.seconds/60;
temp.seconds=temp.seconds%60;
temp.minutes=temp.minutes+minutes+t2.minutes;
temp.hours=temp.minutes/60;
temp.minutes=temp.minutes%60;
temp.hours=temp.hours+hours+t2.hours;
return temp;
}
};
int main()
{
Time t1,t2,t3;
t1.gettime(2,45,35);
t2.gettime(3,30,40);
t3=t1.sum(t2);
cout<<"First time=";
t1.display();
cout<<"Second time=";
t2.display();
cout<<"Sum of two times=";
t3.display();
return 0;
}
```

8. WAP to create two distance objects with data members feet, inches and a function call by one object passing second object as function argument and return third object adding two objects. *Hint: d3=d1.adddistance(d2);*

```
#include<iostream>
using namespace std;
class Distance
{
private:
int feet;
int inches;
public:
void getdata();
Distance add(Distance);
void display();
};
void Distance::getdata()
{
cout<<"Enter feet:"<<endl;
cin>>feet;
cout<<"Enter inches:"<<endl;
cin>>inches;
}
void Distance::display()
{
cout<<feet<<"feet and"<<inches<<"inches"<<endl;
}
Distance Distance::add( Distance d2)
{
Distance sum;
sum.inches=inches+d2.inches;
sum.feet=sum.inches/12;
sum.inches=sum.inches%12;
sum.feet=sum.feet+feet+d2.feet;
return (sum);
}
int main()
{
Distance d1,d2,d3;
cout<<"Enter first Distance"<<endl;
d1.getdata();
cout<<"Enter second Distance"<<endl;
d2.getdata();
d3=d1.add(d2);
cout<<"The sum of two distances=" ;
d3.display();
return 0;}
```


9. Create a class called Rational having data members nume and deno and using friend function find which one is greater.

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

class Rational
{
private:
int nume,deno;
public:
void getdata()
{
cout<<"Enter the value of numerator"<<endl;
cin>>nume;
cout<<"Enter the value of denominator"<<endl;
cin>>deno;
}
friend void max(Rational r);
};

void max(Rational r)
{
if(r.nume>r.deno)
{
cout<<"Maximum value="<<r.nume<<endl;
}
else
{
cout<<"Maximum value="<<r.deno<<endl;
}
}

int main()
{
Rational r;
r.getdata();
max(r);
return 0;
}
```

10. WAP to add the private data of three different classes using friend function.

```
#include<iostream>
using namespace std;
class B;
class C;
class A
{
private:
int x;
public:
void setdata(int num)
{
x=num;
}
friend void sum(A,B,C);
};

class B
{
private:
int y;
public:
void setdata(int num)
{
y=num;
}
friend void sum(A,B,C);
};

class C
{
private:
int z;
public:
void setdata(int num)
{
z=num;
}
friend void sum(A,B,C);
};

void sum(A m,B n,C o)
{
cout<<"sum="<<(m.x+n.y+o.z)<<endl;
}
```

```

int main()
{
A p;
B q;
C r;
p.setdata(5);
q.setdata(10);
r.setdata(15);
sum(p,q,r);
return 0;
}

```

11. Write a program to find the largest of four integers .your program should have three classes and each classes have one integer number.[PU:2014 spring]

```

#include<iostream>
using namespace std;
class Y;
class Z;

class X
{
private:
int a;
public:
void getdata()
{
cout<<"Enter first number"<<endl;
cin>>a;
}
friend void max(X,Y,Z);
};

class Y
{
private:
int b;
public:
void getdata()
{
cout<<"Enter second number"<<endl;
cin>>b;
}
friend void max(X,Y,Z);
};

```

```

class Z
{
private:
int c;

public:

void getdata()
{
cout<<"Enter third number"<<endl;
cin>>c;
}
friend void max(X,Y,Z);
};

```

```

void max(X m,Y n,Z o)
{
int d;
cout<<"Enter fourth number"<<endl;
cin>>d;
if(m.a>n.b&& m.a>o.c&& m.a>d)
{
cout<<"Largest number="<<m.a<<endl;
}
else if(n.b>o.c&& n.b>d)
{
cout<<"Largest number="<<n.b<<endl;
}
else if(o.c>d)
{
cout<<"Largest number="<<o.c<<endl;
}
else
{
cout<<"Largest number="<<d<<endl;
}
}
}

```

```

int main()
{
X p;
Y q;
Z r;
p.getdata();
q.getdata();
r.getdata();
max(p,q,r);
return 0;
}

```

12. WAP to swap the contents of two variables of 2 different classes using friend function.

```

#include <iostream>
using namespace std;
class ABC;
class XYZ
{
private:
int x;
public:
void getdata ()
{
cout<<"Enter Value of class XYZ"<<endl;
cin>>x;
}
void display()
{
cout<<"value1="<<x<<endl;
}
friend void swap (XYZ &,ABC &);
};
class ABC
{
private:
int y ;
public:
void getdata ()
{
cout<<"Enter the value of class ABC"<<endl;
cin>>y;
}
}

```

```

void display()
{
cout<<"value2="<<y<<endl;
}
friend void swap(XYZ &,ABC &);
};

void swap (XYZ &m, ABC &n)
{
int temp;
temp=m.x;
m.x=n.y;
n.y=temp;
}

int main( )
{
XYZ p;
ABC q;
p.getdata() ;
q.getdata() ;
cout<<"Value before swapping"<<endl;
p.display();
q.display();
swap(p,q);
cout<<"Value after swapping"<<endl;
p.display();
q.display();
return 0;
}

```

13. WAP to add two complex numbers of two different classes using friend function.

```

#include<iostream>
using namespace std;
class complex2;
class complex1
{
private:
int real,imag;

```

```

public:
void getdata()
{
cout<<"Enter real and imaginary part"<<endl;
cin>>real>>imag;
}
void display()
{
cout<<real<<"+"<<imag<<"i"<<endl;
}
friend void addcomplex(complex1,complex2);
};

class complex2
{
private:
int real,imag;
public:
void getdata()
{
cout<<"Enter real and imaginary part"<<endl;
cin>>real>>imag;
}
void display()
{
cout<<real<<"+"<<imag<<"i"<<endl;
}
friend void addcomplex(complex1,complex2);
};

void addcomplex(complex1 c1,complex2 c2)
{
int real,imag;
real=c1.real+c2.real;
imag=c1.imag+c2.imag;
cout<<"sum of complex number="<<real<<"+"<<imag<<"i"<<endl;
}

```

```

int main()
{
    complex1 c1;
    complex2 c2;
    c1.getdata();
    c2.getdata();
    cout<<"first complex number"<<endl;
    c1.display();
    cout<<"second complex number"<<endl;
    c2.display();
    addcomplex(c1,c2);
    return 0;
}

```

14. WAP to add complex numbers of two different classes using friend class.

```

#include<iostream>
using namespace std;
class complex2;
class complex1
{
    private:
    int real,imag;
    public:
    void getdata()
    {
        cout<<"Enter real and imaginary part"<<endl;
        cin>>real>>imag;
    }
    void display()
    {
        cout<<real<<"+"<<imag<<endl;
    }
    friend class complex2;
};
class complex2
{
    private:
    int real,imag;

```



```

public:
void getdata()
{
cout<<"Enter real and imaginary part"<<endl;
cin>>real>>imag;
}
void display()
{
cout<<real<<"+"<<imag<<endl;
}
void addcomplex(complex1);
};
void complex2::addcomplex(complex1 c1)
{
real=c1.real+real;
imag=c1.imag+imag;
cout<<"sum of complex number="<<real<<"+"<<imag<<endl;
}
int main()
{
complex1 c1;
complex2 c2;
c1.getdata();
c2.getdata();
cout<<"first complex number"<<endl;
c1.display();
cout<<"second complex number"<<endl;
c2.display();
cout<<"sum="<<endl;
c2.addcomplex(c1);
return 0;
}

```

15. Using class write a program that receives inputs principle amount, time and rate. Keeping rate 8% as the default argument, calculate simple interest for three customers.**[PU:2019 fall]**

```

#include<iostream>
using namespace std;
class customer
{
private:
float principle,rate,si;
int time;

```

```

public:
void setdata(float p,int t,float r=8);
void display();
};
void customer::setdata(float p,int t,float r)
{
principle=p;
time=t;
rate=r;
}
void customer::display()
{
si=(principle*time*rate)/100;
cout<<"Simple Interest="<<si<<endl;
}
int main()
{
float p;
int t,i;
customer c[3];
for(i=0;i<3;i++)
{
cout<<"Enter principle and time for customer"<<i+1<<endl;
cin>>p>>t;
c[i].setdata(p,t);
c[i].display();
}
return 0;
};

```

16. Create a new class named City that will have two member variables CityName (char[20]),and DistFromKtm (float).Add member functions to set and retrieve the CityName and DistanceFromKtm separately. Add new member function AddDistance that takes two arguments of class City and returns the sum of DistFromKtm of two arguments. In the main function, Initialize three city objects .Set the first and second City to be Pokhara and Dhangadi. Display the sum ofDistFromKtm of Pokhara and Dhangadi calling AddDistance function of third City object. **[PU: 2010 Spring]**

```

#include<iostream>
#include<string.h>
using namespace std;

```

```

class City
{
private:
char CityName[20];
float DistFromKtm;
public:
void setdata(char cname[],float d)
{
strcpy(CityName,cname);
DistFromKtm=d;
}
void display()
{
cout<<"City name="<<CityName<<endl;
cout<<"Distance from ktm="<<DistFromKtm<<endl;
}
float AddDistance(City c1,City c2)
{
return (c1.DistFromKtm+c2.DistFromKtm);
}
};
int main()
{
City c1,c2,c3;
c1.setdata("Pokhara",250);
c2.setdata("Dhangadi",150);
cout<<"Information of first city"<<endl;
c1.display();
cout<<"Information of second city"<<endl;
c2.display();
cout<<"sum of DistFromKtm of Pokhara and Dhangadi="<<c3.AddDistance(c1,c2);
return 0;
}

```

17. Create a class called Volume that uses three Variables (length, width, height) of type distance (feet and inches) to model the volume of a room. Read the three dimensions of the room and calculate the volume it represent, and print out the result .The volume should be in (feet³) form ie. you will have to convert each dimension into the feet and fraction of foot. For instance , the length 12 feet 6 inches will be 12.5 ft)

[PU: 2009 spring]

```

#include<iostream>
using namespace std;

```

```

class volume
{
private:
int lf,wf,hf,li,wi,hi;
float vol,length,width,height;
public:
void getdata();
void convertdim();
void display();
};

void volume::getdata()
{
cout<<"Enter the length of room in feet and inches"<<endl;
cin>>lf>>li;
cout<<"Enter the width of room in feet and inches"<<endl;
cin>>wf>>wi;
cout<<"Enter the Height of room in feet and inches"<<endl;
cin>>hf>>hi;
}

void volume::convertdim()
{
length=lf+(float)li/12;
width=wf+(float)wi/12;
height=hf+(float)hi/12;
cout<<"The length of room ="<<length<<endl;
cout<<"The width of room ="<<width<<endl;
cout<<"The Height of room ="<<height<<endl;
}

void volume::display()
{
vol=length*width*height;
cout<<"Volume of Room ="<<vol<<endl;
}

int main()
{
volume v;
v.getdata();
v.convertdim();
v.display();
return 0;
}

```

18. WAP in C++ to calculate simple interest from given principal, time and rate. Set the rate to 15 % as default argument when rate is not provided.

```
#include<iostream>
using namespace std;
float calculate(float principle,int time,float rate=15);
int main()
{
    float p,r;
    int t;
    char ch;
    cout<<"Enter principle and time"<<endl;
    cin>>p>>t;
    cout<<"Do you want to enter rate?"<<endl;
    cin>>ch;
    if(ch=='Y' | | ch=='y')
    {
        cout<<"Enter rate"<<endl;
        cin>>r;
        cout<<"Interest="<<calculate(p,t,r)<<endl;
    }
    else
    {
        cout<<"Interest="<<calculate(p,t)<<endl;
    }
    return 0;
}
float calculate(float principle,int time,float rate)
{
    return(principle*time*rate)/100.0;
}
```

19. Create a class comparison with data members x and y and max and member function getdata() to read the value of x, y, largest() to find greater of two and print() to the greater number.

```
#include<iostream>
using namespace std;
class Comparison
{
private:
int x,y,max;
public:
void getdata()
{
cout<<"Enter two numbers"<<endl;
cin>>x>>y;
}
int largest()
{
max=(x>y)?x:y;
return max;
}
void print()
{
cout<<"Greater number="<<max<<endl;
}
};
int main()
{
Comparison c1;
c1.getdata();
c1.largest();
c1.print();
return 0;
}
```

20. Create a class student with six data members (roll no, name, marks in English,math, science and total).Write a program init() to initializes necessary data members calctotal() and display().Create a program for one student (i.e one object only necessary)

```
#include<iostream>
using namespace std;
class Student
{
private:
char name[20];
int roll;
float me,mm,ms,total;
public:
void init()
{
cout<<"Enter roll number"<<endl;
cin>>roll;
cout<<"Enter name"<<endl;
cin>>name;
cout<<"Enter Marks in English,Math and science"<<endl;
cin>>me>>mm>>ms;
}
void calctotal()
{
total=me+mm+ms;
}
void display()
{
cout<<"Name:"<<name<<endl;
cout<<"RollNo:"<<roll<<endl;
cout<<"Marks in English:"<<me<<endl;
cout<<"Marks in Math:"<<mm<<endl;
cout<<"Marks in science:"<<ms<<endl;
cout<<"Total:"<<total<<endl;
}
};
int main()
{
Student st;
st.init();
st.calctotal();
st.display();
return 0;
}
```

21. Create classes called Vechile1, Vechile2 and Vechile3 with each of having one private member named price. Add member function to set a price(say setPrice()) one each class. Add one more function max() that is friendly to all classes. max() function should compare private member named price of three classes and show maximum among them. Create one-one object of each class and then set a value on them. Display the maximum price among them.

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

class Vechile2;
class Vechile3;

class Vechile1
{
    private:
        long int price;public:
        void setPrice(long int p)
        {
            price=p;
        }
        friend void max(Vechile1,Vechile2,Vechile3);
};

class Vechile2
{
    private:
        long int price;
    public:
        void setPrice(long int p)
        {
            price=p;
        }
        friend void max(Vechile1,Vechile2,Vechile3);
};
```



```

class Vechile3
{
private:
long int price;
public:
void setPrice(long int p)
{
price=p;
}
friend void max(Vechile1,Vechile2,Vechile3);
};

void max(Vechile1 v1,Vechile2 v2,Vechile3 v3)
{
    if(v1.price>v2.price&&v1.price>v3.price)
    {
        cout<<"Maximum price="<<v1.price<<endl;
    }
    else if(v2.price>v3.price)
    {
        cout<<"Maximum price="<<v2.price<<endl;
    }
    else
    {
        cout<<"Maximum price="<<v3.price<<endl;
    }
}

int main()
{
Vechile1 v1;
Vechile2 v2;
Vechile3 v3;
v1.setPrice(5032323);
v2.setPrice(4543043);
v3.setPrice(1232343);
max(v1,v2,v3);
return 0;
}

```

22. Define a class to represent a bank account .Include the following

Data members

- Name of the depositor
- Account number
- Type of account
- Balance amount in the account

Member functions

- To assign Initial values
- To deposit an amount
- To withdraw an amount after checking the balance
- To display name and balance

Write main program to test the program

```
#include<iostream>
#include<string.h>
using namespace std;
class Bank
{
    char name[20];
    long int acno;
    char acctype[20];
    float bal;
public:
    Bank(long int ano, char n[], char at[], float b) //Parameterized Constructor
    {
        acno=ano;
        strcpy(name,n);
        strcpy(acctype, at);
        bal=b;
    }

    void deposit();
    void withdraw();
    void display();
};
void Bank::deposit()
{
    int damt;
    cout<<"Enter Deposit Amount"<<endl;
    cin>>damt;
    bal=bal+damt;
}
```

```

void Bank::withdraw()
{
    int wamt;
    cout<<"Enter Withdraw Amount"<<endl;
    cin>>wamt;
    if(wamt>bal)
    {
        cout<<"Cannot Withdraw Amount";
    }
    else
    {
        bal=bal-wamt;
    }
}

void Bank::display()
{
    cout<<"Account Number="<<acno<<endl;
    cout<<"Name="<<name<<endl;
    cout<<"Account Type:"<<acctype<<endl;
    cout<<"Balance ="<<bal<<endl;
}

int main()
{
    int acc_no;char name[100], acc_type[100];
    float balance;
    cout<<"Enter Details:"<<endl;
    cout<<"Enter Account Number"<<endl;
    cin>>acc_no;
    cout<<"Enter Name"<<endl;
    cin>>name;
    cout<<"Enter Account Type"<<endl;
    cin>>acc_type;
    cout<<"Enter Balance"<<endl;
    cin>>balance;
    Bank b1(acc_no,name,acc_type,balance);
    b1.deposit();
    b1.withdraw();
    b1.display();
    return 0;
}

```

23. Create a class Employee with data members (id, name, Post, address, salary) and read information of 20 Employees and display the name and post of employee whose salary is greater than 50000.

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

class Employee
{
private:
int eid;
char name[20];
char post[20];
char address[20];
float salary;
public:
void getdata()
{
cout<<"Enter the Employee id"<<endl;
cin>>eid;
cout<<"Enter the Employee name"<<endl;
cin>>name;
cout<<"Enter the post "<<endl;
cin>>post;
cout<<"Enter the address"<<endl;
cin>>address;
cout<<"Enter the salary"<<endl;
cin>>salary;
}
void display()
{
    if(salary>50000)
    {
        cout<<"Employee id="<<eid<<endl;
        cout<<"Employee name="<<name<<endl;
        cout<<"Post="<<post<<endl;
        cout<<"Address="<<address<<endl;
        cout<<"Salary="<<salary<<endl;
    }
}
};
```

```

int main()
{
Employee e[20];
int i;
for(i=0;i<20;i++)
{
cout<<"Enter the information of Employee:"<<i+1<<endl;
e[i].getdata();
}
cout<<"Information of Employee who has salary greater than 50000 are:"<<endl;
for(i=0;i<20;i++)
{
e[i].display();
}
return 0;
}

```

24.Using class write a program that receives inputs principle amount, time and rate. Keeping rate 12% as the default argument , calculate the simple interest for five customers.

```

#include<iostream>
using namespace std;
class customer
{
private:
float principle,rate,si;
int time;
public:
void setdata(float p,int t,float r=12);
void display();
};

void customer::setdata(float p,int t,float r)
{
principle=p;
time=t;
rate=r;
}
void customer::display()
{
si=(principle*time*rate)/100;
cout<<"Simple Interest="<<si<<endl;
}

```

```

int main()
{
float p;
int t,i;
customer c[5];
for(i=0;i<5;i++)
{
cout<<"Enter principle and time for customer"<<i+1<<endl;
cin>>p>>t;
c[i].setdata(p,t);
}
for(i=0;i<5;i++)
{
cout<<"for customer"<<i+1<<endl;
c[i].display();
}
return 0;
};

```

25. WAP to input marks of 35 students and find the highest marks using Dynamic memory allocation.

```

#include <iostream>
using namespace std;
int main ()
{
int i;
float *ptr,high=0;
ptr= new float[35];
for (i=0; i<35; i++)
{
cout <<"Enter the marks of student"<<i+1<<endl;
cin >> ptr[i];
}
for (i=0; i<35; i++)
{
    if(ptr[i]>high)
    {
        high=ptr[i];
    }
}
cout<<"Highest marks="<<high<<endl;
delete[] ptr;
return 0;
}

```