**Question No. 1:-**

Define a class Customer with the following specifications: Private Members: Customer\_no (integer type), Customer\_name (char (20) type), OrderQty (integer type), Price, TotalPrice, Discount, and Netprice: (all float types). Member Functions are:

Public members: A constructer to assign initial values of Customer no as 123, Customer name as “Rohit”, Order Quantity as (0) and Price, Discount, Netprice as (0.0) respectively. A Function Input( ) – to read data members (Customer\_no, Customer\_name, Quantity and Price) and call calDiscount() function. A function Show( ) –to display Customer details with values for all data members. Protected members: A function calDiscount( ) - to calculate Discount according to TotalPrice and to calculate value of NetPrice.

|  |  |
| --- | --- |
| **Condition for Total Price** | **Value of Discount** |
| TotalPrice >=50000 | Discount 25% of TotalPrice |
| TotalPrice >=25000 and TotalPrice <50000 | Discount 15% of TotalPrice |
| TotalPrice <250000 | Discount 10% of TotalPrice |
| Formula for calculating total price: **( TotalPrice = Price\*Qty)** | |
| Formula for Calculating Net Price: **( NetPrice = TotalPrice - Discount)** | |

**Table 1: Table for Net Price Calculation**

Write a program for customer class and enter data for two customers one using constructor and other using input function, perform necessary calculations and display value of all data members for both objects.

**Source code:-**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

#include<string.h>

class customer

{

int customer\_no,orderqty;

char customer\_name[20];

float price,totalprice,discount,netprice; public:

customer()

{

customer\_no=123;

strcpy(customer\_name,"Rohit");

orderqty=0;

price=0.0;

totalprice=0.0;

discount=0.0;

netprice=0.0;

}

void input()

{

cout<<"Enter the Customer Number"<<endl; cin>>customer\_no;

cout<<"Enter the Customer Name"<<endl; gets(customer\_name);

cout<<"Enter the Quantity of Order"<<endl; cin>>orderqty;

cout<<"Enter the Price of Order"<<endl; cin>>price;

caldiscount();

}

void display()

{

cout<<"\n\n"<<"Customer Number: "<<customer\_no<<endl<<"Customer Name: "<<customer\_name<<endl<<"Order Quantity: "<<orderqty<<endl<<"Price of each order: "<<price<<endl<<"Total Price: "<<totalprice<<endl<<"Discount you get: "<<discount<<endl<<"NetPrice: "<<netprice<<endl;

cout<<"\n\n--------\*THANK YOU\*--------\n\n"<<"------\*VISIT AGAIN\*------ \n\n"<<"----\*HAVE A NICE DAY\*----\n"<<endl;

}

protected:

void caldiscount()

{

totalprice=price\*orderqty;

if(totalprice>=50000)

{

discount=(totalprice\*25)/100;

}

else if(totalprice>=25000&&totalprice<50000)

{

discount=(totalprice\*15)/100;

}

else if(totalprice<25000)

{

discount=(totalprice\*10)/100;

}

netprice=totalprice-discount;

}

};

void main()

{

clrscr();

customer ob;

ob.display();

customer ob1;

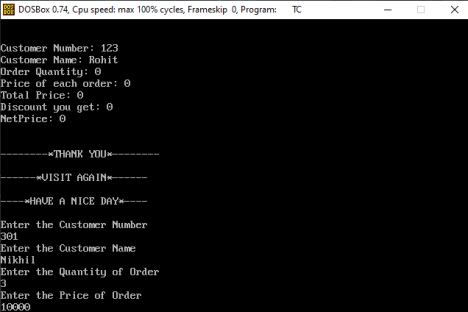
ob1.input();

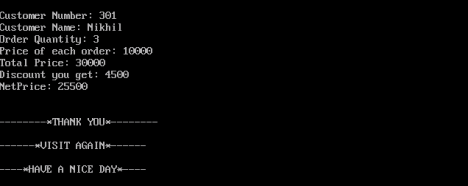
ob1.display();

getch();

}

**Output:-**

****

****

**Question No. 2:-**

Define a class FlightData in C++ with private data members as fltNum (storing flight number), fltDest (to store destination in km’s), fltDist (to store total distance), ppUnit(to store price (in float) of fuel per unit), totalCost (to store flight running cost). A member function calCost() to find out the flight running cost and store it in totalCost, calculated as per the following criteria:

|  |  |  |
| --- | --- | --- |
| **Distance** | **Fuel Required(in Units)** | **Base Charges** |
| <=500 | 1000 | 2865 |
| more than 500 and <=1500 | 3000 | 3875 |
| more than 1500 | 4000 | 4885 |

Flight running Cost is calculated by formula: totalCost=Base Charges + (Distance\*fuel required\*price per unit).

Store values of base charges, fixed for every object of class using 3 integer data members having values mentioned in table above. Public members of class are, a function to allow user to enter all data members value and a function to display all flight information with total flight running cost calculated using calCost function. Create an object of class entering all details calculating and displaying flight running cost for it.

**Source code:-**

#include<iostream.h>

#include<conio.h>

#include<stdio.h>

class flightdata

{

int fltnum,fltdist;

long int base,fuel;

char fltdest[50];

float ppunit,totalcost;

public:

void input()

{

cout<<"Enter flight Number"<<endl;

cin>>fltnum;

cout<<"Enter the destination of flight"<<endl;

gets(fltdest);

cout<<"Enter the distance travelled by the flight in Km's"<<endl; cin>>fltdist;

cout<<"Enter the price of fuel per unit"<<endl;

cin>>ppunit;

calcost();

}

void show()

{

cout<<"\n\n";

cout<<"Flight Number: "<<fltnum<<endl<<"Flight Destination: "<<fltdest<<endl<<"Flight Distance: "<<fltdist<<endl;

cout<<"Price of fuel per unit: "<<ppunit<<endl<<"Base Charge: "<<base<<endl<<"Amount of fuel required per unit: "<<fuel<<endl<<"Total flight running cost: "<<totalcost<<endl;

}

void calcost()

{

if(fltdist<=500)

{

base=2865;

fuel=1000;

}

else if(fltdist>500&&fltdist<=1500)

{

base=3875;

fuel=3000;

}

else if(fltdist>1500)

{

base=4885;

fuel=4000;

}

totalcost=base+(fltdist\*fuel\*ppunit); }

};

void main()

{

clrscr();

flightdata ob;

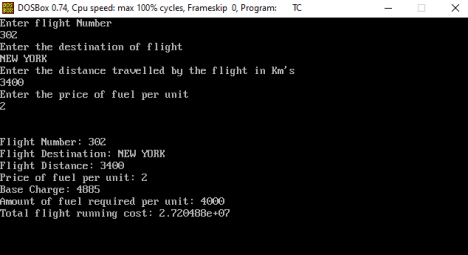
ob.input();

ob.show();

getch();

}

**Output:-**

****

**Question No. 3 :-**

Create two classes first with single integer data member and other class with other integer data members, have a function in program friend to both classes and using it find out which data member is greatest out of two, initialized using respective constructors of classes and print greatest number as result.

**Source code:-**

#include<iostream.h>

#include<conio.h>

class two;

class one

{

int x;

public:

one()

{

x=3;

}

friend void max(one,two);

};

class two

{

int y;

public:

two()

{

y=7;

}

friend void max(one,two);

};

void max(one a,two b)

{

if(a.x>b.y)

cout<<"Class one has greater data member as compared to class two"; else

cout<<"Class two has greater data member as compared to class one"; }

void main()

{

clrscr();

one a;

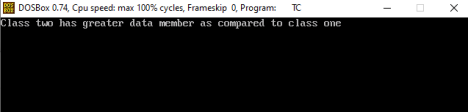
two b;

max(a,b);

getch();

}

**Output:-**

****

**Question No. 4:-**

Create a class Company with Private members as staffName[20] (character array for name), stDesig[15] (character array for designation), staffSalary(integer type variable for salary) and selectData[2] (a static class member array of integer type, used for identifying staff with highest and lowest salary). A function to find and store numbers in selectData to identify the person with highest salary and one with lowest salary. Public members of class are: a function to initialize staffName, stDesig & staffSalary with values taken at runtime. A function result() to display all details of staff members with highest and lowest salary. Create a program for entering data for 5 staff members and display the details of staff having highest and lowest salary.

**Source code:-**

#include<iostream.h>

#include<string.h>

#include<conio.h>

#include<stdio.h>

class Company

{

char staffName[20],stDesig[15];

public:

long long int staffsalary;

void accept\_details()

{

cout<<"Enter Employee Details"<<endl;

cout<<"-------------------------------------------"<<endl;

cout<<"Enter Employee Designation: ";

cin>>stDesig;

cout<<"Enter Employee Name: ";

cin>>staffName;

cout<<"Enter Salary: ";

cin>>staffsalary;

}

void display()

{

cout<<"\n Employee Designation: "<<stDesig; cout<<"\n Employee Name: "<<staffName; cout<<"\n Employee Salary: "<<staffsalary; }

}emp[5];

void main()

{

clrscr();

long long int max,min;

static long long int selectdata[2];

for(int k=0;k<5;k++)

{

emp[k].accept\_details();

}

max=emp[0].staffsalary;

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

{

if(emp[i].staffsalary>max)

{

max=emp[i].staffsalary;

selectdata[0]=i;

}

}

min=emp[0].staffsalary;

for(int j=1;j<5;j++)

{

if(emp[j].staffsalary<min)

{

min=emp[j].staffsalary;

selectdata[1]=j;

}

}

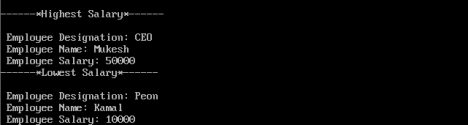
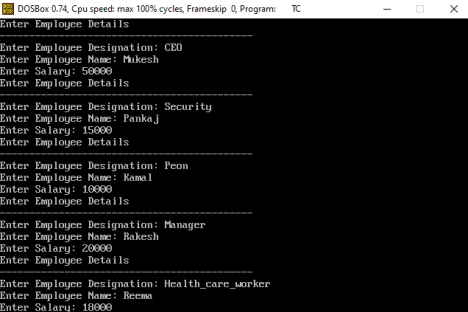
cout<<"\n------\*Highest Salary\*------"<<endl; emp[selectdata[0]].display();

cout<<"\n------\*Lowest Salary\*------"<<endl; emp[selectdata[1]].display();

getch();

}

**Output:-**

****