**Session 6 (unit-4): Friend functions and this keyword and Access Specifiers**

1. **How a member function of a class can act as a friend of another class.**

#include <iostream>

using namespace std;

class A; // forward declaration of A needed by B

class B

{

public:

void display(A obj); //no body defined, only declaration

};

class A

{

int x;

public:

A()

{

x = 4;

}

friend void B::display(A);

};

void B::display(A obj)

{

cout << obj.x << endl;

}

int main()

{

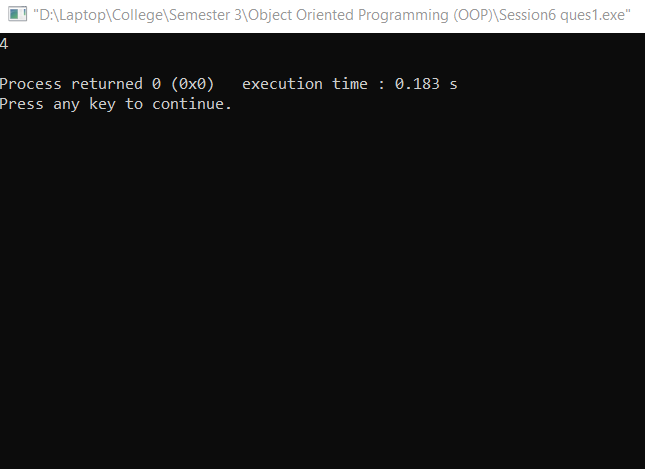
A a;

B b;

b.display(a);

return 0;

}



**Q2 State with the help of an example how ‘this’ keyword can be used in place of constructors in a class. Take an example of entering the student’s details and his marks in three subjects and displaying them using this keyword.**

#include<iostream>

#include<stdio.h>

using namespace std;

class A

{

private :

char name[25];

int roll;

int marks[3];

int i;

public :

void insert()

{

cout<<"Enter the name ";

gets(name);

cout<<"Enter Roll No. ";

cin>>roll;

cout<<"Enter the marks of 3 subject ";

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

{

cin>>marks[i];

}

}

void output()

{

cout<<"Name :";

cout<<this->name<<"\n";

cout<<"Roll No. :";

cout<<this->roll<<"\n";

cout<<"Marks of 3 subject: ";

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

cout<<this->marks[i]<<"\n";

}

};

int main()

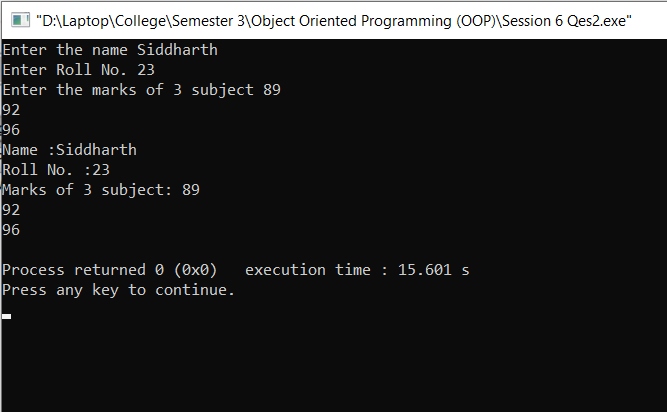
{

A ob;

ob.insert();

ob.output();

}



**Q3 WAP that can calculate the electricity bill of a person under following conditions:  
  
Unit consumed                                          Tariff  
  
First 100 units                                            Rs. 5 Per unit  
  
next 200 units                                            Rs. 7 Per unit  
  
next 300 units                                            Rs. 12 Per unit**

**The class must include details of the customer like his id, name and the functions read() to accept user inputs, display() to display total bill and a function cal\_tariff() to calculate his total bill. Use appropriate access specifier sections and define all the functions outside the scope of the class using the scope resolution operator.**

#include<iostream>

#include<stdio.h>

using namespace std;

class Electricity

{

private:

int id;

char name[25];

int unit;

int tariff;

public :

void read();

void cal\_tariff();

void display();

};

void Electricity::read()

{

cout<<"Enter the name : ";

gets(name);

cout<<"Enter your Id : ";

cin>>id;

cout<<"Enter the Amount of Units Consumed: ";

cin>>unit;

}

void Electricity::cal\_tariff()

{

if(unit<=100)

{

tariff=(unit\*5);

}

else if((unit>100)&&(unit<=200))

{

tariff=(100\*5)+(unit-100)\*7;

}

else if((unit>200)&&(unit<=300))

{

tariff=(100\*5)+(100\*7)+(unit-200)\*12;

}

}

void Electricity::display()

{

cout<<"Name : "<<name<<"\n";

cout<<"ID : "<<id<<"\n";

cout<<"Number Of Units Consumed :"<<unit<<"\n";

cout<<"Total Amount :"<<tariff;

}

int main()

{

Electricity ob;

ob.read();

ob.cal\_tariff();

ob.display();

}

