Object Oriented Programming

Lecture - 8

Scope

- Friend functions
- Friend classes
- Class exercises

Friend functions

- A function defined outside the scope of the class, yet has a right to access the members (both public/non-public) of the class
- To declare a function as a friend of a class, precede the function prototype in the class definition with keyword ***friend***

Example

```
#include<iostream>
class count
{ friend void setx(count &,int);
public: count() { x=0; }
void print() { std::cout<<x;}
private: int x;
};
```

```
void setx(count &c,int
val)
{ c.x=val;}


int main()
{
  count counter;
  counter.print();
  setx(counter,8);
  counter.print();
  return(0);
}
```

Friend classes

- To declare all member functions of class ClassTwo as friends of another class, place a declaration

  friend class ClassTwo;

- Neither symmetric nor transitive

Class assignment

- Take two classes A and B each having one data member (private) as float and int type numbers. Find the sum of the two numbers
  - Using friend function
  - By member function of class A
  - By member function of class B

```
#include<iostream.h>
#include<conio.h>
class B;
class A
{
  friend void sum(A,B);
  int x;
  public:
  A() { x=15; }
};
class B
{
  friend void sum(A,B);
  float y;
  public:
  B() { y=4.5; }
};
```

```
void sum(A a,B b) {
cout<<(a.x+b.y)<<endl; }
void main()
{
  clrscr();
  A a1;
  B b1;
  sum(a1,b1);
  getch();
}
```

```
#include<iostream.h>
#include<conio.h>
class B;
class A
{
  friend class B;
  int x;
  public:
  A() { x=5; }
};
class B
{
  float y;
  public:
  B() { y=4.5; }
  void sum(A a) { cout<<(a.x+y); }
};
```

```
void main()
{
  clrscr();
  A a1; B b;
  b.sum(a1);
  getch();
}
```

Assignment

- Explain Friend Functions and friend classes.