Class List Additions

```
//---------------------------------------------------------------------
//Copy constructor
//---------------------------------------------------------------------
List(List& L)
{
   New(100);
   for (L.First();!L.IsEol();L.Next()) {
      Insert(L.Member());
   }
}

//---------------------------------------------------------------------
//Copy assignment
//---------------------------------------------------------------------
List operator=(List& L)
{
   if (this!=&L) {      //Beware of self assignment
      Empty();
      for (L.First();!L.IsEol();L.Next()) Insert(L.Member());
   }
   return *this;
}

//---------------------------------------------------------------------
//Print the list {v1,v2,...,vn}
//---------------------------------------------------------------------
friend ostream& operator<< (ostream& o,const List& L)
{  o << "{";
   for (int a=1;a<=L.count;a++) {
      if (a>1) o << ",";
      o << L.L[a];
   }
   o << "}";
}
```

Class Set Additions

```
  friend Set& operator+(Set& s1,Set& s2)
  {
      Set* U=new Set;
      U->Union(s1,s2);
      return *U;
  }

  friend Set& operator*(Set& s1,Set& s2)
  {   Set* I=new Set;
      I->Intersection(s1,s2);
     return *I;
  }

  friend Set& operator-(Set& s1,Set& s2)
  {   Set* D=new Set;;
      D->Difference(s1,s2);
      return *D;
  }
```