Operator overloading

Lecture 14

Overloading ! operator

Overload ! operator for class point so that

```
void main()
{ point p(0,0,0);
if(!p)
  cout<<“ p is not origin !!”;
else
  cout<<“ p is origin”
};
```

Solution

```
class point
{  int x;  int y;  int z;
  public:
  point(int i=0,int k=0,int j=0) : x(i),
   y(k), z(j) { }
  void display()
  { cout<<"\n"<<x<<" "<<y<<"
   "<<z; }
  int operator !()
  { return (x!=0 || y!=0 || z!=0); }
};
```

```
void main()
{ point p(0,0,0);
if(!p)
  cout<<“ not origin
else
  cout<<“ p is origin
};
```

Overloading binary operators

- Binary operator can be overloaded as a member function with one argument or as a global function with two arguments

Example

```
class point
{  int x;  int y;  int z;
  public:
  point(int i=0,int k=0,int j=0) : x(i), y(k),
   z(j) { }
  void display()
  { cout<<"\n"<<x<<" "<<y<<" "<<z; }
  void operator +(point p)
  {cout<<(x+p.x)<<(y+p.y)<<(z+p.z);}
};
```

```
void main()
{ point p(0,1,1);
point q(3,2,4);
p+q;
};
```

Class assignment

What change do you need to make to the code to make a call

```
void main()
{ point p(0,1,1);
point q(3,2,4),s;
s=p+q;
s.display();
};
```

Example

```
class point
{  int x;  int y;  int z;
  public:
  point(int i=0,int k=0,int j=0) : x(i), y(k),
   z(j) { }
  void display()
  { cout<<"\n"<<x<<" "<<y<<" "<<z; }
  void operator +(int d)
  {cout<<(x+d)<<(y+d)<<(z+d);}
};
```

```
void main()
{ point p(0,1,1);
point q(3,2,4);
p+5;
};
```

Using global function

```
class point
{  int x;  int y;  int z;
  public:
  point(int i=0,int k=0,int
    j=0) : x(i), y(k), z(j) { }
  void display()
  { cout<<"\n"<<x<<"
    "<<y<<" "<<z; }
friend void operator
    +(point,point)
};
```

```
void operator +(point p1,point p2)
  { cout<<(p1.x+p2.x);
.....}
```

```
void main()
{ point p(0,1,1);
point q(3,2,4);
p+q;
};
```

Class assignment

- What change do you need to make to global function to make a call?

```
void main()
{ point p(0,1,1);
point q(3,2,4),s;
s=p+q;
s.display();
};
```

Example

```
class point
{  int x;  int y;  int z;
  public:
  point(int i=0,int k=0,int
    j=0) : x(i), y(k), z(j) { }
  void display()
  { cout<<"\n"<<x<<"
    "<<y<<" "<<z; }
friend void operator
    +(point,int)
};
```

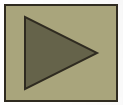
```
void operator +(point p1,int d)
  { cout<<(p1.x+d);
.......}
```

```
void main()
{ point p(0,1,1);
point q(3,2,4);
p+5;
};
```

Class exercise

Overload += operator for the point class

Overload == operator to check origin in point class



Home assignment

- Overload != operator for point class
- Overload = (assignment operator)
- Overload < operator (<=) for point class
- Overload > operator (>=) for point class
- Overload ++ and - - operator for point class