Operators (1A)

Young Won Lim
12/19/16

Pre- and Post- Increment / Decrement

```
x = a ++;

x = a −−;
```

Access First

```
x = a ++;

x = a −−;
```

Update Next

```
x = a ++;

x = a −−;
```

```
x = ++ a ;

x = −− a ;
```

Update First

```
x = ++ a ;

x = −− a ;
```

Access Next

```
x = a ++;

x = a −−;
```

Pre- and Post- Increment / Decrement

```
int a = 3;

 a ++;          a = a + 1;

 a −−;          a = a - 1;
```

```
cont int a = 3;
```

~~a ++;~~

~~a −−;~~

```
double b = 3.1;

 b ++;          b = b + 1;

 b −−;          b = b - 1;
```

```
const double b = 3.1;
```

~~b ++;~~

~~b −−;~~

Pointers with ++ and −− (1)

```
x = * (p ++);      x = *p++;
x = * (p −−);      x = *p−−;
```

Access First

```
x = * (p ++);
x = * (p −−);
```

Update Next

```
x = * (p ++);
x = * (p −−);
```

```
x = * (++ p);      x = *++p;
x = * (−− p);      x = *−−p;
```

Update First

```
x = * (++ p);
x = * (−− p);
```

Access Next

```
x = * (++ p);
x = * (−− p);
```

Pointers with ++ and −− (2)

```
x = (* p) ++;            x = ++ (* p);     x = ++*p;

x = (* p) −−;            x = −− (* p);     x = −−*p;
```

Access First

```
x = (* p) ++;

x = (* p) −−;
```

Update First

```
x = ++ (* p);

x = −− (* p);
```

Update Next

```
x = (* p) ++;

x = (* p) −−;
```

Access Next

```
x = ++ (* p);

x = −− (* p);
```

Pre and Post Increment / Decrement

```
v = *p++;

    v = *p        (access first)
    p = p+1       (increment later) (pointer increment)

v = (*p)++;

    v = *p        (access first)
    *p = *p+1     (increment later) (value increment)

v = *++p;

    p = p+1       (increment first) (pointer increment)
    v = *p        (access later)

v = ++*p;

    *p = *p+1     (increment first) (value increment)
    v = *p        (access later)
```

References

[1] Essential C, Nick Parlante
[2] Efficient C Programming, Mark A. Weiss
[3] C A Reference Manual, Samuel P. Harbison & Guy L. Steele Jr.
[4] C Language Express, I. K. Chun