KAIST
EE 209: Introduction to Programming Systems
Pointer-Related Operators

Key

```
p, p1, p2      Pointer variables

i              An integral expression
```

Operators Meaningful for Any Pointer Variable

Dereference Operator

```
*p The contents of the memory referenced by p.
```

Equality and Inequality Relational Operators

```
p1 == p2 1 if p1 is equal to p2, and 0 otherwise.
p1 != p2 1 if p1 is unequal to p2, and 0 otherwise.
```

Assignment Operator

```
p1 = p2 Side effect: Assign p2 to p1. The new value of p1.
```

Operators Meaningful for Pointers that Reference Array Elements

Arithmetic Operators

```
p + i          The address of the ith element after the one referenced by p.
i + p          The address of the ith element after the one referenced by p.
p - i          The address of the ith element before the one referenced by p.
p++            Side effect: Increment p to point to the next element.
               The previous value of p.
++p            Side effect: Increment p to point to the next element.
               The new value of p.
p--            Side effect: Decrement p to point to the previous element.
               The previous value of p.
--p            Side effect: Decrement p to point to the previous element.
               The new value of p.
```

Arithmetic Operators

```
p1 - p2 The "span" of p1 and p2.
```

Relational Operators

```
p1 < p2 1 if p1 is less than p2, and 0 otherwise.
p1 <= p2 1 if p1 is less than or equal to p2, and 0 otherwise.
p1 > p2 1 if p1 is greater than p2, and 0 otherwise.
p1 >= p2 1 if p1 is greater than or equal to p2, and 0 otherwise.
```

Assignment Operators

```
p += i          Side effect: Increment p so its value is the address of
                the ith element after the one referenced by p.
                The new value of p.
p -= i          Side effect: Decrement p so its value is the address of
                the ith element before the one referenced by p.
                The new value of p.
```

Disallowed

```
p1 + p2
i - p
i += p
i -= p
p == i
```

Array Subscripting Operator

```
p[i]            *(p + i), that is, the contents of memory at the address
                that is i elements after the address referenced by p.
```