Key point: *The increment operator (++) and decrement operator (--) are for incrementing and decrementing a variable by 1.*

- Post increment and post decrement

```
int i = 3, j = 3;
i++;            //i becomes 4
j--;            //j becomes 2

int k = i++;    //is equivalent to
int k = i;
i = i + 1;      //For example, if i=3, then k=3 after which i is incremented to 4.
```

- Pre-increment and pre-decrement

```
int i = 3, j = 3;
++i;            //i becomes 4
--j;            //j becomes 2

int k=++i;      //is equivalent to
i = i + 1;
int k = i;      //For example, if i=3, then i is incremented to 4 and assigned to k,
                //making k=4.
```

Table 2.5 Increment and Decrement Operators

Operator	*Name*	*Description*	*Example (assume i=1)*
++var	preincrement	Increment **var** by **1**, and use the new **var** value in the statement	**int j = ++i;** //j is 2, i is 2
var++	postincrement	Increment **var** by **1**, but use the original **var** value in the statement	**int j = i++;** //j is 1, i is 2
--var	predecrement	Decrement **var** by **1**, and use the new **var** value in the statement	**int j = --i;** //j is 0, i is 0
var--	postdecrement	Decrement **var** by **1**, but use the original **var** value in the statement	**int j = i--** //j is 1, i is 0

```
int i = 10;
int newNum = 10 * (++i);          //Equivalent to i = i + 1; int newNum = 10 * i;
System.out.print("i is " + i + ", newNum is " + newNum);
```

```
i is 11, newNum is 100
```

```
int i = 10;
int newNum = 10 * (++i);          //Equivalent to int newNum = 10 * i; i = i + 1;
System.out.print("i is " + i + ", newNum is " + newNum);
```

```
i is 11, newNum is 110
```