Operators

Operators are special symbols that perform specific operations on one, two, or three *operands*, and then return a result.

- The operators in the following table are listed according to precedence order.
- Operators with higher precedence are evaluated before operators with relatively lower precedence.
- When operators of equal precedence appear in the same expression, a rule must govern which is evaluated first.
- **All binary operators except for the assignment operators are evaluated from left to right;**
- **Assignment operators are evaluated right to left.**

Operators	Precedence
postfix	*expr++ expr--*
unary	++*expr* --*expr* +*expr* -*expr* ~ !
multiplicative	* / %
additive	+ -
shift	<< >> >>>
relational	< > <= >= instanceof
equality	== !=
bitwise AND	&
bitwise exclusive OR	^
bitwise inclusive OR	|
logical AND	&&
logical OR	||
ternary	? :
assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>= (>>> Unsigned right shift)

Questions and Exercises: Operators

Questions

- **Consider the following code snippet.**
  `arrayOfInts[j] > arrayOfInts[j+1]`
    1. **Which operators does the code contain?**

- **Consider the following code snippet.**
  `int i = 10;`
  `int n = i++%5;`

1. **What are the values of `i` and `n` after the code is executed?**
2. **What are the final values of `i` and `n` if instead of using the postfix increment operator (`i++`), you use the prefix version (`++i)`)?**

- **To invert the value of a `boolean`, which operator would you use?**
- **Which operator is used to compare two values, `=` or `==` ?**
- **Explain the following code sample: `result = someCondition ? value1 : value2;`**

Exercises

Change the following program to use compound assignments:

```
class ArithmeticDemo {

     public static void main (String[] args){

          int result = 1 + 2; // result is now 3
          System.out.println(result);

          result = result - 1; // result is now 2
          System.out.println(result);

          result = result * 2; // result is now 4
          System.out.println(result);

          result = result / 2; // result is now 2
          System.out.println(result);

          result = result + 8; // result is now 10
          result = result % 7; // result is now 3
          System.out.println(result);
     }
}
```

In the following program, explain why the value "6" is printed twice in a row:

```
class PrePostDemo {
    public static void main(String[] args){
        int i = 3;
        i++;
        System.out.println(i);    // "4"
        ++i;
        System.out.println(i);    // "5"
        System.out.println(++i);  // "6"
        System.out.println(i++);  // "6"
        System.out.println(i);    // "7"
    }
```

1. }