CECS 174 – Lecture 3 – Operators

Operators

Assignment	=	Assigns a value to the specified variable. The value is assigned from right to left.
Addition	+	Adds two values. Indicates positive value.
Subtraction	-	Subtracts two values. Indicates negative value.
Multiplication	*	Multiplies two values.
Division	/	Divides two values.
Modulus	%	Returns the remainder of division, signed as dividend.

Operator Order of Precedence

1. Anything contained in parenthesis.
2. Multiplication, Division, and Modulus.
3. Addition and Subtraction.
4. Left to right.
5. Assignment.

Example – evaluation is performed on the right according to the operator order of precedence, and then the resulting value is stored into the (single) variable on the left. When values are of different data types, the more precise type is returned.

```
int value = num1 * (num2 + 3);
```

Shortcut Operators

Increment	++	Increments the value of the specified variable by 1.
Decrement	--	Decrements the value of the specified variable by 1.
Add & Assign	+=	Adds a value to the variable and stores the result.
Subtract & Assign	-=	Subtracts a value from the variable and stores the result.
Multiply & Assign	*=	Multiplies a value with the variable and stores the result.
Divide & Assign	/=	Divides the variable by a value and stores the result.
Modulus & Assign	%=	Mods the variable by a value and stores the result.
Bit Shift Left	<<	Shifts the binary form of the value to the left. (*2)
Bit Shift Right	>>	Shifts the binary form of the value to the right. (/2)

Example – most of these operators allow you to perform tasks that you could do with the previous set of arithmetic operators, but using fewer keystrokes.

```
int x = 5, y = 2;
x += 3;   // same as x = x + 3;
x *= y;  // same as x = x * y;
x++;     // same as x = x + 1;
```

Pre and Post Increment

The increment and decrement operators have two forms: pre (++x) and post (x++). This means that when it is combined with another task within the same statement, the pre will perform the increment or decrement before the other task, and post will perform the increment or decrement after the other task.

Example:

```
System.out.println(++y); // inc to 3, prints 3

System.out.println(y++); // prints 3, inc to 4
```

Binary

In memory, values are stored in binary. Binary is made up of 1s and 0s called bits. Each byte in memory is made up of 8 bits. There are 1,024 bytes in a kilobyte, 1,024 kilobytes in a megabyte, 1,024 megabytes in a gigabyte, and so on. Different types of data take up different amounts of space in memory.

Notes About Binary –

Binary is similar to decimal in that it is structured the same way; each column is the next higher power of the base, in decimal it is powers of 10, and in binary it is powers of 2.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	0	1	0	1	1	0	1

=32+8+4+1= 45

Bit Shift –

Bit shifting is literally shifting the digits of the binary value to the right or left and filling in the missing gaps with 0s. Using the previous binary value shifted one place to the right, we get:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
0	1	0	1	1	0	1	0

=64+16+8+2= 90

Example – the value 45 bit shifted once to the right.

```
int x = 45;
x = x<<1;  //90 is now stored in x
```

CECS 174 – Lecture 3 Worksheet – Operator Practice

Example:

Evaluate in order given: `int num1=2, num2=5, num3=1, value;`

1. `value = -num1;` value = _____
2. `value = num1 + 7;` value = _____
3. `value = num1 * num2;` value = _____
4. `value = num2 / num1;` value = _____
5. `value = num2 % num1;` value = _____
6. `value = (num3 + num2) * num1;` value = _____
7. `value = num3 + num2 * num1;` value = _____
8. `value++;` value = _____
9. `value--;` value = _____
10. `value = value + num2 - 3;` value = _____
11. `value = num3++ + num1;` value = _____
12. `value *= ++num2;` value = _____
13. `value = num1<<1;` value = _____
14. `value = num>>2;` value = _____
15. `value = num3<<2;` value = _____