Operators

Assignment Operator (‘=’)

Now that you know about variables and how they are declared we can explain the assignment operator a little more.

```java
int numMarbles = 25;
```

The equals sign here is called the assignment operator because it assigns the number 25 to the variable “numMarbles”.

Unary and Binary Operators

Quickly to go over, unary operators are operators taking effect on a single number where as binary operators are operators taking effect on two numbers. This will be helpful when we go through the different operators in java.

Arithmetic Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication</td>
</tr>
<tr>
<td>/</td>
<td>Division</td>
</tr>
<tr>
<td>%</td>
<td>Modulo</td>
</tr>
</tbody>
</table>

*All arithmetic operators are binary*

You are most likely familiar with the first four operators and how they work however, I am willing to bet you have never had to use modulo or even heard of it.

**Modulo** returns the remainder of a division.

For instance, 7 % 3 would return 1 because when dividing 7 by 3 you are left with a remainder of 1. Here 7 is the dividend and 3 is the divisor.

```java
int remainder = dividend % divisor; //1 = 7 % 3
```

Setting up the other 4 operators is something I am sure you can do, but for good measure...

```java
int sum = 4+2;     //sum = 6
int difference = 7-3;   // difference = 4
int product = 7*7;   // product = 49
int quotient = 8/2;  //quotient = 4
```

***CHALLENGE***

*Using the arithmetic operators to develop a method to find if an integer is even or odd.*

***ANSWER AT END***
### Compound Assignment Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Operation</th>
<th>Example</th>
<th>Equivalent Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>+=</td>
<td>Performs addition operation and assignment</td>
<td><code>sum += 5;</code></td>
<td><code>sum = sum + 5;</code></td>
</tr>
<tr>
<td>-=</td>
<td>Performs subtraction operation and assignment</td>
<td><code>diff -= 3;</code></td>
<td><code>diff = diff - 3;</code></td>
</tr>
<tr>
<td>*=</td>
<td>Performs multiplication operation and assignment</td>
<td><code>prod *= 7;</code></td>
<td><code>prod = prod * 7;</code></td>
</tr>
<tr>
<td>/=</td>
<td>Performs division operation and assignment</td>
<td><code>quot /= 4;</code></td>
<td><code>quot = quot / 4;</code></td>
</tr>
<tr>
<td>%=</td>
<td>Performs modulo operation and assignment</td>
<td><code>modu %= 2;</code></td>
<td><code>modu = modu % 2</code></td>
</tr>
</tbody>
</table>

The compound assignment operators simply shorten the length of your code. This could help increase your code’s readability.

### Increment and Decrement Operators

Programmers tend to add and subtract by 1 more than anything else so for convenience sake the **increment** and **decrement operators** were given.

**Increment Operator (‘++’)**

Given: 

```c
int index = 0;
```

```c
index++; // equivalent to index = index + 1;
```

**Decrement Operator (‘--’)**

```c
index--; // equivalent to index = index - 1;
```

These two operators are used heavily in loops and you will see them throughout your programming adventures.
Code

```java
int dividend = 5; // pick any number
boolean isEven;

if ((dividend % 2) == 0)
{
    isEven = true
}
else
{
    isEven = false
}

System.out.println(isEven); // prints out true or false
```