Expressions and Assignment Statement

Computing integers

```
int x = 2;

int y = 4;

int sum = x + y;

System.out.println(sum);
```

Output: 6

Computing integers

The outcome is the same as mathematical operations:

int x = 4;

Int y = 8;

x + y $\longrightarrow$ 12

x - y $\longrightarrow$ 4

x * y $\longrightarrow$ 32

x / y $\longrightarrow$ 2

Modulus

Modulus (%): Divide and take the reminder

```
int x =17;

Int y = 5;


int z = x % y
```

What is the output?

Output: 2

Order of Operations

1. Parentheses ()
2. Multiplication and Division * / %
3. Addition and Subtraction + -

If there are multiple instances of same precedence, read left to right.

Order of operations

```
int myNum = (4 + 6 * (2 * 3) - 2)

System.out.println(mysteryNum)
```

What is the output?

Output: 38

Order of operations (int and double)

Integer and double values follow the same order of operations.

```
int x = 5;

int y = 2;

System.out.println(5 / 2);
```

Output: 2

The result is an integer, so it gets **truncated**.

Double division

```
double x = 5;

double y = 2;

System.out.println(5 / 2);
```

Output: 2.5

Double division retains the decimals.

Mixed division

```
double x = 5;

int y = 2;

System.out.println(5 / 2);
```

Output: 2.5

An int divided by double or a double divided by int will result in a double.

Dividing by Zero

int x = 5 / 0;

Double y 5.0 / 0;

This will throw an **ArithmeticException**.

What is the result of this expression?

150 % 100

1. 0

2. 100

3. 50

4. 3

What will be the output of the following code snippet?

```
public class Calculator
{
  public static void main(String[] args)
  {
    int first = 7;
    int second = 2;
    int result = first / second;
    System.out.println(result);
  }
}
```

1. 3.5
2. 3.50
3. 3
4. 3 1/2

Shortcuts

It is common in programming to add one or subtract one to a variable.

There are some shortcuts to do that:

myNum = myNum + 1;	myNum++;
myNum = myNum - 1;	myNum- -;

More shortcuts

It is common to modify the current value by adding/subtracting/multiplying/dividing another value.

x = x + y;	x += y;
x = x - y;	x -= y;
x = x * y;	x *= y;
x = x / y;	x /= y;
x = x % y;	x %= y;

Computing strings: String data type

values	Sequences of characters
Typical literals	"Hello, " "1 " " * "
operation	concatenate
operator	+

* **Literal:** Any fixed value (not variables or expressions).

“apcsa”, 26, 3.14, true, ‘&’, -2.5

* **Identifier:** A name given to a class, variable or method.

Concatenation examples

expression	value
"Hi, " + "Bob"	"Hi, Bob"
"1 " + " 2 " + "1"	"1 2 1"
"1234" + " + " + "99"	"1234 + 99"
"1234" + "99"	"123499"