**Arithmetic Expressions by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Operators**: + - \* / % ( plus, minus, time, divide, mod [ remainder after division])  
**Order of Operations**  
Same as math. What does this evaluate to? (1+2) \* 5 + 1 – 4 / 2 + 8 =

**Variables and literals:**  
Variables are names you make up to represent values of a certain type (char, int, double, boolean, String, etc.) Literals are fixed values ("hello", 75, 3.14159, 'x') Underline the literals:  
totalCost = cost + (cost\*0.05) – discount;  
plural = word + “s”;

**int vs double** If there’s no decimal in the problem, there’s no decimal in the answer. Chop off (truncate any decimal part)

**Variables that don’t change** are called constants and use the keyword final. Make daysIinWeek a constant:  
\_\_\_\_\_\_ int daysInWeek = 7;

|  |  |
| --- | --- |
| **What do each of these expressions evaluate to?**   * 2 \* 3 * 8 / 2 * 9 / 10 * -7 / 3 * 14 % 5 | **Given: int x = 5; double xx = 5.0;**   * x + xx = * xx / 2 = * x / 2 = * xx / x = |

**Arithmetic Expressions by \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Operators**: + - \* / % ( plus, minus, time, divide, mod [ remainder after division])  
**Order of Operations**  
Same as math. What does this evaluate to? (1+2) \* 5 + 1 – 4 / 2 + 8 =

**Variables and literals:**  
Variables are names you make up to represent values of a certain type (char, int, double, boolean, String, etc.) Literals are fixed values ("hello", 75, 3.14159, 'x') Underline the literals:  
totalCost = cost + (cost\*0.05) – discount;  
plural = word + “s”;

**int vs double** If there’s no decimal in the problem, there’s no decimal in the answer. Chop off (truncate any decimal part)

**Variables that don’t change** are called constants and use the keyword final. Make daysIinWeek a constant:  
\_\_\_\_\_\_ int daysInWeek = 7;

|  |  |
| --- | --- |
| **What do each of these expressions evaluate to?**   * 2 \* 3 * 8 / 2 * 9 / 10 * -7 / 3 * 14 % 5 | **Given: int x = 5; double xx = 5.0;**   * x + xx = * xx / 2 = * x / 2 = * xx / x = |