



## Worksheet

### Primitive Data Types: `ints`

You will need the `CalculationsV2` class from the previous lesson for this practice session.

- Create a new project called `Arithmetic Expressions`, in the `Unit02 Lessons` folder.
- Create a new class in the project called `Calculationsv3`.
- Copy the `Calculations_v2` class and paste it into the v3 version of the file.
- Don't forget to change the new class name from v2 to v3.

Study the highlighted lines of code shown below. These numbers should look familiar to you from earlier versions of the `CalculationsV2` class.

```
public static void main(String[ ] args)
{
    int iNum1 = 4;
    int iNum2 = 8;

    // Addition
    System.out.println("Addition");
    System.out.print(iNum1 + " plus " + iNum2 + " = ");
    System.out.println(iNum1 + iNum2);
    System.out.println(23.51 + 8.9325);
    System.out.println();

    // Subtraction
```

The highlighted code shows the assignment of values to `int` variables `iNum1` and `iNum2` and their use in new versions of the print statements.

- Variables `iNum1` and `iNum2` have been substituted wherever the integers 4 and 8 had been previously used in the Addition section of the code.
- Concatenation is used in the `print()` method to print the values of the variables and the String literals Remember there is no line feed with this version of the print statement.
- The `println()` method adds the values contained in the two `int` variables, prints the result, and issues a carriage return to terminate the line so the next output will appear on the line below.

The String literal and the arithmetic expression could have been printed in one statement instead of two, but using `print()` and `println()` in combination is a good strategy to reduce typos when printing error-prone output statements.

From now on, use variables instead of values when it is appropriate.