Modulo

One arithmetic operator that we haven’t covered yet and may be less familiar is a *modulo*. A modulo returns a *remainder*, what is left over when we divide a number by another number.

```
4 % 3 = 1
4 % 2 = 0
```

The modulo is the same as the percentage symbol, but it’s important to remember it’s different meaning in this context.

Modulos are useful because they let us know if a number “fits” into a larger number, or if there will be a remainder. For example, how many eggs will be left over if I try and fit 56 eggs into crates of a dozen eggs?

```
int eggs = 56;
int crateAmount = 12;

int eggsLeftOver = eggs % crateAmount;
Console.Write(eggsLeftOver); // prints 8
```

It can also be used to check if a number is odd or even. If a number is even, taking its modulo with 2 it will return a 0 and if it is odd it will return a 1:

```
int myNum = 85939824;
Console.Write(myNum % 2); // prints 0, so number is even
```

☑Instructions

1.

You’re teaching computer science in a classroom and need to break up your students into teams.

Start by creating a variable named `students` that has the value 18.

Hint

Since you can’t have half of a student, this variable should be an `int`.

2.

You need to find a number that will go evenly into 18 (without a remainder) so that there are an even number of students. The groups should have more than 2 students in each group, but no more than 5.

Create a variable named `groupSize`. Enter a number between 3 and 5.

Hint

Since you can’t have half of a student, this variable should be an `int`.

3.

Inside of a `Console.WriteLine()` statement, use the modulo operator to see if `students` will divide evenly into `groupSize`. If it does not, change the value of `groupSize` until it does.

Hint

If `students` divides evenly into `groupSize`, then the result of the operation should be 0. For example:

```
18 % 3; // returns 0
```