

# Exercise - Format the decimal output

10 minutes

In this exercise, you'll calculate the final GPA and modify the console output to achieve the desired reporting format. The GPA is equal to the sum of the total grade points divided by the sum of the total credit hours.

## Calculate the final GPA

1. In the .NET Editor, locate the `Console.WriteLine()` statements that are used to display the course information.
2. Remove the following code from the previous exercise:

C#

```
Console.WriteLine($"{totalGradePoints} {totalCreditHours}");
```

Since you've verified your values are correct, this line is no longer needed.

3. Create a blank code line above the `Console.WriteLine()` statements.
4. On the blank code line that you created, to initialize a variable that will store the final GPA, enter the following code:

C#

```
decimal gradePointAverage = totalGradePoints/totalCreditHours;
```

5. Take a moment to consider the data types you're dividing.

When you want the result of a division calculation to be a decimal value, either the dividend or divisor must be of type decimal (or both). When you use integer variables in the calculation, you need to use the cast operator to temporarily convert an integer to a decimal.

6. To retrieve a decimal value from the division, update your code as follows:

C#

```
decimal gradePointAverage = (decimal) totalGradePoints/totalCreditHours;
```

7. Navigate to the last `Console.WriteLine()` statement and create a new blank code line after the last statement.

8. To display the final GPA, enter the following code:

C#

```
Console.WriteLine($"Final GPA: {gradePointAverage}");
```

9. To view the results, select **Run**.

Compare your application's output with the following output:

```
English 101 4 3  
Algebra 101 3 3  
Biology 101 3 4  
Computer Science I 3 4  
Psychology 101 4 3  
Final GPA: 3.3529411764705882352941176471
```

## Format the decimal output

You might have noticed that decimal result contains many more digits than a standard GPA. In this task, you'll manipulate the decimal GPA value so that only three digits are displayed.

Ultimately, you want to display the first digit of the GPA, a decimal point, followed by the first two digits after the decimal. You can achieve this format by using variables to store the leading and trailing digits respectively, and then printing them together using `Console.WriteLine()`. You can use the math operations you learned to extract the leading and trailing digits.

### Note

As you continue your developer journey, you'll discover built-in operations that can automatically apply formatting to your data. For now, this is a great opportunity to solidify what you've learned so far.

1. Navigate to the top of the Console.WriteLine() statements.
2. Create a blank code line above the Console.WriteLine() statements.
3. On the blank code line that you created, to initialize a variable that will store the leading digit of the GPA, enter the following code:

C#

```
int leadingDigit = (int) gradePointAverage;
```

Notice that to extract the leading digit from the decimal, you're casting it to an integer value. This is a simple and reliable method because casting a fractional value will never round up the result. Meaning if the GPA is 2.99, casting the decimal value to an int will result in 2.

4. To initialize a variable that will store the first two digits after the decimal, enter the following code:

C#

```
int firstDigit = (int) (gradePointAverage * 10) % 10;
```

In the first half of this operation, you move the decimal one place to the right and cast it to an integer. In the second half, you use the remainder, or modulo, operator to get the remainder of division by 10, which isolates the last digit in the integer. Here's an example:

Suppose `gradePointAverage = 2.994573` Then, performing the operation on these values would result in the following steps:

C#

```
int firstDigit = (int) (2.994573 * 10) % 10;  
int firstDigit = 29 % 10;  
int firstDigit = 9;
```

And the resulting value of `firstDigit` is 9.

Next, you'll apply the same operation to retrieve the second digit.

5. On a new blank code line, enter the following code:

C#

```
int secondDigit = (int) (gradePointAverage * 100 ) % 10;
```

In this line, you move the decimal two places and use the modulo operator to retrieve the last digit.

6. To correct the final GPA output, update the last `Console.WriteLine()` statement as follows:

C#

```
Console.WriteLine($"Final GPA: {leadingDigit}.{firstDigit}{secondDigit}");
```

## Check Your Work

In this task, you'll run the code and verify that the output is correct.

1. Check that your code is similar to the following:

C#

```
string studentName = "Sophia Johnson";  
string course1Name = "English 101";  
string course2Name = "Algebra 101";  
string course3Name = "Biology 101";  
string course4Name = "Computer Science I";  
string course5Name = "Psychology 101";  
  
int course1Credit = 3;  
int course2Credit = 3;  
int course3Credit = 4;  
int course4Credit = 4;  
int course5Credit = 3;  
  
int gradeA = 4;  
int gradeB = 3;  
  
int course1Grade = gradeA;
```

```
int course2Grade = gradeB;
int course3Grade = gradeB;
int course4Grade = gradeB;
int course5Grade = gradeA;

int totalCreditHours = 0;
totalCreditHours += course1Credit;
totalCreditHours += course2Credit;
totalCreditHours += course3Credit;
totalCreditHours += course4Credit;
totalCreditHours += course5Credit;

int totalGradePoints = 0;
totalGradePoints += course1Credit * course1Grade;
totalGradePoints += course2Credit * course2Grade;
totalGradePoints += course3Credit * course3Grade;
totalGradePoints += course4Credit * course4Grade;
totalGradePoints += course5Credit * course5Grade;

decimal gradePointAverage = (decimal) totalGradePoints/totalCreditHours;

int leadingDigit = (int) gradePointAverage;
int firstDigit = (int) (gradePointAverage * 10 ) % 10;
int secondDigit = (int) (gradePointAverage * 100 ) % 10;

Console.WriteLine($"{course1Name} {course1Grade} {course1Credit}");
Console.WriteLine($"{course2Name} {course2Grade} {course2Credit}");
Console.WriteLine($"{course3Name} {course3Grade} {course3Credit}");
Console.WriteLine($"{course4Name} {course4Grade} {course4Credit}");
Console.WriteLine($"{course5Name} {course5Grade} {course5Credit}");

Console.WriteLine($"Final GPA: {leadingDigit}.{firstDigit}{secondDigit}");
```

2. To run your code and display the formatted output, select **Run**.
3. To verify that your code is working as expected, compare the output of your application with the following output:

#### Output

```
English 101 4 3
Algebra 101 3 3
Biology 101 3 4
Computer Science I 3 4
Psychology 101 4 3
Final GPA: 3.35
```

If your code displays different results, you'll need to review your code to find your error and make updates. Run the code again to see if you've fixed the problem. Continue updating and running your code until your code produces the expected results.

### Important

Be sure not to delete any of the code you've written so far. You'll build on this code in the next exercise.

## Module complete:

[Unlock achievement](#)[English \(United States\)](#)[✔ ✕ Your Privacy Choices](#)[⚙ Theme ▾](#)[Manage cookies](#)[Previous Versions](#)[Blog <sup>⌵</sup>](#)[Contribute](#)[Privacy <sup>⌵</sup>](#)[Terms of Use](#)[Trademarks <sup>⌵</sup>](#)

© Microsoft 2023

.NET Editor

Press CTRL + M, TAB to exit the editor

 Clear

 Run



1

Output