

# Exercise - Format the output using escaped character sequences

10 minutes

In this exercise, you'll modify the console output from the previous exercise to achieve the specified reporting format.

## Format the console output

In this task, you'll update your code to apply the finishing touches to the reported output. You'll start by including the student's name and adding a header row that includes the column labels. After that, you'll use character escape sequences to add extra padding between the columns of information.

1. In the .NET Editor, locate the `Console.WriteLine()` statements that are used to display the student's course information.
2. Create a blank code line above the `Console.WriteLine()` statements.
3. On the blank code line that you created, to add the student's name, enter the following code:

C#

```
Console.WriteLine($"Student: {studentName}\n");
```

Notice that you're including `\n` at the end of the text. In the "Perform basic string formatting with C#" module, you learned that the `\n` escape character sequence will cause a new line to be created. The location of the escape sequence is important. In this case, the `\n` is at the end of the information that you want to write to the console, so the new line will be added after "Student Name" is displayed.

4. Create a blank code line after the previous `Console.WriteLine()` statement.

5. On the blank code line that you created, to add a header for course information, enter the following code:

C#

```
Console.WriteLine("Course\t\t\t\tGrade\tCredit Hours");
```

Notice you're adding four tabs after the course name. The extra tab stops will allow extra space for longer course names.

6. To format your output as aligned columns of text, replace the spaces between words with the `\t` escape sequence as follows:

C#

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

The `\t` escape sequence will insert a tab stop between the text items. These tab stops should result in left-aligned columns of information.

Notice that you reduce the number of tabs for `course4Name` and `course5Name`. This difference is because those course names are longer than the others. In the previous step, you added extra tabs after the course name column to keep spacing consistent for both long and short course names.

7. To add padding to the final GPA result output, update the code as follows:

C#

```
Console.WriteLine($"Final GPA:\t\t\t\t {leadingDigit}.{firstDigit}{second-Digit}");
```

## 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($"Student: {studentName}\n");
Console.WriteLine("Course\t\t\t\tGrade\tCredit Hours");
```

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

Console.WriteLine($"
Final GPA:\t\t\t {leadingDigit}.{firstDigit}{second-
Digit}");
```

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

Student: Sophia Johnson

| Course             | Grade | Credit Hours |
|--------------------|-------|--------------|
| 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.

## Module complete:

Unlock achievement

[Manage cookies](#)[Previous Versions](#)[Blog](#)[Contribute](#)[Privacy](#)[Terms of Use](#)[Trademarks](#)

© Microsoft 2023

.NET Editor

Press CTRL + M, TAB to exit the editor Clear Run

1

Output

