

[Complete C# Masterclass](#)

Your progress

Share

Partial Classes and Methods

Partial Classes and Methods

Introduction

In this section of the course, we have covered several key concepts related to Object-Oriented Programming (OOP), including classes, methods, and inheritance. However, one important topic that was not included in the video lectures is Partial Classes and Methods. To ensure a smooth learning experience and keep the lessons concise, we have prepared this written explanation.

This article will introduce Partial Classes and Methods, explain why they are useful, provide a real-world analogy, and walk through an implementation with examples. By the end of this article, you will understand how to use this feature effectively in C# development.

1. What are Partial Classes and Methods?

Partial Classes allow a class definition to be split across multiple files. This is particularly useful in large projects where different developers might work on different aspects of the same class.

Similarly, Partial Methods enable the declaration of method signatures in one part of a class while allowing the implementation to be optional in another. This can help create cleaner and more maintainable code.

Analogy: A Collaborative Book

Imagine you and your friend are writing a book together. Instead of working on the same document, you split the chapters into separate files:

- You write Chapter 1 and Chapter 2.
- Your friend writes Chapter 3 and Chapter 4.

When the book is published, all chapters are combined into a single cohesive book.

Partial Classes work in a similar way. They allow different sections of a class to be developed separately while still belonging to the same class.

Partial Methods, on the other hand, are like notes or placeholders left by one writer, which the other writer may or may not decide to include in the final book.

2. Declaring and Using Partial Classes

Basic Syntax

A class can be split across multiple files using the partial keyword:

File: Employee_PersonalDetails.cs

1. `// Partial class declaration`
2. `public partial class Employee`
3. `{`
4. `public string FirstName { get; set; }`
5. `public string LastName { get; set; }`
6. `}`

File: Employee_JobDetails.cs

1. `// Another part of the partial class`
2. `public partial class Employee`
3. `{`
4. `public string JobTitle { get; set; }`
5. `public double Salary { get; set; }`
6. `}`

When compiled, these two files will be treated as one single class named Employee.

Step-by-Step Implementation

Step 1: Define Partial Class in Multiple Files

Create two separate files and use the partial keyword to split the Employee class.

Step 2: Use the Class in Your Program

```
1. class Program
2. {
3.     static void Main()
4.     {
5.         Employee emp = new Employee();
6.         emp.FirstName = "John";
7.         emp.LastName = "Doe";
8.         emp.JobTitle = "Software Developer";
9.         emp.Salary = 75000;
10.
11.         Console.WriteLine($"{emp.FirstName} {emp.LastName} works as a {emp.JobTitle}
            with a salary of {emp.Salary}.");
12.     }
13. }
```

Expected Output:

```
1. John Doe works as a Software Developer with a salary of 75000.
```

3. Declaring and Using Partial Methods

Partial methods allow developers to define method signatures in one part of the class without necessarily implementing them in another part.

Basic Syntax

```
1. public partial class Employee
2. {
3.     // Declaration of a partial method
4.     partial void OnJobAssigned();
```

```
5. }  
6.  
7. public partial class Employee  
8. {  
9.     // Optional implementation of the partial method  
10.    partial void OnJobAssigned()  
11.    {  
12.        Console.WriteLine("A new job has been assigned.");  
13.    }  
14. }
```

Step-by-Step Implementation

Step 1: Define a Partial Method

Create a partial void method in one file.

Step 2: Implement the Partial Method (Optional)

Implement the method in another part of the class.

Step 3: Call the Method in Your Program

```
1. public partial class Employee  
2. {  
3.     public void AssignJob(string jobTitle)  
4.     {  
5.         this.JobTitle = jobTitle;  
6.         OnJobAssigned(); // Call the partial method  
7.     }
```

8. }

If the method implementation exists, it will be executed. If it is missing, the compiler will simply ignore it without errors.

4. Comparing Partial Classes and Methods with Alternatives

Feature	Partial Classes	Partial Methods	Standard Approach	Use Case
Splitting large classes across files	Optional	method implementation	Single-file class & method definition	Flexibility
High – Helps large projects	Medium – Implementation is optional	Low – Everything must be explicitly defined	Compiler Behavior	Combined into one class at compile-time
Ignored if not implemented	Fully enforced			

5. When to Use Partial Classes and Methods

When to Use Partial Classes

- ✓ Large projects where multiple developers work on the same class.
- ✓ Auto-generated code that needs extension without modification.
- ✓ Keeping related functionalities separate for better organization.

When to Use Partial Methods

- ✓ When defining optional hooks that subclasses may implement.
- ✓ For auto-generated code where an optional implementation can be provided later.
- ✓ Reducing unnecessary code execution if no implementation is needed.

6. Best Practices and Common Mistakes

Best Practices

- ✓ Use Partial Classes to organize code efficiently.
- ✓ Use Partial Methods to provide extension points in class definitions.
- ✓ Keep related functionalities within the same partial class.

Common Mistakes

- ✗ Overusing partial classes when not needed, leading to fragmentation.
- ✗ Forgetting to declare partial in both class files.
- ✗ Trying to return values from partial methods (they must be void).

7. Conclusion

Partial classes and methods are powerful features in C# that provide flexibility in structuring code. They allow developers to split class definitions across multiple files and define methods that may or may not be implemented, making them especially useful in large projects and auto-generated code scenarios.

If you have any questions, feel free to ask in the Q&A.

Happy coding!

Course content

Course content

Overview

Q&A Questions and answers

Notes

Announcements

Reviews

Learning tools

Section 1: UPDATED: Introduction, Overview of Visual Studio, DataTypes And Variables

51 / 56 | 3hr 6min 51 of 56 lectures completed 3hr 6min

Section 2: UPDATED: Making Decisions

20 / 28 | 1hr 33min 20 of 28 lectures completed 1hr 33min

Section 3: UPDATED: Loops

22 / 24 | 1hr 37min 22 of 24 lectures completed 1hr 37min

Section 4: UPDATED: Functions and Methods

17 / 20 | 1hr 34min 17 of 20 lectures completed 1hr 34min

Section 5: UPDATED: Object Oriented Programming (OOP)

18 / 43 | 3hr 10min 18 of 43 lectures completed 3hr 10min

- Lecture incomplete. Progress cannot be changed for this item.

Play

109. Objects Intro

2min

- Lecture completed. Progress cannot be changed for this item.

Play

110. Introduction To Classes And Objects

3min

- Lecture completed. Progress cannot be changed for this item.

Play

111. Creating our First own Class

8min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

112. Member Variable and Custom Constructor

7min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

113. Properties - Autogenerated - to protect our member variable

6min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

114. Defining how a property is set

8min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

115. Modifying the Get of our Property Part 1

7min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

116. Modifying the Get of our Property part 2

5min

- Lecture completed. Progress cannot be changed for this item.

Play

117. Having Multiple Constructors

7min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

118. Default Constructor and Use Cases

6min

Resources

- Lecture completed. Progress cannot be changed for this item.

Start

Quiz 12: Understanding Constructors

- Lecture completed. Progress cannot be changed for this item.

Play

119. Methods in Classes

7min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

120. Methods in Classes in more detail

8min

Resources

- Lecture completed. Progress cannot be changed for this item.

Start

121. Expression Bodied Members in C#

3min

- Lecture completed. Progress cannot be changed for this item.

Start

122. What are Inner Classes in C#?

3min

- Lecture completed. Progress cannot be changed for this item.

Start

123. Partial Classes and Methods

3min

- Lecture completed. Progress cannot be changed for this item.

Play

124. Optional Parameters

4min

Resources

- Lecture completed. Progress cannot be changed for this item.

Play

125. Named Parameters

3min

- Lecture completed. Progress cannot be changed for this item.

Start

126. Operator Overloading in C#

3min

- Lecture incomplete. Progress cannot be changed for this item.

Start

127. Passing Arguments by Value and by Reference

4min

- Lecture incomplete. Progress cannot be changed for this item.

Play

128. Computed Properties and No Constructor

3min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

129. Static Methods

7min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Start

Coding Exercise 10: Using Static Methods

- Lecture incomplete. Progress cannot be changed for this item.

Play

130. Static Fields

3min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Start

131. Static Keyword Considerations

3min

- Lecture incomplete. Progress cannot be changed for this item.

Start

132. The is Operator and the as Operator in C#

3min

- Lecture incomplete. Progress cannot be changed for this item.

Play

133. Public and Private Keywords

5min

- Lecture incomplete. Progress cannot be changed for this item.

Play

134. ID Key and readonly

7min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

135. Read Only Properties

3min

- Lecture incomplete. Progress cannot be changed for this item.

Start

Coding Exercise 11: Working with Read-Only Properties

- Lecture incomplete. Progress cannot be changed for this item.

Play

136. Write Only Properties

5min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

137. Const and ReadOnly

5min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Start

Quiz 13: Working with Read-Only Properties

- Lecture incomplete. Progress cannot be changed for this item.

Play

138. Quiz Project Introduction

4min

- Lecture incomplete. Progress cannot be changed for this item.

Play

139. QuizApp - Question Class

5min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

140. Keyword This

3min

- Lecture incomplete. Progress cannot be changed for this item.

Play

141. Displaying Questions

6min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

142. Displaying Answers, Console.WriteLine and Console.ForegroundColor

7min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

143. Getting the UserInput and checking if it is right

6min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

144. Displaying Multiple Questions and if we are right or wrong

8min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Play

145. Displaying the Results

8min

Resources

- Lecture incomplete. Progress cannot be changed for this item.

Start

146. CHEATSHEET - Object Oriented Programming in C#

0min

- Lecture incomplete. Progress cannot be changed for this item.

Start

Coding Exercise 12: ADVANCED EXERCISE: Creating a Class with Properties and Methods

Section 6: UPDATED: Collections in C#

0 / 27 | 2hr 1min0 of 27 lectures completed2hr 1min

Section 7: UPDATED: Error Handling

0 / 14 | 45min0 of 14 lectures completed45min

Section 8: UPDATED: Inheritance

0 / 22 | 1hr 21min0 of 22 lectures completed1hr 21min

Section 9: UPDATED: Interfaces and Polymorphism

0 / 24 | 1hr 22min0 of 24 lectures completed1hr 22min

Section 10: UPDATED: Structs in C#

0 / 9 | 58min0 of 9 lectures completed58min

Section 11: UPDATED: Events and delegates

0 / 14 | 1hr 21min0 of 14 lectures completed1hr 21min

Section 12: UPDATED: Regular Expressions

0 / 11 | 43min0 of 11 lectures completed43min

Section 13: WPF - Windows Presentation Foundation

0 / 42 | 2hr 31min0 of 42 lectures completed2hr 31min

Section 14: WPF Project - Currency Converter - Part 1

0 / 8 | 1hr 14min0 of 8 lectures completed1hr 14min

Section 15: Using Databases With C#

0 / 12 | 2hr 2min0 of 12 lectures completed2hr 2min

Section 16: WPF Project - Currency Converter - Part 2

0 / 9 | 1hr 31min0 of 9 lectures completed1hr 31min

Section 17: Linq

0 / 13 | 2hr 18min0 of 13 lectures completed2hr 18min

Section 18: WPF Project - Currency Converter with GUI Database and API - Part 3

0 / 3 | 31min0 of 3 lectures completed31min

Section 19: The exercises for your coding interviews

0 / 4 | 5min0 of 4 lectures completed5min

Section 20: C# Clean Code

0 / 24 | 1hr 37min0 of 24 lectures completed1hr 37min

Section 21: C# Generics

0 / 18 | 1hr 38min0 of 18 lectures completed1hr 38min

Section 22: Threads

0 / 8 | 1hr 10min0 of 8 lectures completed1hr 10min

Section 23: Unit Testing - Test Driven Development TDD

0 / 36 | 3hr 24min0 of 36 lectures completed3hr 24min

Section 24: UNITY - Basics

0 / 16 | 1hr 35min0 of 16 lectures completed1hr 35min

Section 25: UNITY - Building the Game Pong with Unity

0 / 20 | 2hr 34min0 of 20 lectures completed2hr 34min

Section 26: UNITY - Building a Zig Zag Clone With Unity

0 / 18 | 2hr 11min0 of 18 lectures completed2hr 11min

Section 27: UNITY - Building a Fruit Ninja Clone With Unity

0 / 14 | 2hr 8min0 of 14 lectures completed2hr 8min

Section 28: Thank you for completing the course!

0 / 1 | 4min0 of 1 lecture completed4min

Section 29: Bonus

0 / 1 | 1min0 of 1 lecture completed1min

Teach the world online

Create an online video course, reach students across the globe, and earn money

[Teach on Udemy](#)

Top companies choose [Udemy Business](#) to build in-demand career skills.

Explore top skills and certifications

Certifications by Issuer

- [Amazon Web Services \(AWS\) Certifications](#)
- [Six Sigma Certifications](#)
- [Microsoft Certifications](#)
- [Cisco Certifications](#)
- [Tableau Certifications](#)
- [See all Certifications](#)

Web Development

- [Web Development](#)
- [JavaScript](#)
- [React JS](#)
- [Angular](#)
- [Java](#)

IT Certifications

- [Amazon AWS](#)
- [AWS Certified Cloud Practitioner](#)

- [AZ-900: Microsoft Azure Fundamentals](#)
- [AWS Certified Solutions Architect - Associate](#)
- [Kubernetes](#)

Leadership

- [Leadership](#)
- [Management Skills](#)
- [Project Management](#)
- [Personal Productivity](#)
- [Emotional Intelligence](#)

Certifications by Skill

- [Cybersecurity Certification](#)
- [Project Management Certification](#)
- [Cloud Certification](#)
- [Data Analytics Certification](#)
- [HR Management Certification](#)
- [See all Certifications](#)

Data Science

- [Data Science](#)
- [Python](#)
- [Machine Learning](#)
- [ChatGPT](#)
- [Deep Learning](#)

Communication

- [Communication Skills](#)
- [Presentation Skills](#)
- [Public Speaking](#)

- [Writing](#)
- [PowerPoint](#)

Business Analytics & Intelligence

- [Microsoft Excel](#)
- [SQL](#)
- [Microsoft Power BI](#)
- [Data Analysis](#)
- [Business Analysis](#)

About

- [About us](#)
- [Careers](#)
- [Contact us](#)
- [Blog](#)
- [Investors](#)

Discovery Udemy

- [Get the app](#)
- [Teach on Udemy](#)
- [Plans and Pricing](#)
- [Affiliate](#)
- [Help and Support](#)

Udemy for Business

- [Udemy Business](#)

Legal & Accessibility

- [Accessibility statement](#)
- [Privacy policy](#)
- [Sitemap](#)

- [Terms](#)

© 2025 Udemy, Inc.