

O'REILLY®

Fourth  
Edition

Head First

C#

A Learner's Guide to  
Real-World Programming  
with C# and .NET Core

---

Andrew Stellman  
& Jennifer Greene



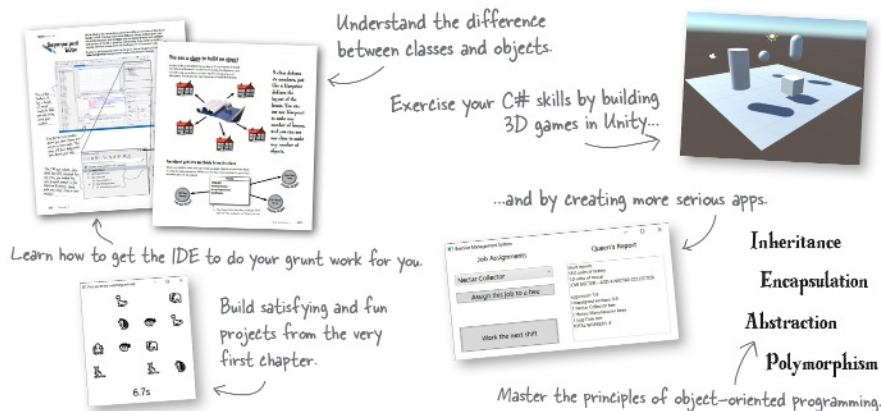
A Brain-Friendly Guide

# Head First

# C#

## What will you learn from this book?

Dive into C# and create apps, user interfaces, games, and more using this fun and highly visual introduction to C#, .NET Core, and Visual Studio. With this completely updated guide, which covers C# 8.0 and Visual Studio 2019, beginning programmers like you will build a fully functional game in the opening chapter. Then you'll learn how to use classes and object-oriented programming, create 3D games in Unity, and query data with LINQ. And you'll do it all by solving puzzles, doing hands-on exercises, and building real-world applications. By the time you're done, you'll be a solid C# programmer—and you'll have a great time along the way!



## What's so special about this book?

Based on the latest research in cognitive science and learning theory, *Head First C#* uses a visually rich format to engage your mind rather than a text-heavy approach that puts you to sleep. Why waste your time struggling with new concepts? This multisensory learning experience is designed for the way your brain really works.

.NET

US \$64.99

CAN \$85.99

ISBN: 978-1-491-97670-8



9

O'REILLY®

"Thank you so much!  
Your books have  
helped me to launch  
my career."

—Ryan White  
Game Developer

"Andrew and Jennifer  
have written a  
concise, authoritative,  
and most of all, fun  
introduction to C#  
development."

—Jon Galloway  
Senior Program Manager on the  
.NET Community Team  
at Microsoft

"If you want to learn  
C# in depth and have  
fun doing it, this is THE  
book for you."

—Andy Parker  
Fledgling C# programmer

## **Praise for *Head First C#***

“Thank you so much! Your books have helped me to launch my career.”

—**Ryan White, Game Developer**

“If you’re a new C# developer (welcome to the party!), I highly recommend *Head First C#*. Andrew and Jennifer have written a concise, authoritative, and most of all, fun introduction to C# development. I wish I’d had this book when I was first learning C#!”

—**Jon Galloway, Senior Program Manager on the .NET Community Team, Microsoft**

“Not only does *Head First C#* cover all the nuances it took me a long time to understand, it has that Head First magic going on where it is just a super fun read.”

—**Jeff Counts, Senior C# Developer**

“*Head First C#* is a great book with fun examples that keep learning interesting.”

—**Lindsey Bieda, Lead Software Engineer**

“*Head First C#* is a great book, both for brand-new developers and developers like myself coming from a Java background. No assumptions are made as to the reader’s proficiency, yet the material builds up quickly enough for those who are not complete newbies—a hard balance to strike. This book got me up to speed in no time for my first large-scale C# development project at work—I highly recommend it.”

—**Shalewa Odusanya, Principal**

“*Head First C#* is an excellent, simple, and fun way of learning C#. It’s the best piece for C# beginners I’ve ever seen—the samples are clear, the topics are concise and well written. The mini-games that guide you through the different programming challenges will definitely stick the knowledge to your brain. A great learn-by-doing book!”

—**Johnny Halife, Partner**

“*Head First C#* is a comprehensive guide to learning C# that reads like a conversation with a friend. The many coding challenges keep it fun, even when the concepts are tough.”

—**Rebeca Dunn-Krahn, founding Partner, Sempahore Solutions**

“I’ve never read a computer book cover to cover, but this one held my interest from the first page to the last. If you want to learn C# in depth and have fun doing it, this is THE book for you.”

—**Andy Parker, fledgling C# Programmer**

## More Praise for *Head First C#*

“It’s hard to really learn a programming language without good, engaging examples, and this book is full of them! *Head First C#* will guide beginners of all sorts to a long and productive relationship with C# and the .NET Framework.”

—**Chris Burrows, Software Engineer**

“With *Head First C#*, Andrew and Jenny have presented an excellent tutorial on learning C#. It is very approachable while covering a great amount of detail in a unique style. If you’ve been turned off by more conventional books on C#, you’ll love this one.”

—**Jay Hilyard, Director and Software Security Architect, and author of *C# 6.0 Cookbook***

“I’d recommend this book to anyone looking for a great introduction into the world of programming and C#. From the first page onwards, the authors walk the reader through some of the more challenging concepts of C# in a simple, easy-to-follow way. At the end of some of the larger projects/labs, the reader can look back at their programs and stand in awe of what they’ve accomplished.”

—**David Sterling, Principal Software Developer**

“*Head First C#* is a highly enjoyable tutorial, full of memorable examples and entertaining exercises. Its lively style is sure to captivate readers—from the humorously annotated examples to the Fireside Chats, where the abstract class and interface butt heads in a heated argument! For anyone new to programming, there’s no better way to dive in.”

—**Joseph Albahari, inventor of LINQPad, and coauthor of *C# 8.0 in a Nutshell* and *C# 8.0 Pocket Reference***

“[*Head First C#*] was an easy book to read and understand. I will recommend this book to any developer wanting to jump into the C# waters. I will recommend it to the advanced developer that wants to understand better what is happening with their code. [I will recommend it to developers who] want to find a better way to explain how C# works to their less-seasoned developer friends.”

—**Giuseppe Turitto, Director of Engineering**

“Andrew and Jenny have crafted another stimulating Head First learning experience. Grab a pencil, a computer, and enjoy the ride as you engage your left brain, right brain, and funny bone.”

—**Bill Mietelski, Advanced Systems Analyst**

“Going through this *Head First C#* book was a great experience. I have not come across a book series which actually teaches you so well... This is a book I would definitely recommend to people wanting to learn C#.”

—**Krishna Pala, MCP**

## **Praise for other *Head First* books**

“I received the book yesterday and started to read it...and I couldn't stop. This is definitely très 'cool.' It is fun, but they cover a lot of ground and they are right to the point. I'm really impressed.”

—**Erich Gamma, IBM Distinguished Engineer, and coauthor of *Design Patterns***

“One of the funniest and smartest books on software design I've ever read.”

—**Aaron LaBerge, SVP Technology & Product Development, ESPN**

“What used to be a long trial and error learning process has now been reduced neatly into an engaging paperback.”

—**Mike Davidson, former VP of Design, Twitter, and founder of Newsvine**

“Elegant design is at the core of every chapter here, each concept conveyed with equal doses of pragmatism and wit.”

—**Ken Goldstein, Executive VP & Managing Director, Disney Online**

“Usually when reading through a book or article on design patterns, I'd have to occasionally stick myself in the eye with something just to make sure I was paying attention. Not with this book. Odd as it may sound, this book makes learning about design patterns fun.

“While other books on design patterns are saying ‘Bueller... Bueller... Bueller...’ this book is on the float belting out ‘Shake it up, baby!’”

—**Eric Wuehler**

“I literally love this book. In fact, I kissed this book in front of my wife.”

—**Satish Kumar**

## **Related books from O'Reilly**

C# 8.0 in a Nutshell  
C# 8.0 Pocket Reference  
C# Database Basics  
C# Essentials, 2nd Edition  
Concurrency in C# Cookbook, 2nd Edition  
Mobile Development with C#  
Programming C# 8.0

## **Other books in O'Reilly's *Head First* series**

Head First 2D Geometry	Head First Networking
Head First Agile	Head First Object-Oriented Analysis and Design
Head First Ajax	Head First PHP & MySQL
Head First Algebra	Head First Physics
Head First Android Development	Head First PMP
Head First C	Head First Programming
Head First Data Analysis	Head First Python
Head First Design Patterns	Head First Rails
Head First EJB	Head First Ruby
Head First Excel	Head First Ruby on Rails
Head First Go	Head First Servlets and JSP
Head First HTML5 Programming	Head First Software Development
Head First HTML with CSS and XHTML	Head First SQL
Head First iPhone and iPad Development	Head First Statistics
Head First Java	Head First Web Design
Head First JavaScript Programming	Head First WordPress
Head First Kotlin	
Head First jQuery	
Head First Learn to Code	
Head First Mobile Web	

# Head First C#

Fourth Edition

WOULDN'T IT BE DREAMY IF  
THERE WAS A C# BOOK THAT WAS  
MORE FUN THAN MEMORIZING  
A DICTIONARY? IT'S PROBABLY  
NOTHING BUT A FANTASY...



Andrew Stellman

Jennifer Greene

Beijing • Boston • Farnham • Sebastopol • Tokyo

**O'REILLY®**

# Head First C#

## Fourth Edition

by Andrew Stellman and Jennifer Greene

Copyright © 2021 Jennifer Greene, Andrew Stellman. All rights reserved.

Printed in the United States of America.

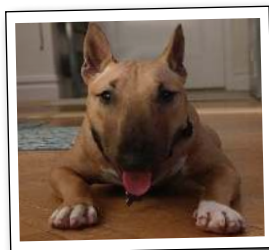
Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

O'Reilly Media books may be purchased for educational, business, or sales promotional use. Online editions are also available for most titles (<http://oreilly.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or [corporate@oreilly.com](mailto:corporate@oreilly.com).

<b>Series Creators:</b>	Kathy Sierra, Bert Bates
<b>Cover Designer:</b>	Ellie Volckhausen
<b>Brain Image on Spine:</b>	Eric Freeman
<b>Editors:</b>	Nicole Taché, Amanda Quinn
<b>Proofreader:</b>	Rachel Head
<b>Indexer:</b>	Potomac Indexing, LLC
<b>Illustrator:</b>	Jose Marzan
<b>Page Viewers:</b>	Greta the miniature bull terrier and Samosa the Pomeranian

## Printing History:

November 2007: First Edition  
May 2010: Second Edition  
August 2013: Third Edition  
December 2020: Fourth Edition



The O'Reilly logo is a registered trademark of O'Reilly Media, Inc. The *Head First* series designations, *Head First C#*, and related trade dress are trademarks of O'Reilly Media, Inc.

Microsoft, Windows, Visual Studio, MSDN, the .NET logo, Visual Basic, and Visual C# are registered trademarks of Microsoft Corporation.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and O'Reilly Media, Inc., was aware of a trademark claim, the designations have been printed in caps or initial caps.

While every precaution has been taken in the preparation of this book, the publisher and the authors assume no responsibility for errors or omissions, or for damages resulting from the use of the information contained herein.

No bees, space aliens, or comic book heroes were harmed in the making of this book.

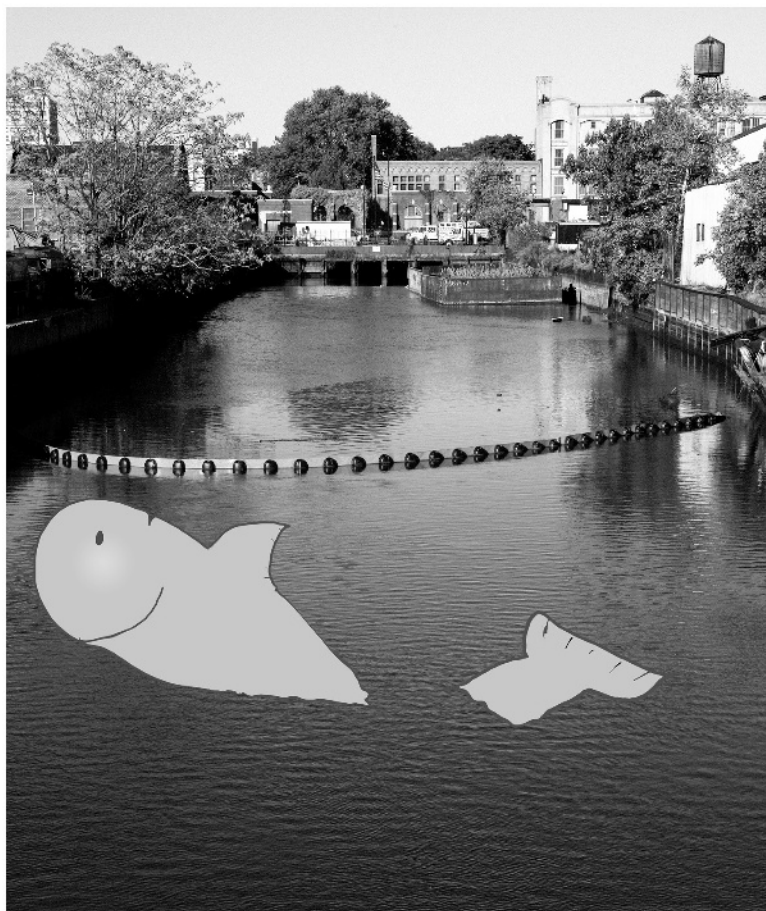
ISBN: 978-1-491-97670-8

[LSI]

[2020-12-18]



*This book is dedicated to the loving memory of Sludgie the Whale,  
who swam to Brooklyn on April 17, 2007.*



*You were only in our canal for a day,  
but you'll be in our hearts forever.*

THANKS FOR READING OUR BOOK!  
WE REALLY LOVE WRITING ABOUT  
THIS STUFF, AND WE HOPE YOU GET  
A LOT OUT OF IT...

Andrew

This photo (and the photo of the  
Gowanus Canal) by Nisha Sondhe



...BECAUSE  
WE KNOW YOU'RE  
GOING TO HAVE  
A GREAT TIME  
LEARNING C#.

Jenny

**Andrew Stellman**, despite being raised a New Yorker, has lived in Minneapolis, Geneva, and Pittsburgh... *twice*, first when he graduated from Carnegie Mellon's School of Computer Science, and then again when he and Jenny were starting their consulting business and writing their first book for O'Reilly.

Andrew's first job after college was building software at a record company, EMI-Capitol Records—which actually made sense, as he went to LaGuardia High School of Music & Art and the Performing Arts to study cello and jazz bass guitar. He and Jenny first worked together at a company on Wall Street that built financial software, where he was managing a team of programmers. Over the years he's been a vice president at a major investment bank, architected large-scale real-time backend systems, managed large international software teams, and consulted for companies, schools, and organizations, including Microsoft, the National Bureau of Economic Research, and MIT. He's had the privilege of working with some pretty amazing programmers during that time, and likes to think that he's learned a few things from them.

When he's not writing books, Andrew keeps himself busy writing useless (but fun) software, playing (and making) both music and video games, practicing krav maga, tai chi, and aikido, and owning a crazy Pomeranian.

**Jennifer Greene** studied philosophy in college but, like everyone else in the field, couldn't find a job doing it. Luckily, she's a great software engineer, so she started out working at an online service, and that's the first time she really got a good sense of what good software development looked like.

She moved to New York in 1998 to work on software quality at a financial software company. She's managed teams of developers, testers, and PMs on software projects in media and finance since then.

Jenny has traveled all over the world to work with different software teams and build all kinds of cool projects.

She loves traveling, watching Bollywood movies, reading the occasional comic book, playing video games, and hanging out with her huge Siberian cat, Sascha, and her miniature bull terrier, Greta.

Jenny and Andrew have been building software and writing about software engineering together since they first met in 1998. Their first book, *Applied Software Project Management*, was published by O'Reilly in 2005. Other Stellman and Greene books for O'Reilly include *Beautiful Teams* (2009), *Learning Agile* (2014), and their first book in the Head First series, *Head First PMP* (2007), now in its fourth edition.

They founded Stellman & Greene Consulting in 2003 to build a really neat software project for scientists studying herbicide exposure in Vietnam vets. In addition to building software and writing books, they've consulted for companies and spoken at conferences and meetings of software engineers, architects, and project managers.

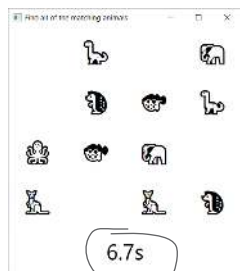
Learn more about them on their website, *Building Better Software*: <https://www.stellman-greene.com>.

Follow @AndrewStellman and @JennyGreene on Twitter

🌐❤️🐱 Jenny and Andrew

# Table of Contents (the summary)

	Intro	xxxi
1	Start building with C#: <i>Building something great...fast!</i>	1
2	Dive into C#: <i>Statements, classes, and code</i>	49
	<i>Unity Lab 1: Explore C# with Unity</i>	87
3	Objects...get oriented: <i>Making code make sense</i>	117
4	Types and references: <i>Getting the reference</i>	155
	<i>Unity Lab 2: Write C# Code for Unity</i>	213
5	Encapsulation: <i>Keep your privates...private</i>	227
6	Inheritance: <i>Your object's family tree</i>	273
	<i>Unity Lab 3: GameObject Instances</i>	343
7	Interfaces, casting, and "is": <i>Making classes keep their promises</i>	355
8	Enums and collections: <i>Organizing your data</i>	405
	<i>Unity Lab 4: User Interfaces</i>	453
9	LINQ and lambdas: <i>Get control of your data</i>	467
10	Reading and writing files: <i>Save the last byte for me!</i>	529
	<i>Unity Lab 5: Raycasting</i>	577
11	Captain Amazing: <i>The Death of the Object</i>	587
12	Exception handling: <i>Putting out fires gets old</i>	623
	<i>Unity Lab 6: Scene Navigation</i>	651
	Downloadable exercise: Animal match boss battle	661
i	Visual Studio for Mac Learner's Guide	663
ii	Code Kata: <i>A learning guide for advanced and impatient readers</i>	725



Let's add some excitement to the game! The time elapsed since the game started will appear at the bottom of the window, constantly going up, and only stopping after the last animal is matched.



# Table of Contents (the real thing)

## Intro

**Your brain on C#.** You're sitting around trying to *learn* something, but your *brain* keeps telling you all that learning *isn't important*. Your brain's saying, "Better leave room for more important things, like which wild animals to avoid and whether nude archery is a bad idea." So how *do* you trick your brain into thinking that your life really depends on learning C#?

Who is this book for?	xxx
We know what you're thinking.	xxxi
And we know what your brain is thinking.	xxxii
Metacognition: thinking about thinking	xxxiii
Here's what WE did	xxxiv
Here's what YOU can do to bend your brain into submission	xxxv
README	xxxvi
The technical review team	xli
Acknowledgments	xli
O'Reilly online learning	xlii



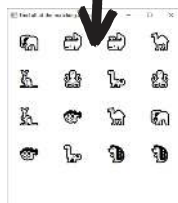


1

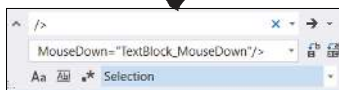
CREATE THE PROJECT



DESIGN THE WINDOW



WRITE C# CODE



HANDLE MOUSE CLICKS



ADD A GAME TIMER

## start building with C#

### Build something great...fast!

#### Want to build great apps...right now?

With C#, you've got a **modern programming language** and a **valuable tool** at your fingertips. And with **Visual Studio**, you've got an **amazing development environment** with highly intuitive features that make coding as easy as possible. Not only is Visual Studio a great tool for writing code, it's also a **really valuable learning tool** for exploring C#. Sound appealing? Turn the page, and let's get coding.

Why you should learn C#	2
Visual Studio is a tool for writing code and exploring C#	3
Create your first project in Visual Studio	4
Let's build a game!	6
Here's how you'll build your game	7
Create a WPF project in Visual Studio	8
Use XAML to design your window	12
Design the window for your game	13
Set the window size and title with XAML properties	14
Add rows and columns to the XAML grid	16
Make the rows and columns equal size	17
Add a TextBlock control to your grid	18
Now you're ready to start writing code for your game	21
Generate a method to set up the game	22
Finish your SetUpGame method	24
Run your program	26
Add your new project to source control	30
The next step to build the game is handling mouse clicks	33
Make your TextBlocks respond to mouse clicks	34
Add the TextBlock_MouseDown code	37
Make the rest of the TextBlocks call the same MouseDown event handler	38
Finish the game by adding a timer	39
Add a timer to your game's code	40
Use the debugger to troubleshoot the exception	42
Add the rest of the code and finish the game	46
Update your code in source control	47

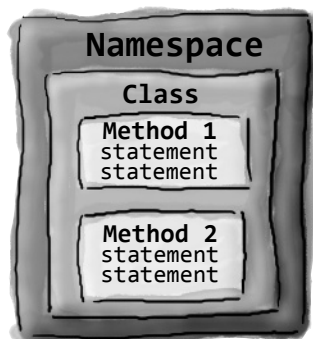
dive into C#

## 2

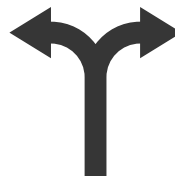
**Statements, classes, and code****You're not just an IDE user. You're a developer.**

You can get a lot of work done using the IDE, but there's only so far it can take you.

Visual Studio is one of the most advanced software development tools ever made, but a **powerful IDE** is only the beginning. It's time to **dig in to C# code**: how it's structured, how it works, and how you can take control of it...because there's no limit to what you can get your apps to do.



Let's take a closer look at the files for a console app	50
Two classes can be in the same namespace (and file!)	52
Statements are the building blocks for your apps	55
Your programs use variables to work with data	56
Generate a new method to work with variables	58
Add code that uses operators to your method	59
Use the debugger to watch your variables change	60
Use operators to work with variables	62
"if" statements make decisions	63
Loops perform an action over and over	64
Use code snippets to help write loops	67
Controls drive the mechanics of your user interfaces	71
Create a WPF app to experiment with controls	72
Add a TextBox control to your app	75
Add C# code to update the TextBlock	78
Add an event handler that only allows number input	79
Add sliders to the bottom row of the grid	83
Add C# code to make the rest of the controls work	84

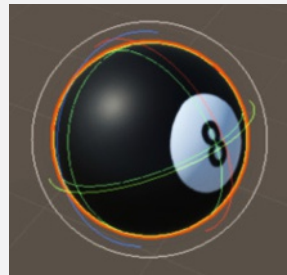
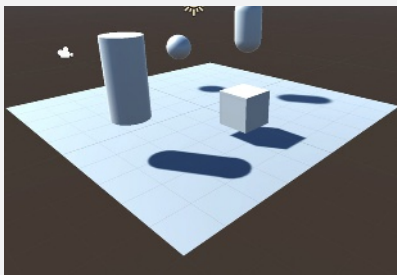


# Unity Lab 1

## Explore C# with Unity

Welcome to your first **Head First C# Unity Lab**. Writing code is a skill, and like any other skill, getting better at it takes **practice and experimentation**. Unity will be a really valuable tool for that. In this lab, you can begin practicing what you've learned about C# in Chapters 1 and 2.

Unity is a powerful tool for game design	88
Download Unity Hub	89
Use Unity Hub to create a new project	90
Take control of the Unity layout	91
Your scene is a 3D environment	92
Unity games are made with GameObjects	93
Use the Move Gizmo to move your GameObjects	94
The Inspector shows your GameObject's components	95
Add a material to your Sphere GameObject	96
Rotate your sphere	99
Get creative!	102



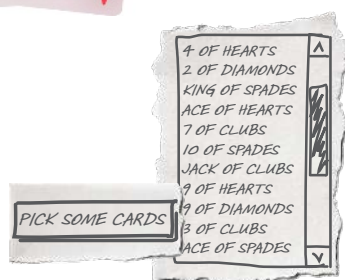
objects...get oriented!

## Making code make sense

# 3

### Every program you write solves a problem.

When you're building a program, it's always a good idea to start by thinking about what *problem* your program's supposed to solve. That's why **objects** are really useful. They let you structure your code based on the problem it's solving so that you can spend your time *thinking about the problem* you need to work on rather than getting bogged down in the mechanics of writing code. When you use objects right—and really put some thought into how you design them—you end up with code that's *intuitive* to write, and easy to read and change.



If code is useful, it gets reused	104
Some methods take parameters and return a value	105
Let's build a program that picks some cards	106
Create your PickRandomCards console app	107
Finish your PickSomeCards method	108
Your finished CardPicker class	110
Ana's working on her next game	113
Build a paper prototype for a classic game	116
Up next: build a WPF version of your card picking app	118
A StackPanel is a container that stacks other controls	119
Reuse your CardPicker class in a new WPF app	120
Use a Grid and StackPanel to lay out the main window	121
Lay out your Card Picker desktop app's window	122
Ana can use objects to solve her problem	126
You use a class to build an object	127
When you create a new object from a class, it's called an instance of that class	128
A better solution for Ana... brought to you by objects	129
An instance uses fields to keep track of things	133
Thanks for the memory	136
What's on your program's mind	137
Sometimes code can be difficult to read	138
Use intuitive class and method names	140
Build a class to work with some guys	146
There's an easier way to initialize objects with C#	148
Use the C# Interactive window to run C# code	154

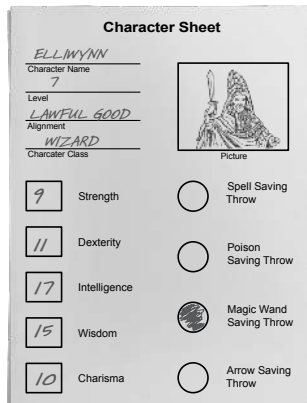
## types and references

### Getting the reference

# 4

**What would your apps be without data?** Think about it for a minute.

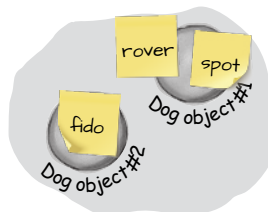
Without data, your programs are...well, it's actually hard to imagine writing code without data. You need **information** from your users, and you use that to look up or produce new information to give back to them. In fact, almost everything you do in programming involves **working with data** in one way or another. In this chapter, you'll learn the ins and outs of C#'s **data types** and **references**, see how to work with data in your program, and even learn a few more things about **objects** (*guess what...objects are data, too!*).



**Creating a reference is like writing a name on a sticky note and sticking it to the object. You're using it to label an object so you can refer to it later.**



Owen could use our help!	156
Character sheets store different types of data on paper	157
A variable's type determines what kind of data it can store	158
C# has several types for storing integers	159
Let's talk about strings	161
A literal is a value written directly into your code	162
A variable is like a data to-go cup	165
Other types come in different sizes, too	166
10 pounds of data in a 5-pound bag	167
Casting lets you copy values that C# can't automatically convert to another type	168
C# does some conversion automatically	171
When you call a method, the arguments need to be compatible with the types of the parameters	172
Let's help Owen experiment with ability scores	176
Use the C# compiler to find the problematic line of code	178
Use reference variables to access your objects	186
Multiple references and their side effects	190
Objects use references to talk to each other	198
Arrays hold multiple values	200
Arrays can contain reference variables	201
null means a reference points to nothing	203
Welcome to Sloppy Joe's Budget House o' Discount Sandwiches!	208



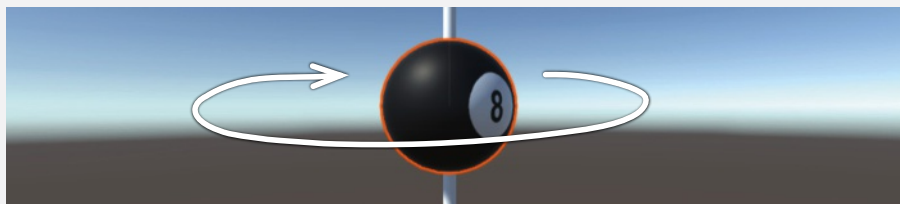


# Unity Lab 2

## Write C# Code for Unity

Unity isn't *just* a powerful, cross-platform engine and editor for building 2D and 3D games and simulations. It's also a **great way to get practice writing C# code**. In this lab, you'll get more practice writing C# code for a project in Unity.

C# scripts add behavior to your GameObjects	214
Add a C# script to your GameObject	215
Write C# code to rotate your sphere	216
Add a breakpoint and debug your game	218
Use the debugger to understand Time.deltaTime	219
Add a cylinder to show where the Y axis is	220
Add fields to your class for the rotation angle and speed	221
Use Debug.DrawRay to explore how 3D vectors work	222
Run the game to see the ray in the Scene view	223
Rotate your ball around a point in the scene	224
Use Unity to take a closer look at rotation and vectors	225
Get creative!	226



encapsulation

5

**Keep your privates...private**

**Ever wished for a little more privacy?**

Sometimes your objects feel the same way. Just like you don't want anybody you don't trust reading your journal or paging through your bank statements, good objects don't let *other* objects go poking around their fields. In this chapter, you're going to learn about the power of **encapsulation**, a way of programming that helps you make code that's flexible, easy to use, and difficult to misuse. You'll **make your objects' data private**, and add **properties** to protect how that data is accessed.



SwordDamage
Roll
MagicMultiplier
FlamingDamage
Damage
CalculateDamage
SetMagic
SetFlaming



Let's help Owen roll for damage	228
Create a console app to calculate damage	229
Design the XAML for a WPF version of the damage calculator	231
The code-behind for the WPF damage calculator	232
Tabletop talk (or maybe...dice discussion?)	233
Let's try to fix that bug	234
Use Debug.WriteLine to print diagnostic information	235
It's easy to accidentally misuse your objects	238
Encapsulation means keeping some of the data in a class private	239
Use encapsulation to control access to your class's methods and fields	240
But is the RealName field REALLY protected?	241
Private fields and methods can only be accessed from instances of the same class	242
Why encapsulation? Think of an object as a black box...	247
Let's use encapsulation to improve the SwordDamage class	251
Encapsulation keeps your data safe	252
Write a console app to test the PaintballGun class	253
Properties make encapsulation easier	254
Modify your Main method to use the Bullets property	255
Auto-implemented properties simplify your code	256
Use a private setter to create a read-only property	257
What if we want to change the magazine size?	258
Use a constructor with parameters to initialize properties	259
Specify arguments when you use the new keyword	260



**RealName: "Herb Jones"**  
**Alias: "Dash Martin"**  
**Password: "the crow flies at midnight"**

## inheritance

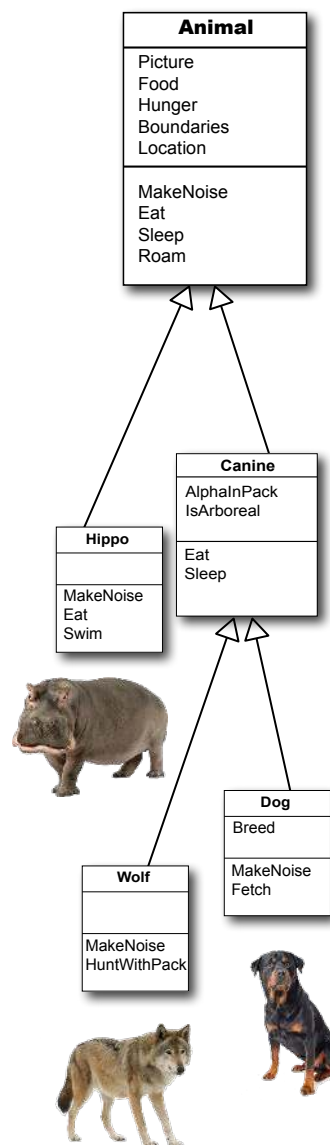
**Your object's family tree**

## 6

**Sometimes you *DO* want to be just like your parents.**

Ever run across a class that *almost* does exactly what you want *your* class to do?

Found yourself thinking that if you could just *change a few things*, that class would be perfect? With **inheritance**, you can **extend** an existing class so your new class gets all of its behavior—with the **flexibility** to make changes to that behavior so you can tailor it however you want. Inheritance is one of the most powerful concepts and techniques in the C# language: with it you'll **avoid duplicate code**, **model the real world** more closely, and end up with apps that are **easier to maintain** and **less prone to bugs**.



Calculate damage for MORE weapons	274
Use a switch statement to match several candidates	275
One more thing...can we calculate damage for a dagger? and a mace?	277
When your classes use inheritance, you only need to write your code once	278
Build up your class model by starting general and getting more specific	279
How would you design a zoo simulator?	280
Every subclass extends its base class	285
Use a colon to extend a base class	290
A subclass can override methods to change or replace members it inherited	292
Some members are only implemented in a subclass	297
Use the debugger to understand how overriding works	298
Build an app to explore virtual and override	300
A subclass can hide methods in the base class	302
Use the override and virtual keywords to inherit behavior	304
When a base class has a constructor, your subclass needs to call it	307
It's time to finish the job for Owen	309
Build a beehive management system	316
The Queen class: how she manages the worker bees	318
The UI: add the XAML for the main window	319
Feedback drives your Beehive Management game	328
Some classes should never be instantiated	332
An abstract class is an intentionally incomplete class	334
Like we said, some classes should never be instantiated	336
An abstract method doesn't have a body	337
Abstract properties work just like abstract methods	338

# Unity Lab 3

## GameObject Instances

C# is an object-oriented language, and since these Head First C# Unity Labs are all **about getting practice writing C# code**, it makes sense that these labs will focus on creating objects.

Let's build a game in Unity!	344
Create a new material inside the Materials folder	345
Spawn a billiard ball at a random point in the scene	346
Use the debugger to understand Random.value	347
Turn your GameObject into a prefab	348
Create a script to control the game	349
Attach the script to the Main Camera	350
Press Play to run your code	351
Use the Inspector to work with GameObject instances	352
Use physics to keep balls from overlapping	353
Get creative!	354

