 [Related books from O’Reilly](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface01.html#preface)

1. [Other books in O’Reilly’s Head First series](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface01.html#id2418)

 [Table of Contents (the real thing)](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface02.html?_gl=1*1howe0w*_ga*Njc2MjM2MjM1LjE3Mjk4NDc5NTQ.*_ga_092EL089CH*MTcyOTg0Nzk1My4xLjEuMTcyOTg0Nzk5NC4xOS4wLjA.#table_of_contents_left_parenthesisthe_r)



 [How to use this Book Intro](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#how_to_use_this_book_intro)

1. [Who is this book for?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#who_is_this_book_forquestion_mark)
   1. [Who should probably back away from this book?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#who_should_probably_back_away_from_this)
2. [We know what you’re thinking](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#we_know_what_youapostrophere_thinking)
3. [We know what your brain is thinking](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#we_know_what_your_brain_is_thinking)
4. [Metacognition: thinking about thinking](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#metacognition_thinking_about_thinking)
5. [Here’s what WE did:](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#hereapostrophes_what_we_did)
6. [Here’s what YOU can do to bend your brain into submission](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#hereapostrophes_what_you_can_do_to_bend)
7. [README.md](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#readmedotmd)
8. [The technical review team](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#the_technical_review_team)
9. [“Shoulders of giants”](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#quotation_markshoulders_of_giantsquotati)
10. [Acknowledgments](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#acknowledgments)
    1. [...and a few more folks who have been so incredibly supportive over the years](https://learning.oreilly.com/library/view/head-first-c/9781098141776/preface04.html#hellipand_a_few_more_folks_who_have_been)

 [1. Start Building Apps with C# Build something great...fast!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#start_building_apps_with_chash_build_som)

1. [Learn C#...and learn to become a great developer](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#learn_chashhellipand_learn_to_become_a_g)
   1. [...with a learning system that’s effective and fun](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#hellipwith_a_learning_system_thatapostro)
2. [Write code and explore C# with Visual Studio](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#write_code_and_explore_chash_with_visual)
   1. [Visual Studio is your free gateway to C#](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#visual_studio_is_your_free_gateway_to_ch)
3. [Install Visual Studio Community Edition](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#install_visual_studio_community_edition)
4. [Run Visual Studio](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#run_visual_studio)
5. [Create and run your first C# project in Visual Studio](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#create_and_run_your_first_chash_project)
6. [You can use Visual Studio Code with Head First C#](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#you_can_use_visual_studio_code_with_head)
7. [Install the C# extensions](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#install_the_chash_extensions)
   1. [Change the C# debug console setting](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#change_the_chash_debug_console_setting)
8. [Create and run your first project in Visual Studio Code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#create_and_run_your_first_project_in_vis)
9. [Set up Visual Studio Code for the next project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#set_up_visual_studio_code_for_the_next_p)
   1. [First open a folder, then add a project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#first_open_a_foldercomma_then_add_a_proj)
   2. [The Command Palette](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#the_command_palette)
   3. [Install .NET MAUI before reading the rest of this chapter](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#install_dotnet_maui_before_reading_the_r)
   4. [If you’re using Linux, you’ll need an Android device for the .NET MAUI projects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#if_youapostrophere_using_linuxcomma_youa)
10. [Let’s build a game!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#letapostrophes_build_a_gameexclamation_m)
11. [Break up large projects into smaller parts](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#break_up_large_projects_into_smaller_par)
12. [Here’s how you’ll build your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#hereapostrophes_how_youapostrophell_buil)
13. [Create a .NET MAUI project in Visual Studio](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#create_a_dotnet_maui_project_in_visual_s)
14. [Create a .NET MAUI project in Visual Studio Code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#create_a_dotnet_maui_project_in-id0001)
15. [Run your new .NET MAUI app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#run_your_new_dotnet_maui_app)
    1. [Stop your MAUI app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#stop_your_maui_app)
16. [MAUI apps work on all of your devices](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#maui_apps_work_on_all_of_your_devices)
17. [Here’s the page that you’ll build](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#hereapostrophes_the_page_that_youapostro)
18. [Start editing your XAML code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#start_editing_your_xaml_code)
19. [Add the XAML for a Button and a Label](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_the_xaml_for_a_button_and_a_label)
20. [Use a FlexLayout to make a grid of animal buttons](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#use_a_flexlayout_to_make_a_grid_of_anima)
21. [Write C# code to add the animals to the buttons](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#write_chash_code_to_add_the_animals_to_t)
22. [Start editing the PlayAgainButton event handler method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#start_editing_the_playagainbutton_event)
    1. [Add a C# statement to the event handler method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_a_chash_statement_to_the_event_handl)
23. [Add more statements to your event handler](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_more_statements_to_your_event_handle)
24. [Add animals to your buttons](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_animals_to_your_buttons)
25. [Run your app!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#run_your_appexclamation_mark)
    1. [Here are just a few things Git can do for you](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#here_are_just_a_few_things_git_can_do_fo)
26. [Visual Studio makes it easy to use Git](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#visual_studio_makes_it_easy_to_use_git)
27. [Add C# code to handle mouse clicks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_chash_code_to_handle_mouse_clicks)
28. [Enter the code for the event handler](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#enter_the_code_for_the_event_handler)
29. [Run your app and find all the pairs](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#run_your_app_and_find_all_the_pairs)
    1. [Uh-oh—there’s a bug in your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#uh_ohem_dashthereapostrophes_a_bug_in_yo)
    2. [When you find a bug, you need to sleuth it out](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#when_you_find_a_bugcomma_you_need_to_sle)
30. [Finish the game by adding a timer](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#finish_the_game_by_adding_a_timer)
31. [Add a timer to your game’s code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_a_timer_to_your_gameapostrophes_code)
32. [Finish the code for your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#finish_the_code_for_your_game)
    1. [Add a field to hold the time elapsed](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#add_a_field_to_hold_the_time_elapsed)
    2. [Finish your TimerTick method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#finish_your_timertick_method)
33. [Even better ifs...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch01.html#even_better_ifsdotdot)

 [2. Variables, Statements, and Methods Dive into C# code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#variablescomma_statementscomma_and_metho)

1. [Take a closer look at the files in your console app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#take_a_closer_look_at_the_files_in_your)
   1. [A statement performs one single action](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#a_statement_performs_one_single_action)
2. [Statements are the building blocks for your apps](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#statements_are_the_building_blocks_for_y)
3. [Statements live inside methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#statements_live_inside_methods)
   1. [Methods do something](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#methods_do_something)
   2. [Methods help you organize your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#methods_help_you_organize_your_code)
4. [Your methods use variables to work with data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#your_methods_use_variables_to_work_with)
   1. [Declare your variables](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#declare_your_variables)
   2. [Variables vary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#variables_vary)
   3. [You need to assign values to variables before you use them](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#you_need_to_assign_values_to_variables_b)
   4. [A few useful types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#a_few_useful_types)
5. [Generate a new method to work with variables](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#generate_a_new_method_to_work_with_varia)
6. [Add code that uses operators to your method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#add_code_that_uses_operators_to_your_met)
7. [Use the debugger to watch your variables change](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#use_the_debugger_to_watch_your_variables)
8. [Use code snippets to help write loops](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#use_code_snippets_to_help_write_loops)
9. [Use operators to work with variables](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#use_operators_to_work_with_variables)
10. [if statements make decisions](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#if_statements_make_decisions)
    1. [if/else statements also do something if a condition isn’t true](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#ifsoliduselse_statements_also_do_somethi)
11. [Loops perform an action over and over](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#loops_perform_an_action_over_and_over)
    1. [while loops keep looping statements while a condition is true](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#while_loops_keep_looping_statements_whil)
    2. [do/while loops run the statements then check the condition](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#dosoliduswhile_loops_run_the_statements)
    3. [for loops run a statement after each loop](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#for_loops_run_a_statement_after_each_loo)
12. [Some useful things to keep in mind about C# code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#some_useful_things_to_keep_in_mind_about)
13. [Controls drive the mechanics of your user interfaces](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#controls_drive_the_mechanics_of_your_use)
    1. [Meet some of the controls you’ll use in this book](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#meet_some_of_the_controls_youapostrophel)
14. [Other controls you’ll use in this book](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#other_controls_youapostrophell_use_in_th)
15. [Build a .NET MAUI app to experiment with controls](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#build_a_dotnet_maui_app_to_experiment_wi)
16. [Create a new app to experiment with controls](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#create_a_new_app_to_experiment_with_cont)
17. [Explore your new MAUI app and figure out how it works](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#explore_your_new_maui_app_and_figure_out)
18. [Add an Entry control to your app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#add_an_entry_control_to_your_app)
19. [Add properties to your Entry control](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#add_properties_to_your_entry_control)
20. [Make your Entry control update a Label control](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#make_your_entry_control_update_a_label_c)
21. [Combine horizontal and vertical stack layouts](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#combine_horizontal_and_vertical_stack_la)
    1. [You’ll nest one Layout inside another](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#youapostrophell_nest_one_layout_inside_a)
22. [Add a Picker control to display a list of choices](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch02.html#add_a_picker_control_to_display_a_list_o)

 [Unity Lab #1 Explore C# with Unity](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#unity_lab_hash1_explore_chash_with_unity)

1. [Unity is a powerful tool for game design](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#unity_is_a_powerful_tool_for_game_design)
2. [Download Unity Hub](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#download_unity_hub)
3. [Use Unity Hub to create a new project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#use_unity_hub_to_create_a_new_project)
   1. [Work with your project in the Unity editor](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#work_with_your_project_in_the_unity_edit)
4. [Take control of the Unity layout](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#take_control_of_the_unity_layout)
   1. [Choose the Wide layout to match our screenshots](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#choose_the_wide_layout_to_match_our_scre)
   2. [Set up Unity to work with Visual Studio](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#set_up_unity_to_work_with_visual_studio)
5. [Your scene is a 3D environment](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#your_scene_is_a_3d_environment)
6. [Unity games are made with GameObjects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#unity_games_are_made_with_gameobjects)
7. [Use the Move Gizmo to move your GameObjects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#use_the_move_gizmo_to_move_your_gameobje)
8. [The Inspector shows your GameObject’s components](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#the_inspector_shows_your_gameobjectapost)
9. [Add a material to your Sphere GameObject](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#add_a_material_to_your_sphere_gameobject)
10. [Rotate your sphere](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#rotate_your_sphere)
    1. [Move the Scene view camera with the View Tool and Scene Gizmo](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#move_the_scene_view_camera_with_the_view)
    2. [Get creative!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword01.html#get_creativeexclamation_mark)

 [3. Namespaces and Classes Organizing your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#namespaces_and_classes_organizing_your_c)

1. [Classes help you organize your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#classes_help_you_organize_your_code)
2. [If code is useful, classes can help you reuse it](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#if_code_is_usefulcomma_classes_can_help)
3. [Some methods take parameters and return a value](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#some_methods_take_parameters_and_return)
4. [Visual Studio helps you explore parameters and return values](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#visual_studio_helps_you_explore_paramete)
5. [Let’s build an app that picks random cards](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#letapostrophes_build_an_app_that_picks_r)
6. [You’ll use an array to store the cards](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#youapostrophell_use_an_array_to_store_th)
7. [Create an app with a Main method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#create_an_app_with_a_main_method)
   1. [Create your PickRandomCards app without top-level statements](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#create_your_pickrandomcards_app_without)
8. [Add a class called CardPicker to your app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#add_a_class_called_cardpicker_to_your_ap)
9. [Use Quick Actions to remove unnecessary using lines](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#use_quick_actions_to_remove_unnecessary)
10. [Convert between namespace styles](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#convert_between_namespace_styles)
11. [Use the new keyword to create an array of strings](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#use_the_new_keyword_to_create_an_array_o)
    1. [Here’s the code for your finished CardPicker class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#hereapostrophes_the_code_for_your_finish)
12. [Ana’s working on her next game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#anaapostrophes_working_on_her_next_game)
13. [Ana’s game is evolving...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#anaapostrophes_game_is_evolvingdotdot)
    1. [...so how can Ana make things easier for herself?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#hellipso_how_can_ana_make_things_easier)
14. [Build a paper prototype for a classic game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#build_a_paper_prototype_for_a_classic_ga)
15. [Build a MAUI version of your random card app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#build_a_maui_version_of_your_random_card)
    1. [Make your app accessible!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#make_your_app_accessibleexclamation_mark)
16. [Make your MAUI app pick random cards](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#make_your_maui_app_pick_random_cards)
17. [Reuse your CardPicker class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#reuse_your_cardpicker_class)
18. [Add a using directive to use code in another namespace](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#add_a_using_directive_to_use_code_in_ano)
19. [Ana’s prototypes look great...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#anaapostrophes_prototypes_look_greatdotd)
    1. [...but what if she wants more than one enemy?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#hellipbut_what_if_she_wants_more_than_on)
20. [Ana can use objects to solve her problem](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#ana_can_use_objects_to_solve_her_problem)
21. [You use a class to build an object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#you_use_a_class_to_build_an_object)
    1. [An object gets its methods from its class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#an_object_gets_its_methods_from_its_clas)
22. [When you create a new object from a class, it’s called an instance of that class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#when_you_create_a_new_object_from_a_clas)
23. [A better solution for Ana...brought to you by objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#a_better_solution_for_anahellipbrought_t)
    1. [Theory and practice](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#theory_and_practice)
24. [An instance uses fields to keep track of things](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#an_instance_uses_fields_to_keep_track_of)
    1. [Methods are what an object does. Fields are what an object knows.](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#methods_are_what_an_object_doesdot_field)
25. [Thanks for the memory](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#thanks_for_the_memory)
26. [What’s on your app’s mind](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#whatapostrophes_on_your_appapostrophes_m)
27. [Sometimes code can be difficult to read](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#sometimes_code_can_be_difficult_to_read)
    1. [Extremely compact code can be especially problematic](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#extremely_compact_code_can_be_especially)
28. [Most code doesn’t come with a manual](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#most_code_doesnapostrophet_come_with_a_m)
29. [Use intuitive class and method names](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#use_intuitive_class_and_method_names)
30. [Build a class to work with some guys](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#build_a_class_to_work_with_some_guys)
31. [There’s an easier way to initialize objects with C#](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#thereapostrophes_an_easier_way_to_initia)
32. [Use the C# Interactive window or csi to run C# code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch03.html#use_the_chash_interactive_window_or_csi)

 [4. Data, Types, Objects, and References Managing your app’s data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#datacomma_typescomma_objectscomma_and_re)

1. [Owen could use our help!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#owen_could_use_our_helpexclamation_mark)
   1. [Storytelling, fantasy, and mechanics](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#storytellingcomma_fantasycomma_and_mecha)
2. [Character sheets store different types of data on paper](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#character_sheets_store_different_types_o)
3. [A variable’s type determines what kind of data it can store](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#a_variableapostrophes_type_determines_wh)
4. [C# has several types for storing integers](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#chash_has_several_types_for_storing_inte)
5. [Types for storing really HUGE and really tiny numbers](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#types_for_storing_really_huge_and_really)
6. [Let’s talk about strings](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#letapostrophes_talk_about_strings)
7. [A literal is a value written directly into your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#a_literal_is_a_value_written_directly_in)
   1. [Use suffixes to give your literals types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_suffixes_to_give_your_literals_types)
8. [A variable is like a data to-go cup](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#a_variable_is_like_a_data_to_go_cup)
   1. [Use the Convert class to explore bits and bytes](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_the_convert_class_to_explore_bits_an)
9. [Other types come in different sizes too](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#other_types_come_in_different_sizes_too)
10. [10 pounds of data in a 5-pound bag](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#ten_pounds_of_data_in_a_5_pound_bag)
11. [Casting lets you copy values that C# can’t automatically convert to another type](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#casting_lets_you_copy_values_that_chash)
    1. [So what happened?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#so_what_happenedquestion_mark)
    2. [When you cast a value that’s too big, C# adjusts it to fit its new container](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#when_you_cast_a_value_thatapostrophes_to)
12. [C# does some conversions automatically](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#chash_does_some_conversions_automaticall)
13. [When you call a method, the arguments need to be compatible with the types of the parameters](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#when_you_call_a_methodcomma_the_argument)
14. [Owen is constantly improving his game...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#owen_is_constantly_improving_his_gamedot)
    1. [...but the trial and error can be time-consuming](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#hellipbut_the_trial_and_error_can_be_tim)
15. [Let’s help Owen experiment with ability scores](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#letapostrophes_help_owen_experiment_with)
16. [Fix the compiler error by adding a cast](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#fix_the_compiler_error_by_adding_a_cast)
    1. [Add a cast to get the AbilityScoreCalculator class to compile...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#add_a_cast_to_get_the_abilityscorecalcul)
    2. [...but there’s still a bug!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#hellipbut_thereapostrophes_still_a_bugex)
    3. [Now we can finally fix Owen’s bug—and get the REAL Sharpen answer](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#now_we_can_finally_fix_owenapostrophes_b)
17. [Use reference variables to access your objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_reference_variables_to_access_your_o)
18. [References are like sticky notes for your objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#references_are_like_sticky_notes_for_you)
19. [If there aren’t any more references, your object gets garbage-collected](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#if_there_arenapostrophet_any_more_refere)
20. [Multiple references and their side effects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#multiple_references_and_their_side_effec)
21. [Two references mean TWO variables that can change the same object’s data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#two_references_mean_two_variables_that_c)
22. [Objects use references to talk to each other](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#objects_use_references_to_talk_to_each_o)
23. [Arrays hold multiple values](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#arrays_hold_multiple_values)
    1. [Use each element in an array like it’s a normal variable](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_each_element_in_an_array_like_itapos)
24. [Arrays can contain reference variables](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#arrays_can_contain_reference_variables)
25. [null means a reference points to nothing](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#null_means_a_reference_points_to_nothing)
    1. [Console.ReadLine returns a null when there are no lines available](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#consoledotreadline_returns_a_null_when_t)
26. [Use the string? type when a string might be null](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_the_stringquestion_mark_type_when_a)
    1. [int.TryParse takes a string? parameter](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#intdottryparse_takes_a_stringquestion_ma)
27. [Welcome to Sloppy Joe’s Budget House o’ Discount Sandwiches!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#welcome_to_sloppy_joeapostrophes_budget)
28. [Sloppy Joe’s menu app uses a Grid layout](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#sloppy_joeapostrophes_menu_app_uses_a_gr)
29. [Grid controls](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#grid_controls)
    1. [Use Grid properties to put a control in a cell](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_grid_properties_to_put_a_control_in)
30. [Define the rows and columns for a Grid](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#define_the_rows_and_columns_for_a_grid)
31. [Create the Sloppy Joe’s menu app and set up the grid](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#create_the_sloppy_joeapostrophes_menu_ap)
    1. [Here’s the XAML for the app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#hereapostrophes_the_xaml_for_the_app)
32. [The C# code for the main page](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#the_chash_code_for_the_main_page)
33. [Can we make the app more accessible?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#can_we_make_the_app_more_accessiblequest)
    1. [Set the main header so the screen reader narrates it](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#set_the_main_header_so_the_screen_reader)
    2. [Try setting the item1 label’s SemanticProperties.Description instead](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#try_setting_the_item1_labelapostrophes_s)
34. [Use the SetValue method to change a control’s semantic properties](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch04.html#use_the_setvalue_method_to_change_a_cont)

 [Unity Lab #2 Write C# Code for Unity](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#unity_lab_hash2_write_chash_code_for_uni)

1. [C# scripts add behavior to your GameObjects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#chash_scripts_add_behavior_to_your_gameo)
2. [Add a C# script to your GameObject](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#add_a_chash_script_to_your_gameobject)
3. [Write C# code to rotate your sphere](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#write_chash_code_to_rotate_your_sphere)
4. [Add a breakpoint and debug your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#add_a_breakpoint_and_debug_your_game)
   1. [Use a hit count to skip frames](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#use_a_hit_count_to_skip_frames)
5. [Use the debugger to understand Time.deltaTime](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#use_the_debugger_to_understand_timedotde)
6. [Add a cylinder to show where the Y axis is](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#add_a_cylinder_to_show_where_the_y_axis)
7. [Add fields to your class for the rotation angle and speed](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#add_fields_to_your_class_for_the_rotatio)
8. [Use Debug.DrawRay to explore how 3D vectors work](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#use_debugdotdrawray_to_explore_how_3d_ve)
   1. [Use Unity to visualize vectors in 3D](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#use_unity_to_visualize_vectors_in_3d)
9. [Run the game to see the ray in the Scene view](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#run_the_game_to_see_the_ray_in_the_scene)
   1. [Add a duration to the ray so it leaves a trail](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#add_a_duration_to_the_ray_so_it_leaves_a)
10. [Rotate your ball around a point in the scene](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#rotate_your_ball_around_a_point_in_the_s)
11. [Use Unity to take a closer look at rotation and vectors](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#use_unity_to_take_a_closer_look_at_rotat)
12. [Get creative!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword02.html#get_creativeexclamation_mark-id0001)

 [5. Encapsulation How objects keep their Secrets](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#encapsulation_how_objects_keep_their_sec)

1. [Let’s help Owen roll for damage](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#letapostrophes_help_owen_roll_for_damage)
2. [Create a console app to calculate damage](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#create_a_console_app_to_calculate_damage)
3. [Design a MAUI version of the damage calculator app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#design_a_maui_version_of_the_damage_calc)
   1. [How your damage calculator app will work](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#how_your_damage_calculator_app_will_work)
4. [Tabletop talk (or maybe...dice discussion?)](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#tabletop_talk_left_parenthesisor_maybehe)
5. [Let’s try to fix that bug](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#letapostrophes_try_to_fix_that_bug)
   1. [Oops! It’s still not working](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#oopsexclamation_mark_itapostrophes_still)
6. [Use Debug.WriteLine to print diagnostic information](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#use_debugdotwriteline_to_print_diagnosti)
7. [It’s easy to accidentally misuse your objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#itapostrophes_easy_to_accidentally_misus)
8. [Encapsulation means keeping some data in a class private](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#encapsulation_means_keeping_some_data_in)
   1. [When in doubt, make it private](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#when_in_doubtcomma_make_it_private)
9. [Use encapsulation to control access to your class’s methods and fields](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#use_encapsulation_to_control_access_to_y)
10. [But is the RealName field REALLY protected?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#but_is_the_realname_field_really_protect)
11. [Private fields and methods can only be accessed from instances of the same class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#private_fields_and_methods_can_only_be_a)
12. [Why encapsulation? Think of an object as an opaque box...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#why_encapsulationquestion_mark_think_of)
    1. [Encapsulation makes your classes...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#encapsulation_makes_your_classesdotdot)
13. [Let’s use encapsulation to improve the SwordDamage class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#letapostrophes_use_encapsulation_to_impr)
    1. [Is every member of the SwordDamage class public?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#is_every_member_of_the_sworddamage_class)
    2. [Are fields or methods being misused?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#are_fields_or_methods_being_misusedquest)
    3. [Is there calculation required after setting a field?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#is_there_calculation_required_after_sett)
    4. [So what fields and methods really need to be public?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#so_what_fields_and_methods_really_need_t)
14. [Encapsulation keeps your data safe](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#encapsulation_keeps_your_data_safe)
    1. [Let’s use encapsulation in a class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#letapostrophes_use_encapsulation_in_a_cl)
15. [Write a console app to test the PaintballGun class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#write_a_console_app_to_test_the_paintbal)
    1. [Our class is well-encapsulated, but...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#our_class_is_well_encapsulatedcomma_butd)
16. [Properties make encapsulation easier](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#properties_make_encapsulation_easier)
    1. [Replace the GetBalls and SetBalls methods with a property](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#replace_the_getballs_and_setballs_method)
17. [Modify your top-level statements to use the Balls property](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#modify_your_top_level_statements_to_use)
    1. [Debug your PaintballGun class to understand how the property works](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#debug_your_paintballgun_class_to_underst)
18. [Auto-implemented properties simplify your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#auto_implemented_properties_simplify_you)
    1. [Use the prop snippet to create an auto-implemented property](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#use_the_prop_snippet_to_create_an_auto_i)
19. [Use a private setter to create a read-only property](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#use_a_private_setter_to_create_a_read_on)
    1. [Make the BallsLoaded setter private](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#make_the_ballsloaded_setter_private)
20. [What if we want to change the magazine size?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#what_if_we_want_to_change_the_magazine_s)
    1. [But there’s a problem...how do we initialize MagazineSize?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#but_thereapostrophes_a_problemhelliphow)
21. [Use a constructor with parameters to initialize properties](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#use_a_constructor_with_parameters_to_ini)
22. [Specify arguments when you use the new keyword](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#specify_arguments_when_you_use_the_new_k)
23. [Initialize fields and properties inline or in the constructor](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#initialize_fields_and_properties_inline)
24. [Make the screen reader announce each roll](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#make_the_screen_reader_announce_each_rol)
25. [A few useful facts about methods and properties](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch05.html#a_few_useful_facts_about_methods_and_pro)

 [6. Inheritance Your Object’s Family Tree](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#inheritance_your_objectapostrophes_famil)

1. [Calculate damage for MORE weapons](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#calculate_damage_for_more_weapons)
2. [Use a switch statement to match several candidates](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_a_switch_statement_to_match_several)
3. [One more thing...can we calculate damage for a dagger? And a mace? And a staff? and...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#one_more_thinghellipcan_we_calculate_dam)
4. [When your classes use inheritance, you only need to write your code once](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#when_your_classes_use_inheritancecomma_y)
5. [Build up your class model by starting general and getting more specific](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#build_up_your_class_model_by_starting_ge)
6. [How would you design a zoo simulator?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#how_would_you_design_a_zoo_simulatorques)
7. [Different animals have different behaviors](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#different_animals_have_different_behavio)
8. [Every subclass extends its base class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#every_subclass_extends_its_base_class)
   1. [C# always calls the most specific method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#chash_always_calls_the_most_specific_met)
9. [Any place where you can use a base class, you can use one of its subclasses instead](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#any_place_where_you_can_use_a_base_class)
10. [Use a colon to extend a base class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_a_colon_to_extend_a_base_class)
11. [We know that inheritance adds the base class fields, properties, and methods to the subclass...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#we_know_that_inheritance_adds_the_base_c)
    1. [...but some birds don’t fly!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#hellipbut_some_birds_donapostrophet_flye)
12. [A subclass can override methods to change or replace members it inherited](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#a_subclass_can_override_methods_to_chang)
13. [Some members are only implemented in a subclass](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#some_members_are_only_implemented_in_a_s)
14. [Use the debugger to understand how overriding works](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_the_debugger_to_understand_how_overr)
15. [Build an app to explore virtual and override](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#build_an_app_to_explore_virtual_and_over)
16. [A subclass can hide methods in the base class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#a_subclass_can_hide_methods_in_the_base)
    1. [Hiding methods vs. overriding methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#hiding_methods_vsdot_overriding_methods)
    2. [Use the new keyword when you’re hiding methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_the_new_keyword_when_youapostrophere)
    3. [Use different references to call hidden methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_different_references_to_call_hidden)
17. [Use the override and virtual keywords to inherit behavior](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_the_override_and_virtual_keywords_to)
18. [A subclass can access its base class using the base keyword](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#a_subclass_can_access_its_base_class_usi)
19. [When a base class has a constructor, your subclass needs to call it](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#when_a_base_class_has_a_constructorcomma)
20. [A subclass and base class can have different constructors](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#a_subclass_and_base_class_can_have_diffe)
21. [It’s time to finish the job for Owen](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#itapostrophes_time_to_finish_the_job_for)
    1. [Use the debugger to really understand how these classes work](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#use_the_debugger_to_really_understand_ho)
22. [A class should do one thing](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#a_class_should_do_one_thing)
23. [Build a beehive management system](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#build_a_beehive_management_system)
24. [How the Beehive Management System app works](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#how_the_beehive_management_system_app_wo)
25. [The page uses a grid to lay out the controls for the UI](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_page_uses_a_grid_to_lay_out_the_cont)
    1. [The Bee Jobs box uses a Border with a VerticalStackLayout](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_bee_jobs_box_uses_a_border_with_a_ve)
26. [The Beehive Management System class model](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_beehive_management_system_class_mode)
27. [All bees in the system extend the Bee class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#all_bees_in_the_system_extend_the_bee_cl)
28. [All the constants are in their own static class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#all_the_constants_are_in_their_own_stati)
29. [The worker bees extend the Bee class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_worker_bees_extend_the_bee_class)
30. [The Queen class: how she manages the worker bees](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_queen_class_how_she_manages_the_work)
31. [Here’s the code-behind for MainPage.xaml.cs](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#hereapostrophes_the_code_behind_for_main)
32. [Feedback drives your beehive management game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#feedback_drives_your_beehive_management)
    1. [Workers and honey are in a feedback loop](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#workers_and_honey_are_in_a_feedback_loop)
33. [The Beehive Management System is turn-based... now let’s convert it to real-time](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_beehive_management_system_is_turn_ba)
34. [Some classes should never be instantiated](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#some_classes_should_never_be_instantiate)
35. [An abstract class is an intentionally incomplete class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#an_abstract_class_is_an_intentionally_in)
    1. [Let’s plan a trip to another planet](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#letapostrophes_plan_a_trip_to_another_pl)
36. [Like we said, some classes should never be instantiated](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#like_we_saidcomma_some_classes_should_ne)
    1. [Solution: use an abstract class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#solution_use_an_abstract_class)
37. [An abstract method doesn’t have a body](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#an_abstract_method_doesnapostrophet_have)
38. [Abstract properties work just like abstract methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#abstract_properties_work_just_like_abstr)
39. [The Deadly Diamond of Death](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch06.html#the_deadly_diamond_of_death)

 [Unity Lab #3 GameObject Instances](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#unity_lab_hash3_gameobject_instances)

1. [Let’s build a game in Unity!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#letapostrophes_build_a_game_in_unityexcl)
2. [Create a new material inside the Materials folder](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#create_a_new_material_inside_the_materia)
3. [Spawn a billiard ball at a random point in the scene](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#spawn_a_billiard_ball)
4. [Use the debugger to understand Random.value](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#use_the_debugger_to_understand_randomdot)
5. [Turn your GameObject into a prefab](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#turn_your_gameobject_into_a_prefab)
6. [Create a script to control the game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#create_a_script_to_control_the_game)
7. [Attach the GameController script to the Main Camera](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#attach_the_gamecontroller_script_to_the)
8. [Press Play to run your code](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#press_play_to_run_your_code)
   1. [Watch the live instances in the Hierarchy window](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#watch_the_live_instances_in_the_hierarch)
9. [Use the Inspector to work with GameObject instances](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#use_the_inspector_to_work_with_gameobjec)
10. [Use physics to keep balls from overlapping](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#use_physics_to_keep_balls_from_overlappi)
11. [Get creative!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword03.html#get_creativeexclamation_mark-id0002)

 [7. Interfaces, Casting, and “is” Making classes keep their Promises](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#interfacescomma_castingcomma_and_quotati)

1. [The beehive is under attack!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#the_beehive_is_under_attackexclamation_m)
   1. [HiveDefender needs a DefendHive method because enemies can attack at any time](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#hivedefender_needs_a_defendhive_method_b)
2. [We could use casting to call the DefendHive method…](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#we_could_use_casting_to_call_the_defendh)
   1. […but what if we add more Bee subclasses that can defend?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#hellipbut_what_if_we_add_more_bee_subcla)
3. [An interface defines methods and properties that a class must implement…](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#an_interface_defines_methods_and_propert)
   1. […but there’s no limit to the number of interfaces a class can implement](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#hellipbut_thereapostrophes_no_limit_to_t)
4. [Interfaces let unrelated classes do the same job](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#interfaces_let_unrelated_classes_do_the)
5. [Get a little practice using interfaces](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#get_a_little_practice_using_interfaces)
6. [Fireside Chats](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#fireside_chats)
7. [You can’t instantiate an interface, but you can reference an interface](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#you_canapostrophet_instantiate_an_interf)
   1. [If you try to instantiate an interface, your code won’t build](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#if_you_try_to_instantiate_an_interfaceco)
   2. [Use the interface to reference an object you already have](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_the_interface_to_reference_an_object)
8. [Pool Puzzle](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#pool_puzzle)
9. [Pool Puzzle Solution](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#pool_puzzle_solution)
10. [Interface references are ordinary object references](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#interface_references_are_ordinary_object)
11. [The RoboBee 4000 can do a worker bee’s job without using valuable honey](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#the_robobee_4000_can_do_a_worker_beeapos)
12. [The IWorker’s Job property is a hack](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#the_iworkerapostrophes_job_property_is_a)
13. [Use is to check the type of an object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_is_to_check_the_type_of_an_object)
14. [Use is to access methods in a subclass](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_is_to_access_methods_in_a_subclass)
15. [What if we want different animals to swim or hunt in packs?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#what_if_we_want_different_animals_to_swi)
16. [Use interfaces to work with classes that do the same job](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_interfaces_to_work_with_classes_that)
    1. [Use the “is” keyword to check if the Animal is a swimmer or pack hunter](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_the_quotation_markisquotation_mark_k)
17. [Safely navigate your class hierarchy with is](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#safely_navigate_your_class_hierarchy_wit)
18. [C# has another tool for safe type conversion: the as keyword](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#chash_has_another_tool_for_safe_type_con)
19. [Use upcasting and downcasting to move up and down a class hierarchy](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_upcasting_and_downcasting_to_move_up)
20. [A quick example of upcasting](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#a_quick_example_of_upcasting)
21. [Upcasting turns your CoffeeMaker into an Appliance](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#upcasting_turns_your_coffeemaker_into_an)
22. [Downcasting turns your Appliance back into a CoffeeMaker](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#downcasting_turns_your_appliance_back_in)
23. [Upcasting and downcasting work with interfaces too](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#upcasting_and_downcasting_work_with_inte)
24. [Interfaces can inherit from other interfaces](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#interfaces_can_inherit_from_other_interf)
25. [Interfaces can have static members](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#interfaces_can_have_static_members)
26. [Default implementations give bodies to interface methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#default_implementations_give_bodies_to_i)
27. [Add a ScareAdults method with a default implementation](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#add_a_scareadults_method_with_a_default)
28. [Data binding updates MAUI controls automatically](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#data_binding_updates_maui_controls_autom)
29. [Add data binding to the default MAUI app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#add_data_binding_to_the_default_maui_app)
30. [Make Moods implement the INotifyPropertyChanged interface](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#make_moods_implement_the_inotifyproperty)
31. [Use the PropertyChanged event to make data binding work](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#use_the_propertychanged_event_to_make_da)
32. [Polymorphism means that one object can take many different forms](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#polymorphism_means_that_one_object_can_t)
    1. [Keep your eyes open for polymorphism!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#keep_your_eyes_open_for_polymorphismexcl)
    2. [The four core principles of object-oriented programming](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch07.html#the_four_core_principles_of_object_orien)

 [8. Enums and Collections Organizing your data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#enums_and_collections_organizing_your_da)

1. [If a constructor just sets fields, use a primary constructor instead](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#if_a_constructor_just_sets_fieldscomma_u)
2. [A primary constructor can extend a base constructor](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#a_primary_constructor_can_extend_a_base)
   1. [Overloaded constructors use this to call the primary constructor](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#overloaded_constructors_use_this_to_call)
3. [Strings don’t always work for storing categories of data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#strings_donapostrophet_always_work_for_s)
4. [Enums let you work with a set of valid values](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#enums_let_you_work_with_a_set_of_valid_v)
   1. [An enum defines a new type](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#an_enum_defines_a_new_type)
5. [Enums let you represent numbers with names](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#enums_let_you_represent_numbers_with_nam)
6. [We could use an array to create a deck of cards…](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#we_could_use_an_array_to_create_a_deck_o)
   1. […but what if you wanted to do more?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#hellipbut_what_if_you_wanted_to_do_moreq)
7. [Arrays can be annoying to work with](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#arrays_can_be_annoying_to_work_with)
8. [Lists make it easy to store collections of…anything](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#lists_make_it_easy_to_store_collections)
9. [Lists are more flexible than arrays](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#lists_are_more_flexible_than_arrays)
10. [Let’s build an app to store shoes](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#letapostrophes_build_an_app_to_store_sho)
11. [Generic collections can store any type](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#generic_collections_can_store_any_type)
    1. [Generic lists are declared using <angle brackets>](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#generic_lists_are_declared_using_less_th)
12. [You can use collection expressions to create Lists](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#you_can_use_collection_expressions_to_cr)
13. [Let’s create a List of Ducks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#letapostrophes_create_a_list_of_ducks)
    1. [Use a collection expression to create a List of Ducks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_a_collection_expression_to_create_a)
14. [Sorting lists can be tricky](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#sorting_lists_can_be_tricky)
    1. [Lists know how to sort themselves](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#lists_know_how_to_sort_themselves)
15. [IComparable<Duck> helps your List sort its Ducks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#icomparableless_thanduckgreater_than_hel)
    1. [An object’s CompareTo method compares it to another object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#an_objectapostrophes_compareto_method_co)
16. [Use IComparer to tell your List how to sort](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_icomparer_to_tell_your_list_how_to_s)
    1. [Add an IComparer to your project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#add_an_icomparer_to_your_project)
17. [Create an instance of your comparer object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#create_an_instance_of_your_comparer_obje)
    1. [Multiple IComparer implementations, multiple ways to sort your objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#multiple_icomparer_implementationscomma)
18. [Comparers can do complex comparisons](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#comparers_can_do_complex_comparisons)
19. [Overriding a ToString method lets an object describe itself](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#overriding_a_tostring_method_lets_an_obj)
    1. [Override the ToString method to see your Ducks in the IDE](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#override_the_tostring_method_to_see_your)
20. [Update your foreach loops to let your Ducks and Cards write themselves to the console](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#update_your_foreach_loops_to_let_your_du)
    1. [Add a ToString method to your Card object too](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#add_a_tostring_method_to_your_card_objec)
21. [You can upcast an entire list using IEnumerable<T>](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#you_can_upcast_an_entire_list_using_ienu)
22. [Use a Dictionary to store keys and values](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_a_dictionary_to_store_keys_and_value)
    1. [How to use a Dictionary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#how_to_use_a_dictionary)
23. [The Dictionary functionality rundown](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#the_dictionary_functionality_rundown)
    1. [Your key and value can be different types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#your_key_and_value_can_be_different_type)
24. [Write an app that uses a Dictionary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#write_an_app_that_uses_a_dictionary)
    1. [Use a collection initializer to create the Dictionary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_a_collection_initializer_to_create_t)
25. [And yet MORE collection types…](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#and_yet_more_collection_typeshellip)
    1. [Generic .NET collections implement IEnumerable](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#generic_dotnet_collections_implement_ien)
26. [A queue is FIFO—first in, first out](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#a_queue_is_fifoem_dashfirst_incomma_firs)
27. [A stack is LIFO—last in, first out](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#a_stack_is_lifoem_dashlast_incomma_first)
28. [CollectionView is a MAUI control built for displaying collections](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#collectionview_is_a_maui_control_built_f)
29. [ObservableCollection is a collection made for data binding](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#observablecollection_is_a_collection_mad)
30. [Add your Card class to the project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#add_your_card_class_to_the_project)
31. [Use XAML to instantiate your objects for data binding](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_xaml_to_instantiate_your_objects_for)
32. [Modify your app to use a resource Dictionary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#modify_your_app_to_use_a_resource_dictio)
33. [Modify the event handlers to use the resource Dictionary](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#modify_the_event_handlers_to_use_the_res)
34. [Use what you’ve learned to build an app with two decks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch08.html#use_what_youapostropheve_learned_to_buil)

 [Unity Lab #4 User Interfaces](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#unity_lab_hash4_user_interfaces)

1. [Add a score that goes up when the player clicks a ball](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#add_a_score_that_goes_up_when_the_player)
2. [Add two different modes to your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#add_two_different_modes_to_your_game)
3. [Add game mode to your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#add_game_mode_to_your_game)
4. [Add a UI to your game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#add_a_ui_to_your_game)
   1. [Use the 2D view to work with the Canvas](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#use_the_2d_view_to_work_with_the_canvas)
5. [Set up the Text that will display the score in the UI](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#set_up_the_text_that_will_display_the_sc)
6. [Add a Button that calls a method to start the game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#add_a_button_that_calls_a_method_to_star)
7. [Make the Play Again button and Score Text work](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#make_the_play_again_button_and_score_tex)
8. [Finish the code for the game](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#finish_the_code_for_the_game)
9. [Get creative!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword04.html#get_creativeexclamation_mark-id0003)

 [9. LINQ and Lambdas Get Control of your data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_and_lambdas_get_control_of_your_dat)

1. [Jimmy’s a Captain Amazing superfan...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#jimmyapostrophes_a_captain_amazing_super)
2. [...but his collection’s all over the place](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#hellipbut_his_collectionapostrophes_all)
3. [Use LINQ to query your collections](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_linq_to_query_your_collections)
4. [LINQ works with any sequence](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_works_with_any_sequence)
   1. [LINQ methods enumerate your sequences—just like foreach loops](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_methods_enumerate_your_sequencesem)
5. [LINQ’s query syntax](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linqapostrophes_query_syntax)
   1. [LINQ queries are built from clauses](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_queries_are_built_from_clauses)
6. [LINQ works with objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_works_with_objects)
7. [Use a LINQ query to finish the app for Jimmy](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_a_linq_query_to_finish_the_app_for_j)
8. [The var keyword lets C# figure out variable types for you](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#the_var_keyword_lets_chash_figure_out_va)
   1. [When you use var, C# figures out the variable’s type automatically](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#when_you_use_varcomma_chash_figures_out)
9. [LINQ is versatile](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_is_versatile)
10. [LINQ queries aren’t run until you access their results](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_queries_arenapostrophet_run_until_y)
11. [Use a group query to separate your sequence into groups](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_a_group_query_to_separate_your_seque)
12. [Use join queries to merge data from two sequences](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_join_queries_to_merge_data_from_two)
13. [Use the new keyword to create anonymous types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_new_keyword_to_create_anonymous)
14. [Unit tests help you make sure your code works](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#unit_tests_help_you_make_sure_your_code)
    1. [Add a second project to your solution for the unit tests](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#add_a_second_project_to_your_solution_fo)
15. [Start writing your first test method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#start_writing_your_first_test_method)
    1. [Run the unit test](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#run_the_unit_test)
16. [Give your unit tests access to the classes they’re testing](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#give_your_unit_tests_access_to_the_class)
    1. [Add a project reference so the unit tests can access the classes they need to test](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#add_a_project_reference_so_the_unit_test)
17. [One project can only access public classes in another project](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#one_project_can_only_access_public_class)
    1. [Make your JimmyLinq classes and enums public](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#make_your_jimmylinq_classes_and_enums_pu)
18. [Use the Arrange-Act-Assert pattern to write an effective test](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_arrange_act_assert_pattern_to_wr)
    1. [Use Assert.AreEqual to check that a calculated value matches an expected value](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_assertdotareequal_to_check_that_a_ca)
19. [Finish your first unit test](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#finish_your_first_unit_test)
20. [Write a unit test for the GetReviews method](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#write_a_unit_test_for_the_getreviews_met)
21. [Write unit tests to handle edge cases and weird data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#write_unit_tests_to_handle_edge_cases_an)
22. [Use the => operator to create lambda expressions](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_equalsgreater_than_operator_to_c)
23. [Refactor a clown with lambdas](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#refactor_a_clown_with_lambdas)
24. [Use the ?: operator to make your lambdas make choices](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_question_mark_operator_to_make_y)
25. [LINQ queries are made up of methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_queries_are_made_up_of_methods)
    1. [You can use lambda expressions with methods that take a Func parameter](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#you_can_use_lambda_expressions_with_meth)
26. [LINQ declarative syntax can be refactored into chained methods](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#linq_declarative_syntax_can_be_refactore)
27. [Use the => operator to create switch expressions](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_equalsgreater-id0001)
28. [Explore the Enumerable class](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#explore_the_enumerable_class)
    1. [Enumerable.Empty creates an empty sequence of any type](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#enumerabledotempty_creates_an_empty_sequ)
    2. [Enumerable.Repeat repeats a value a specific number of times](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#enumerabledotrepeat_repeats_a_value_a_sp)
    3. [So what exactly is an IEnumerable<T>?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#so_what_exactly_is_an_ienumerableless_th)
29. [Create an enumerable sequence by hand](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#create_an_enumerable_sequence_by_hand)
30. [Use yield return to create your own sequences](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_yield_return_to_create_your_own_sequ)
    1. [Use the debugger to explore yield return](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_the_debugger_to_explore_yield_return)
31. [Use yield return to refactor ManualSportSequence](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#use_yield_return_to_refactor_manualsport)
    1. [Add an indexer to BetterSportSequence](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#add_an_indexer_to_bettersportsequence)
32. [Collectioncross](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#collectioncross)
33. [Collectioncross solution](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch09.html#collectioncross_solution)

 [10. Reading and Writing Files Save the last byte for me](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#reading_and_writing_files_save_the_last)

1. [.NET uses streams to read and write data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#dotnet_uses_streams_to_read_and_write_da)
2. [Different streams read and write different things](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#different_streams_read_and_write_differe)
   1. [Things you can do with a stream:](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#things_you_can_do_with_a_stream)
3. [A FileStream reads and writes bytes in a file](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#a_filestream_reads_and_writes_bytes_in_a)
4. [Write text to a file in three simple steps](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#write_text_to_a_file_in_three_simple_ste)
5. [The Swindler launches another diabolical plan](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#the_swindler_launches_another_diabolical)
   1. [StreamWriter.WriteLine works just like Console.WriteLine](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#streamwriterdotwriteline_works_just_like)
6. [StreamWriter Magnets](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#streamwriter_magnets)
7. [StreamWriter Magnets Solution](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#streamwriter_magnets_solution)
8. [Use a StreamReader to read a file](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_a_streamreader_to_read_a_file)
9. [Data can go through more than one stream](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#data_can_go_through_more_than_one_stream)
10. [Use the static File and Directory classes to work with files and directories](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_the_static_file_and_directory_classe)
    1. [Things you can do with the static File class:](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#things_you_can_do_with_the_static_file_c)
    2. [Things you can do with the static Directory class:](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#things_you_can_do_with_the_static_direct)
11. [IDisposable makes sure objects are closed properly](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#idisposable_makes_sure_objects_are_close)
    1. [Use the IDE to explore IDisposable](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_the_ide_to_explore_idisposable)
12. [Avoid filesystem errors with using statements](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#avoid_filesystem_errors_with_using_state)
    1. [Use multiple using statements for multiple objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_multiple_using_statements_for_multip)
13. [Use a MemoryStream to stream data to memory](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_a_memorystream_to_stream_data_to_mem)
    1. [Use Encoding.UTF8.GetString to convert byte arrays to strings](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_encodingdotutf8dotgetstring_to_conve)
14. [What happens to an object when it’s serialized?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#what_happens_to_an_object_when_itapostro)
15. [But what exactly IS an object’s state? What needs to be saved?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#but_what_exactly_is_an_objectapostrophes)
16. [When an object is serialized, all of the objects it refers to get serialized too…](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#when_an_object_is_serializedcomma_all_of)
17. [Use JsonSerializer to serialize your objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_jsonserializer_to_serialize_your_obj)
18. [JSON only includes data, not specific C# types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#json_only_includes_datacomma_not_specifi)
19. [Next up: we’ll take a deep dive into our data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#next_up_weapostrophell_take_a_deep_dive)
20. [C# strings are encoded with Unicode](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#chash_strings_are_encoded_with_unicode)
21. [Visual Studio works really well with Unicode](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#visual_studio_works_really_well_with_uni)
22. [.NET uses Unicode to store characters and text](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#dotnet_uses_unicode_to_store_characters)
23. [C# can use byte arrays to move data around](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#chash_can_use_byte_arrays_to_move_data_a)
24. [Use a BinaryWriter to write binary data](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_a_binarywriter_to_write_binary_data)
25. [Use BinaryReader to read the data back in](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_binaryreader_to_read_the_data_back_i)
26. [A hex dump lets you see the bytes in your files](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#a_hex_dump_lets_you_see_the_bytes_in_you)
    1. [How to make a hex dump of some plain text](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#how_to_make_a_hex_dump_of_some_plain_tex)
27. [Use StreamReader to build a hex dumper](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_streamreader_to_build_a_hex_dumper)
28. [Use Stream.Read to read bytes from a stream](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_streamdotread_to_read_bytes_from_a_s)
29. [Modify your hex dumper to read directly from the stream](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#modify_your_hex_dumper_to_read_directly)
30. [Run your app from the command line](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#run_your_app_from_the_command_line)
    1. [Use the args variable to access command-line arguments](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#use_the_args_variable_to_access_command)
31. [Pass command-line arguments to an app run in the IDE](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch10.html#pass_command_line_arguments_to_an_app_ru)

 [Unity Lab #5 Raycasting](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#unity_lab_hash5_raycasting)

1. [Create a new Unity project and start to set up the scene](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#create_a_new_unity_project_and_start_to)
2. [Set up the camera](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#set_up_the_camera)
3. [Create a GameObject for the player](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#create_a_gameobject_for_the_player)
4. [Introducing Unity’s navigation system](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#introducing_unityapostrophes_navigation)
5. [Install the AI Navigation package](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#install_the_ai_navigation_package)
6. [Things you’ll do with navigation](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#things_youapostrophell_do_with_navigatio)
7. [Set up the NavMesh](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#set_up_the_navmesh)
   1. [Fix the carved-out hole in your NavMesh](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#fix_the_carved_out_hole_in_your_navmesh)
8. [Make your player automatically navigate the play area](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword05.html#make_your_player_automatically_navigate)

 [11. Captain Amazing the Death of the Object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#head_first_chash_four_bucks_captain_amaz)

1. [The life and death of an object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#the_life_and_death_of_an_object)
2. [Use the GC class (with caution) to force garbage collection](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#use_the_gc_class_left_parenthesiswith_ca)
3. [Your last chance to DO something... your object’s finalizer](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#your_last_chance_to_do_somethinghellip_y)
4. [When EXACTLY does a finalizer run?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#when_exactly_does_a_finalizer_runquestio)
   1. [You can SUGGEST to .NET that it’s time to collect the garbage](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#you_can_suggest_to_dotnet_that_itapostro)
5. [Finalizers can’t depend on other objects](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#finalizers_canapostrophet_depend_on_othe)
   1. [Don’t use finalizers for serialization](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#donapostrophet_use_finalizers_for_serial)
6. [A struct looks like an object...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#a_struct_looks_like_an_objectdotdot)
   1. [...but isn’t an object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#hellipbut_isnapostrophet_an_object)
7. [Values get copied; references get assigned](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#values_get_copiedsemicolon_references_ge)
8. [Structs are value types; objects are reference types](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#structs_are_value_typessemicolon_objects)
   1. [Here’s what happened...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#hereapostrophes_what_happeneddotdot)
9. [The stack vs. the heap: more on memory](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#the_stack_vsdot_the_heap_more_on_memory)
10. [Use out parameters to make a method return more than one value](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#use_out_parameters_to_make_a_method_retu)
11. [Pass by reference using the ref modifier](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#pass_by_reference_using_the_ref_modifier)
12. [Use optional parameters to set default values](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#use_optional_parameters_to_set_default_v)
13. [A null reference doesn’t refer to any object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#a_null_reference_doesnapostrophet_refer)
14. [Non-nullable reference types help you avoid NREs](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#non_nullable_reference_types_help_you_av)
    1. [Use encapsulation to prevent your property from ever being null](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#use_encapsulation_to_prevent_your_proper)
15. [Nullable value types can be null...and handled safely](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#nullable_value_types_can_be_nullhellipan)
16. [The null-coalescing operator ?? checks for nulls automatically](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#the_null_coalescing_operator_question_ma)
    1. [?? checks for null and returns an alternative](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#question_markquestion_mark_checks_for_nu)
    2. [??= assigns a value to a variable only if it’s null](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#question_markquestion_markequals_assigns)
17. [“Captain” Amazing...not so much](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#quotation_markcaptainquotation_mark_amaz)
18. [Records give your objects value equality automatically](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#records_give_your_objects_value_equality)
19. [Don’t modify records—copy them](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#donapostrophet_modify_recordsem_dashcopy)
    1. [Use the with keyword to create a modified copy of your Duck](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#use_the_with_keyword_to_create_a_modifie)
20. [Extension methods add new behavior to EXISTING classes](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#extension_methods_add_new_behavior_to_ex)
21. [Extending a fundamental type: string](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch11.html#extending_a_fundamental_type_string)

 [12. Exception Handling Putting out fires gets old](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#exception_handling_putting_out_fires_get)

1. [Your hex dumper reads a filename from the command line](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#your_hex_dumper_reads_a_filename_from_th)
   1. [But what happens if you give HexDump an invalid filename?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#but_what_happens_if_you_give_hexdump_an)
2. [When your program throws an exception, the CLR generates an Exception object](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#when_your_program_throws_an_exceptioncom)
3. [All Exception objects inherit from System.Exception](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#all_exception_objects_inherit_from_syste)
4. [There are some files you just can’t dump](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#there_are_some_files_you_just_canapostro)
5. [What happens when a method you want to call is risky?](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#what_happens_when_a_method_you_want_to_c)
6. [Handle exceptions with try and catch](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#handle_exceptions_with_try_and_catch)
7. [Use the debugger to follow the try/catch flow](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#use_the_debugger_to_follow_the_trysolidu)
8. [If you have code that ALWAYS needs to run, use a finally block](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#if_you_have_code_that_always_needs_to_ru)
9. [Catch-all exceptions handle System.Exception](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#catch_all_exceptions_handle_systemdotexc)
   1. [Avoid catch-all exception with multiple catch blocks](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#avoid_catch_all_exception_with_multiple)
10. [Pool Puzzle](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#pool_puzzle-id0001)
11. [Pool Puzzle Solution](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#pool_puzzle_solution-id0001)
12. [Use the right exception for the situation](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#use_the_right_exception_for_the_situatio)
    1. [Catch custom exceptions that extend System.Exception](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#catch_custom_exceptions_that_extend_syst)
13. [Exception Magnets](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#exception_magnets)
14. [Exception Magnets Solution](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#exception_magnets_solution)
15. [Exception filters help you create precise handlers](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#exception_filters_help_you_create_precis)
16. [The worst catch block EVER: catch-all plus comments](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#the_worst_catch_block_ever_catch_all_plu)
    1. [You should handle your exceptions, not bury them](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#you_should_handle_your_exceptionscomma_n)
17. [Temporary solutions are OK (temporarily)](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#temporary_solutions_are_ok_left_parenthe)
18. [Use NuGet to add a logging library to your app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#use_nuget_to_add_a_logging_library_to_yo)
    1. [Packages add DLLs with classes](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#packages_add_dlls_with_classes)
19. [Add logging to your ExceptionExperiment app](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ch12.html#add_logging_to_your_exceptionexperiment)

 [Unity Lab #6 Scene Navigation](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#unity_lab_hash6_scene_navigation)

1. [Let’s pick up where the last Unity Lab left off](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#letapostrophes_pick_up_where_the_last_un)
2. [Add a platform to your scene](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#add_a_platform_to_your_scene)
3. [Use bake options to make the platform walkable](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#use_bake_options_to_make_the_platform_wa)
4. [Include the stairs and ramp in the NavMesh](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#include_the_stairs_and_ramp_in_the_navme)
5. [Make the player navigate around the obstacles](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#make_the_player_navigate_around_the_obst)
   1. [Fix the moving obstacle](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#fix_the_moving_obstacle)
6. [Get creative!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#get_creativeexclamation_mark-id0004)
7. [Thank you for reading our book!](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#thank_you_for_reading_our_bookexclamatio)
   1. [But wait, there’s more! Your journey’s just begun...](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#but_waitcomma_thereapostrophes_moreexcla)
   2. [Check out these essential (and amazing!) books written by our friends and colleagues, also published by](https://learning.oreilly.com/library/view/head-first-c/9781098141776/afterword06.html#check_out_these_essential_left_parenthes)

 [Index](https://learning.oreilly.com/library/view/head-first-c/9781098141776/ix01.html#id2415)

 [The Authors](https://learning.oreilly.com/library/view/head-first-c/9781098141776/colophon01.html#the_authors)