Exam Ref 70-483 Programming In C# (MCSD)
Synopsis
Prepare for Microsoft Exam 70-483 "and help demonstrate your real-world mastery of programming in C#. Designed for experienced software developers ready to advance their status, Exam Ref focuses on the critical-thinking and decision-making acumen needed for success at the Microsoft Specialist level. Focus on the expertise measured by these objectives: Manage Program Flow Create and Use Types Debug Applications and Implement Security Implement Data Access This Microsoft Exam Ref: Organizes its coverage by exam objectives. Features strategic, what-if scenarios to challenge you.

Book Information
Series: Exam Ref
Paperback: 384 pages
Publisher: Microsoft Press; 1 edition (July 25, 2013)
Language: English
ISBN-10: 0735676828
Product Dimensions: 7.3 x 0.9 x 8.9 inches
Shipping Weight: 1.4 pounds (View shipping rates and policies)
Average Customer Review: 3.5 out of 5 stars See all reviews (39 customer reviews)
Best Sellers Rank: #139,430 in Books (See Top 100 in Books) #50 in Computers & Technology > Certification > Microsoft #54 in Books > Textbooks > Computer Science > Object-Oriented Software Design #59 in Books > Computers & Technology > Programming > Languages & Tools > C#

Customer Reviews
This book will lead people down bad pathways, confuse them, or frustrate them. There are many problems with it and until they are addressed, I strongly recommend this book not be read. Here are the major issues I have with this book:- A lot of technical information is incorrect. I was able to find 119 serious and minor technical errors. Some examples are "First of all, equal items should have equal hash codes. This means that you can check to determine whether two items are equal by checking their hash codes", "There are two places in memory where CLR stores items while your code executes. One is the stack; the other is the heap. The stack keeps track of what’s executing in your code, and the heap keeps track of your objects...For an object on the heap, there is always a reference on the stack that points to it", "The enumerator has a MoveNext method that returns the
next item in the collection", and the code on page 156 showing how WeakReferences should be used uses them incorrectly and could possibly return a null reference if the garbage collector collects the target after it is verified to have a non-null reference. Also, there are many code examples which don’t compile, answers which were copied and pasted from other questions, and correct answers listed as incorrect and vice versa.- There are many other errors are well. I was able to find 60 typos and language or formatting errors. Many of these are careless: references to code listings using the wrong listing numbers, listing descriptions being copied and pasted from other listings and then never updated, and code listings with unconventional formatting which makes it difficult to read.- There is a lot of missing or incomplete information.

I debated giving this book 1 star, but there is some useful information and it’s clear that the author does know his stuff. The problem is that the subject organization is terrible.My background: Professional programmer for 25 years, the last 8 with C#/.NET on a daily basis. I already passed the Microsoft C# exam for an older version of C# (2.0) and have worked with versions through 4.5 at work, I just wanted to "fill in the blanks" to upgrade my skill set. So this seemed like the ideal book.Here are my main issues:1) The only thing more inexplicable than Microsoft arranging the exam requirements in such a random order is that this *book* arranges them in the same order. Although I suppose it allows the user to verify that all of the requirements are being met more easily, it places complex programming concepts before basic concepts which makes it extremely difficult to follow. As an example, the books starts off (as the exam requirements do) with Asynchronous programming - one of the more complex programming ideas to wrap your head around - before it has even covered the basic language types, code structure, etc.2) Similarly, the author frequently references features and concepts that he hasn’t covered yet. As a quick example, he uses Lambda expressions in one of the earliest Asynchronous examples (which again, is complex in itself) with barely a mention as to what it even is. Then several pages later, he "introduces" the concept of Lambda expressions. Well that’s great, but it sure would have been nice to have them explained *before* he used them in the code.

Download to continue reading...