

Exam Ref 70-483: Programming in C#

Wouter de Kort

ISBN-13: 978-0-7356-7682-4

First printing: July, 2013

To ensure the ongoing accuracy of this book and its companion content, we've reviewed and confirmed the errors listed below. If you find a new error, we hope you'll report it to us on our website: www.microsoftpressstore.com/contact-us/errata.

Page	Location	Description	Date corrected
xvi	Who should read this book	<p>Remove the following paragraph at the end of Who should read this book:</p> <p>Developers reading this book should have a basic understanding of writing a simple C# program. You can create and run applications by using Visual Studio.</p> <p>Add the following to the Assumptions section behind behind the first sentence:</p> <p>To run the examples from this book you should be able to create a console application in Visual Studio.</p>	8/9/2013
6	Listing 1-4	<p>Remove the following lines from Listing 1-4:</p> <pre>public static void ThreadMethod(object o) { for(int I =0: I< (int)o;i++) { Console.WriteLine("ThreadProc: {0}", I); Thread.Sleep(0); } }</pre>	10/11/2013
9	1st paragraph	<p>Reads:</p> <p>You can use the Thread.Current-Thread class to ask for information about the thread that's executing.</p> <p>Should read:</p> <p>You can use the Thread.CurrentThread property to ask for information about the thread that's executing.</p>	

Page	Location	Description	Date corrected
18	1st paragraph after Listing 1.18	<p>Change the first line after Listing 1-18</p> <p>Reads: Because the entry method of an application can't be marked as async, the example uses the Wait method in Main.</p> <p>Should read: Because the entry method of an application can't be marked as async, the example accesses the Result property in the Main method which blocks the code until the async method DownloadContent is finished.</p>	10/11/2013
33	Listing 1-37	<p>The second Console.WriteLine reads: Console.WriteLine("Locked A and B")</p> <p>Should read: Console.WriteLine("Locked B and A")</p>	8/9/2013
39	Listing 1-44	<p>Add the following line after the closing } of the while statement:</p> <p>throw new OperationCanceledException();</p>	8/9/2013
42	Listing 1-46	<p>Change Listing title to:</p> <p>Using the equality operator</p>	8/9/2013
43	Listing 1-50	<p>Change the listing title to:</p> <p>Short-circuiting the AND operator</p>	8/9/2013
52	Listing 1-69	<p>Change listing title to:</p> <p>Implementing a for loop with a while statement</p>	8/9/2013
59	Paragraph before Listing 1-79	<p>Reads: Listing 1-79 shows how you would write the example in Listing 1-73 with newer lambda syntax.</p> <p>Should read: Listing 1-79 shows how you would write the example in Listing 1-75 with newer lambda syntax.</p>	

Page	Location	Description	Date corrected
80	Listing 1-98	<p>Reads: protected OrderProcessingException(SerializationInfo info, StreamingContext context)</p> <p>Should read: protected EntityOperationException(SerializationInfo info, StreamingContext context) : base(info, context)</p> <p>Reads: public void GetObjectData(SerializationInfo info, StreamingContext context) { info.AddValue("OrderId", OrderId, typeof(int)); }</p> <p>Should read: public override void GetObjectData(SerializationInfo info, StreamingContext context) { base.GetObjectData(info, context); info.AddValue("entityId", EntityId, typeof (int)); }</p>	
82	Objective review #3	Objective 1.5, question 3. All occurrences of LogonFailed in answer A-D should be changed to LogonFailedException.	
98	Listing 2-11	<p>Reads: this.maximumNumberOfCards = maximumNumberOfCards;</p> <p>Should read: _maximumNumberOfCards = maximumNumberOfCards;</p>	10/11/2013
109	Objective 2.2 - explicit conversion	<p>Reads: Where you can go implicitly from a derived type to a base type, you need to cast from a derived to a base type</p> <p>Should read: Where you can go implicitly from a derived type to a base type, you need to cast from a base type to a derived type</p>	8/9/2013
116	Table 2-3	<p>Change the description in Table 2-3 for the public accessmodifier to:</p> <p>None; unrestricted access.</p>	10/11/2013
123	Objective summary, 5th bullet	<p>Remove the comma between protected, internal.</p> <p>Reads: The access modifiers are public, internal, protected, protected, internal, and private.</p> <p>Should read: The access modifiers are public, internal, protected, protected internal, and private.</p>	10/11/2013

Page	Location	Description	Date corrected
132	Listing 2-50	<p>Change the code in Listing 2-50 to:</p> <pre> class Rectangle { public Rectangle(int width, int height) { Width = width; Height = height; } public virtual int Height { get; set; } public virtual int Width { get; set; } public int Area { get { return Height*Width; } } } </pre> <p>Change the code in Listing 2-51 to:</p> <pre> private class Square : Rectangle { public Square(int size) : base(size, size) { } public override int Width { get { return base.Width; } set { base.Width = value; base.Height = value; } } public override int Height { get { return base.Height; } set { base.Height = value; base.Width = value; } } } </pre>	10/11/2013

Page	Location	Description	Date corrected
133	1st sentence of section Implementing standard .NET Framework interfaces	Reads: "...that can you can use on your own types" Should read: "...that you can use on your own types"	10/11/2013
141	Paragraph below the listing 2-62	Change GetAttribute and GetAttributes to GetCustomAttribute and GetCustomAttributes in the paragraph below Listing 2-62.	
141	Listing 2-62	Reads: ConditionalAttribute conditionalAttribute = (ConditionalAttribute)Attribute.GetCustomAttribute(typeof(ConditionalClass), typeof(ConditionalAttribute)); Should read: ConditionalAttribute conditionalAttribute = (ConditionalAttribute)Attribute.GetCustomAttributes(typeof(ConditionalClass), typeof(ConditionalAttribute)).First();	
142	Listing 2-66	Change AttributeTargets.Class to AttributeTargets.Parameter in Listing 2-66	
153	2nd paragraph	Reads: Idiposable Should read: Idisposable	10/11/2013

Page	Location	Description	Date corrected
154	Example 2-84	<p>Change the Implementing IDisposable and a finalizer section to:</p> <p>Implementing IDisposable and a finalizer Creating your own custom type that implements IDisposable and a finalizer correctly is not a trivial task.</p> <p>As an example, suppose you have a wrapper class around a file resource and an unmanaged buffer. You implement IDisposable so users of your class can immediately cleanup if they want. You also implement a finalizer in case they forget to call Dispose. Listing 2-84 shows how to do this.</p> <p>LISTING 2-84 Implementing IDisposable and a finalizer.</p> <pre> Using System; using System.IO; using System.Runtime.InteropServices; class UnmanagedWrapper : IDisposable { private IntPtr unmanagedBuffer; public FileStream Stream { get; private set; } public UnmanagedWrapper() { CreateBuffer(); this.Stream = File.Open("temp.dat", FileMode.Create); } private void CreateBuffer() { byte[] data = new byte[1024]; new Random().NextBytes(data); unmanagedBuffer = Marshal.AllocHGlobal(data.Length); Marshal.Copy(data, 0, unmanagedBuffer, data.Length) } ~UnmanagedWrapper() { Dispose(false); } public void Close() { Dispose(); } public void Dispose() { Dispose(true); } </pre>	8/9/2013

Page	Location	Description	Date corrected
		<pre> System.GC.SuppressFinalize(this); } protected virtual void Dispose(bool disposing) { Marshal.FreeHGlobal(unmanagedBuffer); if (disposing) { if (Stream != null) { Stream.Close(); } } } </pre> <p>There are a couple of things to notice about this implementation:</p> <ul style="list-style-type: none"> • The finalizer only calls Dispose passing false for disposing. • The extra dispose method with the Boolean argument does the real work. This method checks if it's being called in an explicit dispose or if it's being called from the finalizer: <ul style="list-style-type: none"> o If the finalizer called Dispose, you only release the unmanaged buffer. The Stream object also implements a finalizer and the garbage collector will take care of calling the finalizer of the FileStream instance. Because the order in which the garbage collector calls the finalizers is unpredictable you can't call any methods on the FileStream. o If Dispose is called explicitly, you also close the underlying FileStream. It's important to be defensive in coding this method and always check for any source of possible exceptions It could be that Dispose is called multiple times and that shouldn't cause any errors. • The regular Dispose method calls GC.SuppressFinalize(this) to make sure that the object is removed from the finalization list that the garbage collector is keeping track of. The instance has already cleaned up after itself; so it's not necessary that the garbage collector calls the finalizer. <p>EXAM TIP</p> <p>It's important to know the difference between implementing IDisposable and a finalizer. Both cleanup your object but a finalizer is called by the Garbage Collector while the Dispose method can be called from code.</p> <p>MORE INFO Implementing IDisposable and a finalizer For more information on how to implement IDisposable and a finalizer see: http://msdn.microsoft.com/en-us/library/b1yfk5e.aspx.</p>	
160	Next to last line	<p>The paragraph after Listing 2-88 reads:</p> <pre>new String("x", 10000)</pre> <p>Should read:</p> <pre>new String('x',10000)</pre>	

Page	Location	Description	Date corrected
161	4th paragraph	Reads: TextWriter Should read: TextReader	10/11/2013
175	Example 1-1	Remove the line //ThreadProc: 10 at the end of the listing.	8/9/2013
189	Listing 3-11	Change listing title to: Collapse multiple spaces with RegEx	8/9/2013
201	Fourth paragraph	After Listing 3-22 (4th paragraph) remove the line: This means that you can check to determine whether two items are equal by checking their hash codes.	
226	2nd paragraph under Managing program database...	The sentence should read 'You can instruct the compiler...'	10/11/2013
226	Listing 3-43	Change the first line of Listing 3-43to: [DebuggerDisplay("Name = {FirstName} {LastName}")]	
243	3rd bullet of Performance Counter types	Reads: AvergateTimer32 Should read: AverageTimer32	10/11/2013
247	Objective 3.1 Review question 1 answer A	The extra comma (,) after 'the user enters an invalid date' should be removed.	10/11/2013
247	Within Objective 3.1: Review	The answer for Objective 3.1, question 1 is missing. The answer should be: 1. Correct answer: B A. Incorrect: The regular Parse method will throw an exception if the value the user entered is not a DateTime. This is something you need to expect when you parse user input. Using TryParse avoids the exception. B. Correct: TryParse should be used when working with user input. C. Incorrect: Convert.ToDateTime calls Parse internally. This will throw an exception when the user input is not in the correct format. D. Incorrect: RegEx.Match will search the input string for the first match to the specified regular expression. It won't convert the input string to a DateTime.	
248	Objective 3.2: Review # 3 C	Objective 3.2 question number 3 should have C marked as Incorrect in the answers section.	

Page	Location	Description	Date corrected
249	Objective 3.3: Review - Question 2	Mark answer D for question 2 of objective 3.3 on page 249 as correct. Should read: Correct. Strongly naming an assembly is required to deploy an assembly to the GAC where it can be shared by all applications.	10/11/2013
259	Paragraph below the listing 4-12	Change GetExtensions to GetExtension	
260		In Listing 4-14, 4-15, 4-18 and 4-19 replace the << character with " string path = @"c:\temp\test.dat"; On page 261 in Listing 4-15 change <MyValue> to "MyValue" Listing 4-19 change <<A line of tekst.>> to "A line of text."	10/11/2013
286	LISTING 4-43	In Listing 4-43 the line: xmlReader.ReadStartElement("Person"); should be changed to: xmlReader.ReadStartElement("person"); And the line: xmlReader.ReadStartElement("ContactDetails"); should be changed to: xmlReader.ReadStartElement("contactdetails");	10/11/2013
314	2nd paragraph in the chapter "Using DataContract"	Reads: The most noticeable difference is that you use DataContractAttribute instead of SerializableAttribute. Another important difference is that members are not serialized by default. You have to explicitly mark them with the DataMember attribute. Should read: The most noticeable difference is that you use DataContractAttribute instead of SerializableAttribute. If you omit both the DataContract and DataMember attribute, your public members are serialized. With the DataContract attribute applied, you have to specify which members you want to serialize by using the DataMember attribute on them.	
330	Answer 2 D	Change the explanation for answer D, question 4.2 to the following: A Web service will only move the problem to another layer of your application. Inside the Web service you still have to use some kind of storage to save your application data.	
333	Objective 4.5	On question 2 for Objective 4.5, the answer B should be removed both from the first line and the Correct should be changed to incorrect. Only answer D is correct.	10/11/2013
333	Objective 4.5: Objective review 1 D	The first answer should only have B marked as correct. D should be incorrect.	10/11/2013