About This Exam Review Kit

Microsoft Technology Associate Certification Exam Review Kit: 98-372 Microsoft .NET Fundamentals

Exam Review Kit Description

- This Microsoft® Technology Associate (MTA) Certification Exam Review Kit contains a series of 20 review lessons intended to reinforce concepts in preparation for the *MTA Certification Exam: 98-372 Microsoft .NET Fundamentals* and serve as a resource and guide for teachers and faculty to create their own additional student learning experiences.
- It is assumed that students taking an MTA certification exam have completed or are currently taking academic courses, have job experience that addresses the exam objective domain, or both.
- The Exam Review Kits:
    - Are intended to supplement (not supplant) existing academic courses.
    - Are not intended to serve as foundational content for academic courses.
    - Are tied directly and closely to the objective domain of each individual MTA certification exam.
    - Are platform-specific or -agnostic in accord with the objective domain of each MTA certification exam. Code samples are provided in both Microsoft Visual C#® 2010 and Microsoft Visual Basic® 2010 when appropriate.
- Because each certification exam has approximately 20 objectives, this Exam Review Kit includes 20 review lessons of 50 minutes apiece.
- The materials for each review lesson include a lesson plan, lesson delivery materials, and Student Activity documents.
- MTA certification exams test breadth of technical knowledge and help students explore career options before choosing a specialized career path with minimal investment of time and money. MTA certifications measure and validate the fundamental technology skills that are in demand today and provide an essential foundation to build a career in technology. Earning MTA certification provides students with a credential that validates fundamental technology industry knowledge and motivates them to succeed in continued studies, compete for admission to advanced training, and prepare for a career in technology. The MTA certifications enable students to prove their commitment to technology and connect with a community of more than 5 million Microsoft Certified Professionals (MCPs).
- Teachers and faculty can integrate the new MTA certification exams easily into existing schedules and curricula and deliver exams right in the classroom, on their own schedules.

Audience

- This Exam Review Kit is intended for students attending high schools and two-year colleges and technology workers who are preparing for the *MTA Certification Exam: 98-372Microsoft .NET Fundamentals* and seek to prove introductory knowledge of and skills with Microsoft Windows® programming and application development.
- It is recommended that exam candidates be familiar with the concepts of and have hands-on experience with the technologies described here, either by taking relevant training courses or

by working with Microsoft Visual Studio® and tutorials and samples available on MSDN®. Although minimal hands-on experience with the technologies is recommended, job experience is not assumed for these exams.

- Candidates for this exam are in the process of expanding their knowledge and skills in the following areas:
    - .NET namespace and class organizations
    - Core .NET knowledge
    - Managed code theory
    - Memory management in .NET
    - Language parity

Student Prerequisites

This course requires that you meet the following prerequisites:

- It is assumed that students taking an MTA certification exam have completed or are currently taking academic courses, have job experience that addresses the exam objective domain, or both.
- While the test is intended to be language-agnostic, it is expected that students have had experience using a Microsoft .NET language such as C# or Visual Basic.

Exam Review Kit Objective Domain

This Exam Review Kit provides lessons that reinforce previous learning in the following objectives:

1. Understanding .NET Framework Concepts

1.1. Understand basic application settings.
This objective may include but is not limited to: understanding app.config and web.config.

1.2. Understand events and event handling in the .NET Framework.
This objective may include but is not limited to: understanding the event-driven programming model and event handlers, raising events, and implementing delegates.

1.3. Understand structured exception handling in the .NET Framework.
This objective may include but is not limited to: understanding error handling concepts, exceptions, and exception types.

2. Understanding Namespaces and Classes in the .NET Framework

2.1. Understand .NET class hierarchies.
This objective may include but is not limited to: understanding system classes, classifications of classes, and logical organization of classes.

2.2. Understand Object Oriented Concepts in the .NET Framework.
This objective may include but is not limited to: understanding how inheritance works in .NET, polymorphism, and interfaces.

2.3. Understand .NET namespaces.
This objective may include but is not limited to: understanding the reason for namespaces, the organization of namespaces in the class library, and how to use namespaces in an application.
2.4. Understand and create class libraries.
This objective may include but is not limited to: understanding the logical grouping of classes and the logic behind class libraries (what they are, why they are important).
2.5. Understand and use different data types in the .NET Framework.
This objective may include but is not limited to: understanding intrinsic data types, values types, reference types, boxing, unboxing, and .NET collection classes.
2.6. Understand generics.
This objective may include but is not limited to: understanding generics infrastructure, generic interfaces, generic delegates, contravariant and covariant generic type arguments, generic methods, verifiability, and constraints.

3. Understanding .NET Code Compilation

3.1. Understand the fundamentals of Microsoft Intermediate Language (MSIL) and Common Language Infrastructure (CLI).
This objective may include but is not limited to: understanding what MSIL is, what the CLI is, how source code is compiled into MSIL, and how code is Just-in-Time (JIT) compiled.
3.2. Understand the use of strong naming.
This objective may include but is not limited to: understanding why strong naming is used, how to sign assemblies to support strong naming, and Global Assembly Cache (GAC).
3.3. Understand version control.
This objective may include but is not limited to: understanding how .NET applications are versioned and how to run different versions on the same computer.
3.4. Understand assemblies and metadata.
This objective may include but is not limited to: understanding .NET assemblies, shared assemblies, and metadata in .NET.

4. Understanding I/O Classes in the .NET Framework

4.1. Understand .NET file classes.
This objective may include but is not limited to: understanding read/write file classes and stream readers and writers.
4.2. Understand console I/O.
This objective may include but is not limited to: understanding System.Console classes for input and output.
4.3. Understand XML classes in the .NET Framework.
This objective may include but is not limited to: understanding XMLReader, XMLWriter, and XML Schemas.

5. **Understanding Security**
   5.1. Understand the System Security namespace.
   This objective may include but is not limited to: understanding permissions and what cryptography is.
   5.2. Understand authentication and authorization.
   This objective may include but is not limited to: understanding code access security, access control, and policies.

6. **Understanding .NET Languages**

   6.1. Understand language interoperability.
   This objective may include but is not limited to: calling code written in one language from another language, understanding .NET language parity
   6.2. Understand type safety.
   This objective may include but is not limited to: understanding memory type safety, type safety in classes, strong types, and security policies.

7. **Understanding Memory Management**
   7.1. Understand resource allocation.
   This objective may include but is not limited to: understanding garbage collection and memory allocation, understanding stack versus heap.
   7.2. Understand the difference between managed and unmanaged applications.
   This objective may include but is not limited to: understanding why managed code is called managed code, understanding the differences between coding in managed versus unmanaged code.

Exam Review Kit Timing

Each of the 20 Review Lessons in this collection is intended to be used in a single 50-minute class period.

Exam Review Kit Materials

The following materials are included in this Exam Review Kit:

- **Review Lessons:** A plan for teacher and student activities in reviewing the learning objectives and providing the key points that are critical to the success of the in-class review experience.
- **Microsoft PowerPoint® presentations:** A structure for classroom lectures and discussions.
- **Student Activities:** A hands-on platform for applying the knowledge and skills reviewed in the lesson.
- **Student Activity Answer Keys:** Solutions to Student Activities.
- **Additional resources:** Various resources to expand reviewing and learning opportunities.
- Duplicate resources may be available for some lessons in both C# and Visual Basic as appropriate.

Software Requirements

The following software is suggested for this series of Review Lessons:

- Microsoft Visual Studio 2010, or
  - Microsoft Visual Basic 2010, Express Edition (*http://www.microsoft.com/express/downloads/#2010-Visual-Basic)*
  - Microsoft Visual C# 2010 Express Edition (*http://www.microsoft.com/express/downloads/#2010-Visual-CS*)
- Microsoft PowerPoint 2007

Instructional Preparation Activities

It is highly recommended that you complete the following instructional preparation activities:

- Familiarize yourself with the objectives of each lesson.
- Walk through each Review Lesson presentation slideshow and read the corresponding Instructor Notes (located in the Notes view of the presentation slideshow) for the lesson.
- Familiarize yourself with the Student Activities.
- Practice presenting each module.
- Identify the key points and must-know information for each topic.
- Perform each demonstration and hands-on lab.
- Anticipate the questions that students might have.
- Identify examples, analogies, impromptu demonstrations, and additional delivery tips that will help to clarify module content and provide a more meaningful learning experience for your specific audience.
- Customize and enhance your instructor notes.
- Review the updated information about the Microsoft Certification Program on the Microsoft Learning Certifications website (*http://www.microsoft.com/learning/en/us/certification/cert-default.aspx*).