



# **Microsoft Technology Associate Certification Exam Review Kit:**

**98-361 Software Development Fundamentals**

© 2010 Microsoft Corporation. All Rights Reserved. This courseware material (“Material”) is provided as part of your MTA Certification Program. You may not copy, adapt, modify, prepare derivative works of, distribute, display publicly, or resell this Material without first receiving permission from Microsoft. If you have questions about reuse of this Material, please write to us at [learningperm@microsoft.com](mailto:learningperm@microsoft.com) with details about your request.

Microsoft®, Access®, MS®, MSDN®, MS-DOS®, Outlook®, PowerPoint®, Visual Studio®, Windows®, and Windows Live® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are property of their respective owners.

Microsoft Technology Associate Certification Exam Review Kit:  
98-361 Software Development Fundamentals

# About This Exam Review Kit

## ***Microsoft Technology Associate Certification Exam Review Kit: 98-361 Software Development 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: 98-361 Software Development Fundamentals* and/or 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 and/or are currently taking academic courses, have job experience that addresses the exam objective domain, or both.
- The MTA Certification Exam Review Kits:
  - Are intended to supplement (not supplant) existing academic courses
  - Are not intended to serve as foundational content for academic courses
  - Are directly and closely tied 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.
- Because each certification exam has approximately twenty 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 on admissions, 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 is intended for students attending high schools and two-year colleges who are preparing for the *MTA Certification Exam 98-361: Software Development Fundamentals* and seek to prove knowledge of fundamental software development concepts and basic programming skills.

- 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 tutorials and samples available on MSDN® and in Microsoft Visual Studio®. 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:
  - Core programming concepts
  - Object-oriented design and programming
  - General software development strategies and practices
  - Understanding Web applications
  - Understanding Windows® Forms and console-based applications
  - Understanding relational database systems

## Student Prerequisites

This course requires that you meet the following prerequisites:

- It is assumed that students taking an MTA certification exam have completed and/or are currently taking academic courses, have job experience that addresses the exam objective domain, or both.
- It is expected that students have had experience using Visual Studio® with 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 Core Programming

#### 1.1. Understand computer storage and data types.

*This objective may include but is not limited to:* how a computer stores programs and the instructions in computer memory; memory stacks and heaps; memory size requirements for the various data storage types; numeric data and textual data.

#### 1.2. Understand computer decision structures.

*This objective may include but is not limited to:* various decision structures used in all computer programming languages; If decision structures; multiple decision structures such as If...Else and switch/Select Case; reading flowcharts; decision tables; evaluating expressions.

#### 1.3. Identify the appropriate method for handling repetition.

*This objective may include but is not limited to:* For loops, While loops, Do...While loops, and recursion.

## 1.4. Understand error handling.

*This objective may include but is not limited to:* structured exception handling.

**2. Understanding Object-Oriented Programming**

## 2.1. Understand the fundamentals of classes.

*This objective may include but is not limited to:* properties, methods, events, and constructors; how to create a class; how to use classes in code.

## 2.2. Understand inheritance.

*This objective may include but is not limited to:* inheriting the functionality of a base class into a derived class.

## 2.3. Understand polymorphism.

*This objective may include but is not limited to:* extending the functionality in a class after inheriting from a base class; overriding methods in the derived class.

## 2.4. Understand encapsulation.

*This objective may include but is not limited to:* creating classes that hide their implementation details while still allowing access to the required functionality through the interface; access modifiers.

**3. Understanding General Software Development**

## 3.1. Understand application life cycle management.

*This objective may include but is not limited to:* phases of application life cycle management; software testing.

## 3.2. Interpret application specifications.

*This objective may include but is not limited to:* reading and translating application specifications into prototypes, code, and components.

## 3.3. Understand algorithms and data structures.

*This objective may include but is not limited to:* arrays, stacks, queues, linked lists, and sorting algorithms; performance implications of various data structures; choosing the right data structure.

NOT: algorithm analysis.

**4. Understanding Web Applications**

## 4.1. Understand Web page development.

*This objective may include but is not limited to:* HTML, Cascading Style Sheets (CSS), JavaScript.

## 4.2. Understand Microsoft ASP.NET Web application development.

*This objective may include but is not limited to:* page life cycle; event model; state management; client-side vs. server-side programming.

## 4.3. Understand Web hosting.

*This objective may include but is not limited to:* creating virtual directories and Web sites; deploying Web applications; understanding the role of Internet Information Services.

## 4.4. Understand Web services.

*This objective may include but is not limited to:* Web services that will be consumed by client applications; accessing Web services from a client application; SOAP and Web Service Definition Language (WSDL).

**5. Understanding Desktop Applications**

## 5.1. Understand Windows Forms applications.

*This objective may include but is not limited to:* Windows Forms event model; visual inheritance; UI design; use of Multiple Document Interface (MDI) and Single Document Interface (SDI) applications.

## 5.2. Understand console-based applications.

*This objective may include but is not limited to:* characteristics and capabilities of console-based applications.

## 5.3. Understand Windows Services.

*This objective may include but is not limited to:* characteristics and capabilities of Windows Service.

**6. Understanding Databases**

## 6.1. Understand relational database management systems.

*This objective may include but is not limited to:* characteristics and capabilities of database products; database design; Entity Relationship Diagrams (ERDs); normalization concepts.

## 6.2. Understand database query methods.

*This objective may include but is not limited to:* structured query language (SQL), creating and accessing stored procedures, updating data, selecting data.

## 6.3. Understand database connection methods.

*This objective may include but is not limited to:* connecting to various types of data stores such as flat file; XML file; in-memory object; resource optimization.

**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:

- Visual Studio 2008, or
  - Visual Basic 2008, Express Edition  
( Visual Basic 2008 Express Edition  
(<http://www.microsoft.com/express/downloads/#2008-Visual-Basic>)
  - Visual C# 2008 Express Edition  
(<http://www.microsoft.com/express/downloads/#2008-Visual-BasicCS>)
  - Visual C# 2008, Express Edition  
(<http://www.microsoft.com/express/downloads/#2008-Visual-CS>)
  - Visual Web Developer 2008, Express Edition  
(<http://www.microsoft.com/express/downloads/#2008-Visual-Web-Developer>)
  - Microsoft SQL Server®
  - 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 slide deck and read the corresponding Instructor Notes (located in the Notes view of the presentation slide deck) 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>).

# About the Authors

## Pat Yongpradit

"A man can do nothing better than to eat and drink and find satisfaction in his work." Pat loves being a computer science teacher at Springbrook High School in Silver Spring, Maryland. As a computer science student in high school, the last thing he wanted to do was learn programming. Now Pat's goal is to change the hearts and minds of those who feel that way. He teaches four levels of computer programming: introductory programming, Advanced Placement computer science, video game programming, and research. Pat's general goal is to encourage students to enjoy the beautiful challenge of logic and algorithms. He is currently pursuing this goal by promoting Lego robotics and XNA® Game Studio.



## Peggy Fisher

Peggy teaches computer science at Indian Valley High School (IVHS), a rural high school in Central, Pennsylvania, that offers courses in programming (C#, Visual Basic®, and Java for Advanced Placement courses), and Web design, including Expression Web, Hypertext Markup Language (HTML), JavaScript, and cascading style sheets (CSS). Prior to teaching, Peggy worked for a large insurance company outside Philadelphia. She started as a job coordinator, moved to programming, and then became a systems analyst and manager. Finally, she was promoted to director prior to leaving the corporate world to join the field of education. She has been at IVHS for the past eight years and truly enjoys her new career. Peggy also teaches part-time at Pennsylvania State University in the Continuing Education program. Her goal in teaching is to instill the love of learning so that her students will not only graduate, but also become life-long learners.



## Patricia Philips

Patricia taught computer science for 20 years in Janesville, Wisconsin. She served on Microsoft's National K–12 Faculty Advisory Board and edited the Microsoft® MainFunction website for technology teachers for two years. For the past four years, she has worked with Microsoft in a variety of roles related to K–12 curriculum development and pilot programs, including Web design and XNA. In her role as an author and editor, Patricia wrote several articles and a student workbook on topics including computer science, Web design, and computational thinking. She is currently the editor of the Computer Science Teachers Association newsletter, the *Voice*.





## Peer and Technical Reviewers

### **Kenney Chan**

Kenney has taught computer science for four years at Phillips Exeter Academy, in Exeter, New Hampshire. He has experience in C, C++, SQL<sup>®</sup>, Visual Basic, and Microsoft Access<sup>®</sup>. He received his BS in computer science from Cornell University, and his M.Ed. from Harvard University.

### **Peter Gruenbaum**

Peter has been a physicist, software developer, a technical writer, and a teacher. He has written about technologies as diverse as the Tablet PC, mobile phones, distance learning, and cloud computing. As a software developer, he has written software using Tablet PCs, Augmented Reality, 3-D visualization, and computer-aided design. He created a program to teach creative technology classes to low-income youth to inspire them to consider technology careers, obtaining grant money from the Gates Foundation, Microsoft, and others. He founded the company SDK Bridge to bring his writing and teaching together. Peter received his BA in physics from the University of Chicago and his Ph.D. in applied physics from Stanford University.

### **Brian G. Scarbeau**

Brian currently teaches computer science at Lake Highland Preparatory School in Orlando, Florida. He has 31 years of teaching experience, including 5 spent in college-level computer science. Brian was a Microsoft MVP for 5 years. Brian traveled the United States and Canada teaching teachers how to use Microsoft .NET products with their students. He developed curricula in ASP.NET Web Matrix and DotNetNuke for Microsoft and did a series of webcasts on DotNetNuke. He is a co-author of *Professional DotNetNuke 5* (Wrox 2009).

## **A special thanks to the editorial and production team:**

Project Manager: Valerie Woolley  
Production Manager: Diane Kohnen, S4Carlisle Publishing Services  
Technical Editor: Edward Laverick  
Copyeditor: Susan McClung