

CompuScholar, Inc.
 Alignment to Microsoft Technology Associates (MTA)
 "Software Development Fundamentals "
 Certification Exam Requirements

Microsoft Course Details:

Course Name:	MTA Software Development Fundamentals
Course Code(s):	98-361
Standards Link:	https://www.microsoft.com/en-us/learning/exam-98-361.aspx

CompuScholar Course Details:

Course Title:	Windows Programming
Course ISBN:	978-0-9887070-0-9
Course Year:	2015

Note 1: Citation(s) listed may represent a subset of the instances where objectives are met throughout the

Introduction

The Microsoft Technology Associates (MTA) "Software Development Fundamentals" exam is designed to test a broad range of knowledge spanning general programming concepts, software development lifecycles, website design and web services, desktop applications and Windows services, and databases. Due to the diversity and depth of the topics, more than one year of study and more than one course is generally needed for preparation.

The CompuScholar "TeenCoder: Windows Programming" course teaches fundamental programming concepts using the Microsoft C# language. It can be used as preparation for the programming aspects of this MTA exam. Other exam topics are marked as "n/a" below and would need to be covered by supplemental resources, with additional time.

Exam Requirements

Understanding core programming (15-20%)	CITATION(S)
Understand computer storage and data types	See Below
How a computer stores programs and the instructions in computer memory	Chapter 1, Lesson 1
memory stacks and heaps	n/a

memory size requirements for the various data storage types	Chapter 4, Lesson 1
numeric data and textual data	Chapter 4, Lesson 1 Chapter 4, Lesson 2 Chapter 4, Lesson 4
Understand computer decision structures	See Below
Various decision structures used in all computer programming languages	Chapter 5, Lesson 2 Chapter 5, Lesson 3 Chapter 5, Lesson 4
If decision structures	Chapter 5, Lesson 2
multiple decision structures, such as If...Else and switch/Select Case	Chapter 5, Lesson 2 (if/else)
reading flowcharts	Chapter 7, Lesson 3
decision tables	n/a
evaluating expressions	Chapter 5, Lesson 1
Identify the appropriate method for handling repetition	See Below
For loops	Chapter 5, Lesson 3
While loops	Chapter 5, Lesson 4
Do...While loops	Chapter 5, Lesson 4
recursion	Chapter 14, Lesson 2 Chapter 14, Lesson 3
Understand error handling	See Below
Structured exception handling	Chapter 10, Lesson 3

Understanding object-oriented programming (20-25%)	CITATION(S)
Understand the fundamentals of classes	See Below
Properties, methods, events	Chapter 12, Lesson 3 Chapter 13, Lesson 2
constructors	Chapter 13, Lesson 4
how to create a class	Chapter 13, Lesson 1

how to use classes in code	Chapter 13, Lesson 1 Chapter 13, Lesson 3
Understand inheritance	See Below
Inheriting the functionality of a base class into a derived class	Chapter 16, Lesson 1 Chapter 16, Lesson 2 Chapter 16, Lesson 3
Understand polymorphism	See Below
Extending the functionality in a class after inheriting from a base class	Chapter 16, Lesson 3 Chapter 16, Lesson 5
overriding methods in the derived class	Chapter 16, Lesson 3 Chapter 16, Lesson 5
Understand encapsulation	See Below
Creating classes that hide their implementation details while still allowing access to the required functionality through the interface	Chapter 13, Lesson 3
access modifiers	Chapter 13, Lesson 3

Understanding general software development (15-20%)	CITATION(S)
Understand application life cycle management	See Below
Phases of application life cycle management	Supplemental Lesson 6
software testing	Chapter 10, Lesson 4 Supplemental Lesson 6
Interpret application specifications	See Below
Reading application specifications and translating them into prototypes	Chapter 7, Lesson 3 Supplemental Lesson 6 Supplemental Activity 6
code	Chapter 7, Lesson 3 Supplemental Lesson 6 Supplemental Activity 6
select appropriate application type, and components	Chapter 7, Lesson 3 Supplemental Lesson 6 Supplemental Activity 6
Understand algorithms and data structures	See Below
Arrays	Chapter 11, Lesson 1
stacks	n/a

queues	n/a
linked lists	Chapter 11, Lesson 2
sorting algorithms	Chapter 14, Lesson 1
performance implications of various data structures	Chapter 11, Lesson 1 Chapter 11, Lesson 2 Chapter 14, Lesson 1
choosing the right data structure	Chapter 11, Lesson 1 Chapter 11, Lesson 2 Chapter 14, Lesson 1

Understanding web applications (15-20%)	CITATION(S)
Understand web page development	See Below
HTML, Cascading Style Sheets (CSS), JavaScript	n/a (See KidCoder: Web Design for in-depth coverage)
Understand Microsoft ASP.NET web application development	See Below
Page life cycle, event model, state management, client-side versus server-side programming	n/a
Understand web hosting	
Creating virtual directories and websites	n/a
deploying web applications	n/a
understanding the role of Internet Information Services	n/a
Understand web services	See Below
Web services that will be consumed by client applications, accessing web services from a client application, SOAP and Web Service Definition Language (WSDL)	n/a

Understanding desktop applications (15-20%)	CITATION(S)
Understand Windows Store apps	See Below
UI design guideline categories, characteristics and capabilities of Store Apps, identify gestures	n/a

Understand console-based applications	See Below
Characteristics and capabilities of console-based applications	Chapter 2, Lesson 3 Chapter 2, Lesson 4
Understand Windows Services	See Below
Characteristics and capabilities of Windows Services	n/a

Understanding databases (15-20%)	CITATION(S)
Understand relational database management systems	See Below
Characteristics and capabilities of database products, database design, Entity Relationship Diagrams (ERDs), normalization concepts	n/a
Understand database query methods	See Below
Structured query language (SQL), creating and accessing stored procedures, updating data and selecting data	n/a
Understand database connection methods	See Below
Connecting to various types of data stores, such as flat file; XML file; in-memory object; resource optimization	n/a