

Full MTA Guides

By Zane Chalich

Abstract

Here is a list of [every Microsoft Technology Associate certification](#) and their requirements. It also has examples and tips on how to accomplish primary and secondary objectives. My notes and links are below the title and in-line, frequently with square bullet points.

Table of Contents	
Abstract	2
Table of Contents	3
Introduction to Programming Using JavaScript	4
Windows Operating System Fundamentals	7
Introduction to Programming Using Java	10
Exam 98-367 Security Fundamentals	12
Introduction to Programming Using HTML and CSS	29
HTML5 Application Development Fundamentals	35
Introduction to Programming Using Block-Based Languages (Touch Develop)	37
Software Development Fundamentals	40
Introduction to Programming Using Python	45
Database Fundamentals	49
Windows Server Administration Fundamentals	52
Mobility and Devices Fundamentals	56
Cloud Fundamentals	58

Introduction to Programming Using JavaScript

Program with JavaScript Operators, Methods, and Keywords (20-25%)

- Complete or debug code that uses assignment and arithmetic operators
 - Assignment; increment; decrement; addition; subtraction; division; multiplication; modulus; compound assignment operators
 - [Modulus](#)
- Apply JavaScript best practices
 - Comments; indentations; naming conventions; noscript; constants; reserved keywords; debugger keyword; setting breakpoints; console.log
- Evaluate the use of inline and external scripts
 - When to use, how to use, and what happens when both are used
 - [W3 Schools \(external scripts\)](#)
- Implement exception handling
 - try; catch; finally
 - [W3 Schools \(try, catch, finally\)](#)
- Complete and debug code that interacts with the Browser Object Model (BOM)
 - Manage state; display dialogs; determine screen size

Program with Variables, Data Types, and Functions (25-30%)

- Declare and use variables of primitive data types
 - Number; Boolean; String; Null; Undefined; typeof operator; type checking functions; use strict; converting between data types; formatting numbers; string operations; single quote vs double quote (nesting); initialization
- Declare and use arrays
 - Single-dimensional arrays; multi-dimensional arrays; iteration; initialization; define an array; sort and search an array; use push, pop, shift, and unshift methods; use the length property; access an array element;
 - [Javascript arrays](#)

- `let array = ["a", "b", "c"];`
 - `array.unshift("at start");`
 - `array.push("at end")`
 - Will give you `["at start", "a", "b", "c", "at end"]`
 - `array.pop();`
 - `array.shift();`
 - Will give you `["a", "b", "c"]`
 - `array.sort();`
- Complete and debug code that uses objects
 - Properties; methods; instantiation; Date object; retrieve date and time parts; localize date format (MM/DD vs DD/MM); add and subtract dates
 - [Date:](#)
 - `var date = new Date(year, monthStartingAtZero, dayOfMonth);`
 - `var now = new Date();` //is automatically set to now
 - `now.getMonth();`
- Complete and debug code that uses built-in Math functions
 - Random; round; abs; floor; ceiling; min; max; pow; sqrt
 - [JavaScript built-in math](#)
- Complete and debug a function that accepts parameters and returns a value
 - Reusable code; local versus global scope, redefine variables, pass parameters, value versus reference, return values

[Implement and Analyze Decisions and Loops \(20-25%\)](#)

- Evaluate expressions that use logical and comparison operators
 - `==; !=; <; >; <=; >=; !; &&; ||`
- Complete and debug decision statements
 - `if; else if; switch; nested if`
 - [Control structure documentation](#)
- Complete and debug loops
 - `for; while; do; break; continue`

- [basic loop documentation](#)

[Interact with the Document Object Model \(15-20%\)](#)

[For these use the Mozilla Document Object Model documentation](#)

- Identify and construct the Document Object Model (DOM) tree
 - window; document; body; other HTML elements
 - Learn DOM at [Code Academy DOM](#) or at [Mozilla DOM](#)
- Identify and handle HTML events
 - onchange; onmouseover; onload; onclick; onmouseout; onkeydown
- Complete and debug code that outputs to an HTML document
 - innerHTML; document.write
- Complete and debug code that locates, modifies, and adds HTML elements and attributes
 - getElementById; getElementsByTagName; getElementsByClassName; setAttribute; createElement
 - [getElementById](#)

[Interact with HTML Forms \(5-10%\)](#)

- Complete and debug code that retrieves input from forms and sets form field values
 - Retrieve form values; identify the DOM path; get values from different types of elements; prepopulate values; mask values
- Complete and debug code that performs input validation
 - Case; string comparisons; Not-A-Number (NaN)
- Describe the form submission process
 - onsubmit; post versus get; potential targets for submission
 - [Form submission](#)

What you need to know:

Syntax:

[for; while; do; break; continue](#)

[switch\(\){case:break:}](#)

[%=](#)

[document.getElementById\("id"\).innerHTML = "Inner HTML";](#)

[Random: round: abs: floor: ceiling: min: max: pow: sqrt](#)

Arrays VVVV:

```
var array = ["a", "b", "c"];
```

```
array.unshift("at start");
```

```
array.push("at end")
```

Will give you ["at start", "a", "b", "c", "at end"]

```
array.pop();
```

```
array.shift();
```

Will give you ["a", "b", "c"]

```
array.sort();
```

Date:

```
var date = new Date(year, monthStartingAtZero, dayOfMonth);
```

```
var now = new Date();//is automatically set to now
```

```
now.getMonth();
```

If you don't know JS, learn on [Code Academy](#) or this massive [List of sites to learn on](#)

Once you think your ready: [W3 Schools Quiz](#)

Windows Operating System Fundamentals

Understanding operating system configurations (15–20%)

- Configure Control Panel options
 - Configure administrative tools, configure accessibility options, configure power settings, configure File Explorer settings
- Configure desktop settings
 - Configure profiles, display settings, and shortcuts; configure and customize Start Menu; configure Task Bar settings; configure toolbars; configure notifications
- Configure native applications and tools
 - Configure Microsoft Edge, configure Cortana, configure Hyper-V, configure settings using MSCONFIG, configure processes and applications using Task Manager, configure computer management
- Configure mobility settings
 - Configure Sync Center, Windows Mobility Center, and Remote Desktop

- Configure and use management tools
 - Configure MMC, configure the Windows PowerShell console and Windows PowerShell ISE

Preparation resources

- [Administrative Tools in Windows 10](#)
- [Manage Windows 10 Start and taskbar layout](#)
- [Windows 10 Start Layout Customization](#)
- [Configure Windows 10 Taskbar](#)
- [Cool Stuff about PowerShell 5.0 in Windows 10](#)

Installing and upgrading client systems (15–20%)

- Identify Windows operating system editions
 - Identify hardware and compatibility requirements, determine appropriate editions per device type
- Identify upgrade paths
 - Identify upgrade paths from previous Windows versions, identify application compatibility
- Understand installation types
 - Perform a clean install, upgrade via Windows Update, migrate from previous Windows versions, perform removable media installation, perform network installation
- Understand operating system architecture
 - Understand kernel mode; understand user mode; understand memory, IRQs, drivers, CPUs, and UI; understand 32-bit versus 64-bit architecture

Preparation resources

- [Windows 10 Upgrade Path](#)
- [Windows 10 Compatibility](#)
- [Windows 10 deployment considerations](#)
- [Architecture of Windows 10](#)

Managing applications (15–20%)

- Configure applications
 - Understand local versus network applications, configure desktop applications, configure app startup options, configure Windows features, configure Windows SmartScreen, configure application removal, configure Windows Store apps
- Configure user account control (UAC)
 - Understand standard user versus administrative user, understand types of UAC prompts and levels
- Configure antivirus settings
 - Configure Windows Defender and the Malicious Software Removal tool
- Understand services
 - Understand service start-up types, service accounts, and service dependencies

Preparation resources

- [User Account Control](#)
- [Windows Defender Advanced Threat Protection](#)

- [Windows Defender in Windows 10](#)
- [Microsoft Malicious Software Removal Tool](#)

Managing files and folders (15–20%)

- Understand file systems
 - Understand FAT32, NTFS, and ReFS
- Understand file and print sharing
 - Configure File System permissions, configure Share Permissions, configure HomeGroup settings, configure print drivers, configure effective permissions, create shares, create mapped drives
- Understand encryption
 - Understand BitLocker and Encrypting File Systems (EFS)
- Understand libraries
 - Understand offline files, configure libraries, add multiple local locations to a library, add networked locations

Preparation resources

- [BitLocker](#)
- [HomeGroup from start to finish](#)

Managing devices (15–20%)

- Connect devices
 - Connect, enable, and disable plug-and-play and Bluetooth devices; connect and disconnect printers; install third-party software for devices; install device drivers
- Understand storage
 - Understand disk types, security (encryption), storage device types (eSATA, USB, IEEE 1394, iSCSI, InfiniBand), and storage drive types (basic, primary, extended, logical, dynamic disk, VHDs); configure OneDrive use; optimize drives
- Understand printing devices
 - Understand local printers, network printers, print queues, print-to-file, and Internet printing
- Understand system devices
 - Understand video, audio, and infrared input devices; understand Device Manager

Preparation resources

- [Storage Spaces in Windows 10](#)
- [Whats New in Windows 10](#)

Understanding operating system maintenance (15–20%)

- Understand backup and recovery methods
 - Configure System Restore; configure a recovery drive; configure recovery boot options, such as Last Known Good, and various Safe Mode options; recover files from OneDrive
- Understand maintenance tools
 - Understand Disk Defragmenter, Disk Cleanup, Task Scheduler, and System Information
- Configure updates

- Configure Windows Update options; implement Insider Preview, Current Branch, and Current Branch for Business scenarios; manage update history; roll back updates

Preparation resources

- [Windows 10 Servicing Options](#)

Introduction to Programming Using Java

Understand Java fundamentals (15-20%)

- Describe the use of main in a Java application
 - Signature of main, why it is static; how to consume an instance of your own class; command-line arguments
- Perform basic input and output using standard packages
 - Print statements; import and use the Scanner class
- Evaluate the scope of a variable
 - Declare a variable within a block, class, or method

Work with data types, variables, and expressions (40-45%)

- Declare and use primitive data type variables
 - Data types, including byte, char, int, double, short, long, float, boolean; identify when precision is lost; initialization; how primitives differ from wrapper object types such as Integer and Boolean
- Construct and evaluate code that manipulates strings
 - String class and string literals, comparisons, concatenation, case and length; String.format methods; string operators; converting a primitive data type to a string; the immutable nature of strings; initialization; null
- Construct and evaluate code that creates, iterates, and manipulates arrays and array lists
 - One- and two-dimensional arrays, including initialization, null, size, iterating elements, accessing elements; array lists, including adding and removing elements, traversing the list
- Construct and evaluate code that performs parsing, casting and conversion
 - Implementing code that casts between primitive data types, converts primitive types to equivalent object types, or parses strings to numbers
- Construct and evaluate arithmetic expressions
 - Arithmetic operators, assignment, compound assignment operators, operator precedence

Implement flow control (15-20%)

- Construct and evaluate code that uses branching statements
 - if, else, else if, switch; single-line versus block; nesting; logical and relational operators
- Construct and evaluate code that uses loops
 - while, for, for each, do while; break and continue; nesting; logical, relational, and unary operators

Perform object-oriented programming (10-15%)

- Construct and evaluate a class definition
 - Constructors; constructor overloading; one class per .java file; this keyword; inheritance and overriding at a basic level
- Declare, implement, and access data members in a class
 - private, public, protected; instance data members; static data members; using static final to create constants; describe encapsulation
- Declare, implement, and access methods
 - private, public, protected; method parameters; return type; void; return value; instance methods; static methods; overloading
- Instantiate and use a class object in a program
 - Instantiation; initialization; null; accessing and modifying data members; accessing methods; accessing and modifying static members; importing packages and classes

Compile and debug code (5-10%)

- Troubleshoot syntax errors, logic errors, and runtime errors
 - print statement debugging; output from the javac command; analyzing code for logic errors; console exceptions after running the program; evaluating a stack trace
- Implement exception handling
 - try catch finally; exception class; exception class types; display exception information

Exam 98-367 Security Fundamentals

Understand security layers (25–30%)

- Understand core security principles
 - Confidentiality; integrity; availability; how threat and risk impact principles; principle of least privilege; social engineering; attack surface analysis; threat modelling
- Understand physical security
 - Site security; computer security; removable devices and drives; access control; mobile device security; keyloggers
- Understand Internet security
 - Browser security settings; secure websites
- Understand wireless security
 - Advantages and disadvantages of specific security types; keys; service set identifiers (SSIDs); MAC filters

Preparation resources

- [Windows Server 2008 in an organization's defense in depth strategy](#)
- [Secure Windows Server](#)
- [Using Windows Server 2008: Controlling communication with the Internet](#)

Understand operating system security (35-40%)

- Understand user authentication
 - Multifactor authentication; physical and virtual smart cards; Remote Authentication Dial-In User Service (RADIUS); biometrics; use Run As to perform administrative tasks
- Understand permissions
 - File system permissions; share permissions; registry; Active Directory; enable or disable inheritance; behavior when moving or copying files within the same disk or on another disk; multiple groups with different permissions; basic permissions and advanced permissions; take ownership; delegation; inheritance
- Understand password policies
 - Password complexity; account lockout; password length; password history; time between password changes; enforce by using Group Policies; common attack methods; password reset procedures; protect domain user account passwords
- Understand audit policies

- Types of auditing; what can be audited; enable auditing; what to audit for specific purposes; where to save audit information; how to secure audit information
- Understand encryption
 - Encrypting file system (EFS); how EFS-encrypted folders impact moving/copying files; BitLocker (To Go); TPM; software-based encryption; MAIL encryption and signing and other uses; virtual private network (VPN); public key/private key; encryption algorithms; certificate properties; certificate services; PKI/certificate services infrastructure; token devices; lock down devices to run only trusted applications
- Understand malware
 - Buffer overflow; viruses, polymorphic viruses; worms; Trojan horses; spyware; ransomware; adware; rootkits; backdoors; zero day attacks

Preparation resources

- [Windows authentication](#)
- [Password policy](#)
- [Audit policies](#)

[Understand network security \(20–25%\)](#)

- Understand dedicated firewalls
 - Types of hardware firewalls and their characteristics; when to use a hardware firewall instead of a software firewall; stateful versus stateless firewall inspection; Security Compliance Manager; security baselines
- Understand network isolation
 - Routing; honeypot; perimeter networks; network address translation (NAT); VPN; IPsec; server and domain isolation
- Understand protocol security
 - Protocol spoofing; IPsec; tunneling; DNSsec; network sniffing; denial-of-service (DoS) attacks; common attack methods

Preparation resources

- [Windows Firewall](#)
- [Network Access Protection](#)
- [IPsec](#)

[Understand security software \(15–20%\)](#)

- Understand client protection

- Antivirus; protect against unwanted software installations; User Account Control (UAC); keep client operating system and software updated; encrypt offline folders, software restriction policies; principle of least privilege
- Understand email protection
 - Antispam, antivirus, spoofing, phishing, and pharming; client versus server protection; Sender Policy Framework (SPF) records; PTR records
- Understand server protection
 - Separation of services; hardening; keep server updated; secure dynamic Domain Name System (DNS) updates; disable unsecure authentication protocols; Read-Only Domain Controllers (RODC)

Preparation resources

- [What's new for operating system hardening and integrity for Windows Server 2008](#)
- [Software restriction policies](#)
- [What's new for server protection in Windows Server 2008](#)

[Folder Here](#)

MTA Networking Quizlets (Pauley):

- D107 MTA Networking Fundamentals 98-366 (OSI Model): (https://quizlet.com/_6fov3)
- D107 MTA Networking Fundamentals 98-366 (Protocols): (https://quizlet.com/_6esj4r)
- D107 MTA Networking Fundamentals 98-366 (Hubs, Switches, and Routers): (https://quizlet.com/_6fovjl)
- D107 MTA Networking Fundamentals 98-366 (Network Topology and Media): (https://quizlet.com/_6fovz6)
- D107 MTA Networking Fundamentals 98-366 (Misc Networking): (https://quizlet.com/_6fovoji)

Section 1: Introduction

- [Instructor Introduction](#) (6:55)
- [Course Prerequisites](#) (5:07) (.pdf)
- [Sharing My Certificate](#) (1:45)
- [Why MTA Certification](#) (6:51)
- [Exam Objectives](#) (5:24) (.pdf)
- [Level Setting Expectations](#) (2:05)
- [Course Live Demonstrations](#) (1:10)
- [Course Quizzes And Exams](#) (0:45)

Section 2: MTA 98-367 Study Strategies

[Study Flash Cards](#) (1:36) (.doc)

[Study Timeline Calculator & Recommendations](#) (5:18) (.xlsx)

[General Study & Exam Prep Strategies](#) (3:50) (.pdf)

Section 3: Course Pre-Assessment Quiz

[Course Pre-Assessment Quiz Overview](#) (0:57)

[Course Pre-Assessment Quiz](#)

Section 4: Setting Up Your Virtualized Test Environment

[Introduction](#) (3:44)

[Virtualization Requirements and Install Overview](#) (5:58)

[Downloading Operating System ISO Files](#) (3:06)

[Creating Virtualbox Virtual Machines](#) (8:39)

[Some Additional Virtual Machine Settings](#) (6:32)

[Installing Windows 10 Enterprise Operating System](#) (11:20)

[Installing Windows Server 2016 Operating System](#) (8:06)

[Installing Windows Updates](#) (4:08)

[Setting Up Our Sandbox Internal LAN](#) (7:11)

[Introduction to Active Directory](#) (9:33)

[Active Directory Installation & Configuration](#) (9:12)

[DHCP Installation & Scope Configuration](#) (8:26)

Section 5: Domain 1 Overview: Understanding Security Layers

[Domain 1 Overview](#) (1:31)

Section 6: Domain 1 – Part 1: Understanding Core Security Principles

Section Introduction

[The CIA Triad](#) (5:59) (Pdf)

[Authentication, Authorization, and Accounting \(AAA\)](#) (4:08) (Pdf)

[Defense in Depth](#) (5:18) (Pdf)

[Least Privilege](#) (3:41) (Pdf)

[Risk Management](#) (18:38) (Pdf)

[Exploring Risk and Threats](#) (6:12) (Pdf)

[Attack Surface Analysis \(9:10\) \(pdf\)](#)

[Social Engineering \(9:11\) \(Pdf\)](#)

[Understanding Core Security Principles Quiz](#)

Section 7: Domain 1 – Part 2: Understanding Physical Security

Section Introduction

[Introduction to Access Control \(10:20\) \(Pdf\)](#)

[Physical Site Security \(7:58\) \(Pdf\)](#)

[Computer Equipment Physical Security \(7:47\) \(Pdf\)](#)

[Removable & Mobile Device Security \(11:23\) \(Pdf\)](#)

[Keyloggers \(6:40\) \(Pdf\)](#)

[Understanding Physical Security Quiz](#)

Section 8: Domain 1 – Part 3: Understanding Internet Security

Section Introduction

[Section Overview \(0:53\)](#)

[Live Demo: Microsoft Edge Security Settings \(8:18\)](#)

[Live Demo: Internet Explorer Browser Settings \(19:15\) \(pdf\)](#)

[Start Understanding Internet Security Quiz](#)

Section 9: Domain 1 – Part 4: Understanding Wireless Security

Section Introduction

[Wireless Encryption Standards Overview \(2:50\) \(Pdf\)](#)

[Wireless Equivalent Privacy \(WEP\) \(3:27\)](#)

[Wi-Fi Protected Access \(WPA\) \(2:38\)](#)

[Wi-Fi Protected Access 2 \(WPA2\) \(3:00\)](#)

[Wi-Fi Protected Access 3 \(WPA3\) \(3:54\)](#)

[WPA Personal versus Enterprise Mode \(3:20\)](#)

[Live Demo: Wireless Router Configuration \(13:03\)](#)

[Understanding Wireless Security Quiz](#)

Section 10: Domain 2 Overview: Understanding Operating System Security

[Domain 2 Overview \(1:30\)](#)

Section 11: Domain 2 – Part 1: Understanding User Authentication

[Authentication Basics \(13:36\) \(Pdf\)](#)
[Physical and Virtual Smart Cards \(4:25\) \(Pdf\)](#)
[Remote Authentication Dial-In User Service \(RADIUS\) \(7:46\) \(pdf\)](#)
[Live Demo: Windows Server 2016 RADIUS Server \(9:50\)](#)
[Live Demo: Using Run As to Perform Administrative Tasks \(11:51\)](#)
[Understanding User Authentication Quiz](#)

Section 12: Domain 2 – Part 2: Understanding Permissions

[Basic NT File System \(NTFS\) Permissions \(9:15\) \(Pdf\)](#)
[NTFS Advanced Permissions \(4:39\) \(Pdf\)](#)
[NTFS Inheritance \(3:40\) \(Pdf\)](#)
[Permissions Behavior when Moving and Copying Files \(3:40\) \(Pdf\)](#)
[Share Permissions \(7:16\) \(Pdf\)](#)
[Active Directory Groups \(7:01\) \(Pdf\)](#)
[Active Directory Organizational Units \(3:24\) \(Pdf\)](#)
[The Registry \(6:29\) \(Pdf\)](#)
[Live Demo: The Registry \(4:22\)](#)
[Live Demo: Basic NTFS Permissions \(4:04\)](#)
[Live Demo: Inheritance and Changing Ownership \(8:42\)](#)
[Live Demo: Effective Access \(2:45\)](#)
[Live Demo: Advanced NTFS Permissions \(2:36\)](#)
[Live Demo: Network Share and NTFS Permissions \(12:23\)](#)
[Live Demo: Moving and Copying Files and Folders \(11:18\)](#)
[Live Demo: Active Directory Groups \(6:58\)](#)
[Live Demo: Active Directory AD OUs, Delegation and Group Policy Management \(13:45\)](#)
[Live Demo: Active Directory Permissions \(7:25\)](#)
[Understanding Permissions Quiz](#)

Section 13: Domain 2 – Part 3: Understanding Password Policies

[Understanding Microsoft Password Policies \(12:47\) \(Pdf\)](#)
[Understanding Microsoft Account Lockout Policies \(8:53\) \(Pdf\)](#)
[Common Password Attack Methods \(6:04\) \(Pdf\)](#)
[Live Demo: Password and Account Lockout Settings Demonstration \(8:37\)](#)
[Understanding Password Policies Quiz](#)

Section 14: Domain 2 – Part 4: Understanding Audit Policies

[Introduction to Auditing](#) (12:15)
[What Can Be Audited](#) (10:30) (pdf)
[Saving and Securing Audit Information](#) (4:17) (Pdf)
[Auditing Best Practices and Recommendations](#) (7:49) (PDF)
[Live Demo: Auditing](#) (9:46)
[Understanding Audit Policies Quiz](#)

Section 15: Domain 2 – Part 5: Understanding Encryption

[Introduction to Cryptography](#) (5:43) (Pdf)
[Symmetric Encryption](#) (9:08) (Pdf)
[Asymmetric Encryption](#) (4:38) (Pdf)
[Hashing Algorithms](#) (8:43) (Pdf)
[Digital Certificates and Certificate Authorities](#) (8:50) (Pdf)
[Live Demo: Digital Certificates](#) (5:19)
[Email Encryption Use Cases](#) (12:55) (Pdf)
[Windows Encrypting File System \(EFS\) Overview](#) (6:16) (Pdf)
[Live Demo: Windows EFS](#) (17:02)
[Live Demo: Moving and Copying Files/Folders with EFS](#) (7:01)
[BitLocker and BitLocker To Go Overview](#) (6:55) (Pdf)
[Live Demo: BitLocker](#) (8:09)
[Software versus Hardware Based Encryption](#) (4:29) (Pdf)
[Security Token Devices](#) (2:16) (pdf)
[Live Demo: Run Only Trusted Applications via Software Restriction Policies](#) (2:51)
A Quick Note on VPNs
[Understanding Encryption Quiz](#)

Section 16: Domain 2 – Part 6: Understanding Malware

[Introduction to Malware](#) (1:11) (Pdf)
[Buffer Overflows](#) (6:06) (Pdf)
[Viruses & Polymorphic Viruses](#) (4:54) (Pdf)
[Worms](#) (2:18) (Pdf)
[Trojan Horses](#) (2:04) (Pdf)
[Spyware and Adware](#) (3:34) (Pdf)
[Ransomware](#) (4:16)
[Rootkits](#) (4:04) (Pdf)
[Backdoors](#) (4:03)

[Zero Day Attacks \(2:30\) \(Pdf\)](#)
[Understanding Malware Quiz](#)

Section 17: Domain 3 Overview: Understanding Network Security

[Domain 3 Overview \(1:00\)](#)

Section 18: Domain 3 – Part 1: Understanding Dedicated Firewalls

[Hardware Firewalls \(10:59\) \(Pdf\)](#)
[Stateful versus Stateless Firewalls \(2:57\) \(pdf\)](#)
[Using Hardware versus Software Firewalls \(3:29\) \(Pdf\)](#)
[Live Demo: Windows Firewall \(4:49\)](#)
[Live Demo: Microsoft Security Compliance Manager \(9:10\) \(Pdf\)](#)
[Understanding Dedicated Firewalls Quiz](#)

Section 19: Domain 3 – Part 2: Understanding Network Isolation

[Understanding Network Isolation \(1:26\) \(Pdf\)](#)
[Perimeter Networks \(5:57\) \(Pdf\)](#)
[Switch VLANs \(5:36\) \(pdf\)](#)
[Routers \(3:20\) \(Pdf\)](#)
[Honeypots \(2:20\) \(pdf\)](#)
[Network Address Translation \(NAT\) \(7:09\) \(Pdf\)](#)
[Virtual Private Network \(VPN\) \(9:24\) \(Pdf\)](#)
[IPSec \(4:35\) \(Pdf\)](#)
[VPN, IPSec and Network Isolation \(1:13\) \(Pdf\)](#)
[Live Demo: Server and Domain Isolation \(7:15\)](#)
[Understanding Network Isolation Quiz](#)

Section 20: Domain 3 – Part 3: Understanding Protocol Security

[Protocol Spoofing \(7:34\) \(pdf\)](#)
[Live Demo: Network Sniffing with WireShark \(6:01\)](#)
[Common Attack Methods \(10:26\) \(Pdf\)](#)
[DNSsec \(2:29\) \(Pdf\)](#)
[Live Demo: Enabling DNSsec on Windows Server 2016 \(2:57\)](#)
[Understanding Protocol Security Quiz](#)

Section 21: Domain 4 Overview: Understanding Security Software

[Domain 4 Overview](#) (1:04)

Section 22: Domain 4 – Part 1: Understanding Client Protection

[User Account Control \(UAC\)](#) (4:06) [\(Pdf\)](#)

[Live Demo: How to Adjust UAC Settings in Windows](#) (5:14)

[Live Demo: AppLocker & Software Restriction Policies](#) (9:22)

[Live Demo: Windows Defender Antivirus](#) (7:02)

[Keeping Windows OS and Software Updated](#) (6:42) [\(Pdf\)](#)

[Live Demo: Encrypting Offline Folders](#) (2:13)

[Understanding Client Protection Quiz](#)

Section 23: Domain 4 – Part 2: Understanding Email Protection

[Email Spam, Spoofing, Phishing and Pharming](#) (14:56) [\(Pdf\)](#)

[Dealing with Unwanted Emails](#) (11:37) [\(Pdf\)](#)

[Client versus Server Protection](#) (2:13) [\(Pdf\)](#)

[Understanding Email Protection Quiz](#)

Section 24: Domain 4 – Part 3: Understanding Server Protection

[Server Hardening](#) (3:50) [\(Pdf\)](#)

[Separation of Services](#) (2:26) [\(Pdf\)](#)

[Keeping Servers Updated](#) (2:13) [\(Pdf\)](#)

[Secure Dynamic Domain Name System \(DNS\) Updates](#) (3:10) [\(Pdf\)](#)

[Live Demo: Secure Dynamic DNS](#) (1:20)

[Disabling Unsecure Authentication Protocols](#) (2:48)

[Read-Only Domain Controllers \(RODC\)](#) (3:31) [\(Pdf\)](#)

[Understanding Server Protection Quiz](#)

Section 25: Scheduling & Taking Your MTA 98-367 Exam

[Exam Details & Cost](#) (3:50)

[Scheduling Your Exam](#) (6:02)

[Testing Center Expectations](#) (3:03)

Section 26: Practice Exams

[Practice Exams Overview](#) (9:02)

Practice Exam #1
Practice Exam #2

Section 27: Next Steps

[Congratulations & Thank You!](#) (1:00)
[Certification Next Steps](#) (1:11)

Exam 98-366 Networking Fundamentals

[Folder Here](#)

MTA Networking Quizlets (Pauley):

D107 MTA Networking Fundamentals 98-366 (OSI Model): (https://quizlet.com/_6fov3)
D107 MTA Networking Fundamentals 98-366 (Protocols): (https://quizlet.com/_6esj4r)
D107 MTA Networking Fundamentals 98-366 (Hubs, Switches, and Routers): (https://quizlet.com/_6fovjl)
D107 MTA Networking Fundamentals 98-366 (Network Topology and Media): (https://quizlet.com/_6fovz6)
D107 MTA Networking Fundamentals 98-366 (Misc Networking): (https://quizlet.com/_6fovoji)

1: Course Introduction

[Instructor Introduction](#) (5:45)
[Sharing my Microsoft MTA 98-366 Exam Transcript & Certification Certificate](#) (1:46)
[Why MTA Certifications?](#) (5:45) ([pdf](#))
[Microsoft MTA 98-366 Exam Objectives](#) (5:47) ([pdf](#))
[Level-Setting Expectations: Entry Level Certification](#) (1:33)
[Course Curriculum Overview](#) (5:37)
[Course Pre-Assessment, Section Quizzes and Practice Exams](#) (0:56)

Section 2: MTA 98-366 Study Strategies

[Study Flash Cards](#) (1:54) ([.Docx](#))

[Study Timeline Calculator & Recommendation](#) (5:29) (.xlsx)
[General Study & Exam Prep Strategies](#) (6:32) (.pdf)

Section 3: Course Pre-Assessment Quiz

[Course Pre-Assessment Quiz Overview](#) (1:40)

Section 4: Overview of Computer Networks

[What is a Computer Network? \(1:43\)](#) (.pdf)
[Types of Networks](#) (13:02) (.pdf)
[Some Basic Computer Networking Rules](#) (2:35) (.pdf)
[Why Build a Computer Network?](#) (7:07) (.pdf)
[Overview of Computer Networks Section Quiz](#)

Section 5: How Computer Networks Work

[Section Introduction](#) (1:03)
[Introduction to Computer Networking Protocols](#) (4:38) (.pdf)
[Introduction to the OSI Model](#) (11:27) (.pdf)
[Introduction to the TCP/IP Model](#) (3:52) (.pdf)
[Introduction to MAC Addresses](#) (5:54) (.pdf)
[Introduction to IP Addresses](#) (3:40) (.pdf)
[IP versus MAC Addresses](#) (2:08) (.pdf)
[Introduction to Ethernet](#) (3:32) (.pdf)
[Duplex Communication](#) (3:26) (.pdf)
[How Computer Networks Work Section Quiz](#)

Section 6: Network Topologies

[Introduction to Network Topologies](#) (2:25) (.pdf)
[Physical Network Topologies](#) (12:55) (.pdf)
[Network Topologies Section Quiz](#)

Section 7: Introduction to Networking Components

[Section Introduction](#) (1:19)
[Network Interface Cards \(Wired & Wireless\)](#) (3:35) (.pdf)
[Hubs](#) (3:25) (.pdf)
[Switches](#) (4:30) (.pdf)
[Hubs versus Switches](#) (2:03) (.pdf)
[Wireless Access Points](#) (3:34) (.pdf)
[Routers](#) (5:05) (.pdf)
[Firewalls](#) (3:25) (.pdf)
[DHCP Servers](#) (3:09) (.pdf)
[Networking Components Section Quiz](#)

Section 8: Network Cabling

[Section Introduction](#) (0:24)
[Types of Network Cabling](#) (1:26) (.pdf)
[Ethernet](#) (5:48) (.pdf)
[Twisted Pair Copper Network Cabling](#) (11:06) (.pdf)
[Wiring Standards: 568A & 568B and Cable Types](#) (4:38) (.pdf)
[Plenum-Rated Cabling](#) (3:57) (.pdf)
[Fiber Optic Network Cabling](#) (7:49) (.pdf)
[Network Cabling Section Quiz](#)

Section 9: The OSI Model

[Section Introduction](#) (0:19)
[OSI Layer 7: The Application Layer](#) (2:06) (Pdf)
[OSI Layer 6: The Presentation Layer](#) (4:02) (Pdf)
[OSI Layer 5: The Session Layer](#) (1:57) (Pdf)
[OSI Layer 4: The Transport Layer](#) (3:49) (Pdf)
[OSI Layer 3: The Network Layer](#) (2:55) (Pdf)
[OSI Layer 2: The Data Link Layer](#) (2:17) (pdf)
[OSI Layer 1: The Physical Layer](#) (2:44) (pdf)
[OSI Model Section Quiz](#)

Section 10: Introduction to TCP/IP

[Section Introduction](#) (1:14)

[The TCP/IP Protocol Suite \(3:00\) \(Pdf\)](#)
[TCP/IP Protocols \(3:16\) \(Pdf\)](#)
[Types of Communication \(3:56\) \(Pdf\)](#)
[Access Methods: CSMA/CD, CSMA/CA, Token Passing \(4:16\) \(Pdf\)](#)

Section 11: TCP/IP Internet Layer Protocols

[Address Resolution Protocol \(ARP\) \(3:48\) \(Pdf\)](#)
[Internet Protocol \(IPv4\) \(5:13\) \(Pdf\)](#)
[Internet Protocol \(IPv6\) \(3:35\) \(Pdf\)](#)
[Internet Control Message Protocol \(ICMP\) \(4:05\) \(Pdf\)](#)

Section 12: TCP/IP Transport Layer Protocols

[Transmission Control Protocol \(TCP\) \(9:16\) \(Pdf\)](#)
[User Datagram Protocol \(UDP\) \(5:18\) \(Pdf\)](#)
[Understanding Ports \(10:47\) \(Pdf\)](#)

Section 13: Common TCP/IP Application Layer Protocols

[Management Protocols \(5:47\) \(Pdf\)](#)
[Remote Communication Protocols \(3:51\) \(pdf\)](#)
[File Transfer Application Protocols \(7:06\) \(Pdf\)](#)
[Mail Application Protocols \(5:52\) \(Pdf\)](#)
[Browser Application Protocols \(6:44\) \(Pdf\)](#)
[TCP/IP Section Quiz](#)

Section 14: IP Addressing Version 4 (IPv4) Fundamentals

[Section Introduction \(6:10\)](#)
[Understanding IPv4 Addresses \(13:31\) \(Pdf\)](#)
[IPv4 Address Anatomy \(2:38\) \(pdf\)](#)
[IPv4 Fundamentals Section Quiz](#)

Section 15: Basics of Binary Math

[Binary Math Basics \(18:44\) \(pdf\)](#)

Binary Math Student Exercise [Worksheet](#)

Section 16: A Deeper Dive into IPv4 Addresses and Subnet Masks

[Section Introduction](#) (0:38)

[IP Address Classification and Subnet Masks](#) (19:56)

[The Power of 2's](#) (5:25) (Pdf)

[Public versus Private IP Addresses](#) (9:28) (Pdf)

[Loopback IP Address](#) (4:23) (Pdf)

[APIPA](#) (4:35) (pdf)

[IPv4 Addresses and Subnet Masks Section Quiz](#)

Section 17: Introduction to IPv4 Subnetting

[Introduction to Subnetting](#) (19:54) (Pdf)

[Subnetting Reference Tables](#) (3:05) (Pdf)

Section 18: Subnetting Class C IPv4 Networks

[Section Introduction](#) (0:31)

[Subnetting a Class C Network: Example #1](#) (6:58) (Pdf)

[Subnetting Example #1 Detailed Discussion](#) (9:01) (Pdf)

[Subnetting a Class C Network: Example #2](#) (8:20) (Pdf)

[Subnetting Example #2 Detailed Discussion](#) (7:25) (Pdf)

[Subnetting a Class C Network: Example #3](#) (8:39) (Pdf)

[Subnetting Example #3 Detailed Discussion](#) (3:58) (Pdf)

[Subnetting Class C Networks Student Practice Worksheet](#)

[Answers Walk-Through: Student Practice Worksheet](#) (15:47) (Pdf)

Section 19: IP Addressing Version 6 (IPv6) Fundamentals

[Section Introduction](#) (5:00) (pdf)

[Disadvantages of IPv4 and IPv6 Solutions](#) (12:11)

[IPv6 Addressing](#) (5:53) (Pdf)

[Types of IPv6 Addresses](#) (6:51) (Pdf)

[IPv4 to IPv6 Translation & Compatibility](#) (6:59) (Pdf)

[IPv6 Fundamentals Section Quiz](#)

Section 20: Understanding Switches

[Section Introduction](#) (0:41)

[Refresher on Hubs vs Switches](#) (5:53) (Pdf)

[Collision Domains & CSMA/CD](#) (9:26) (pdf)

[Switch Form Factor](#) (1:52) (Pdf)

[Managed versus Unmanaged Switches](#) (2:23) (Pdf)

[How a Switch Learns MAC Addresses](#) (5:27) (Pdf)

[Virtual LANs \(VLANs\)](#) (6:04) (Pdf)

[Layer 3 Switches](#) (3:22) (Pdf)

[Switch Speed](#) (11:16) (Pdf)

[Switch Security](#) (7:10) (Pdf)

[Understanding Switches Section Quiz](#)

Section 21: Understanding Routers

[Section Introduction](#) (1:14)

[Refresher on Routers](#) (2:29) (Pdf)

[Routing Tables](#) (4:48)

[EGPs & IGPs](#) (10:08) (Pdf)

[Static versus Dynamic Routing](#) (4:26) (Pdf)

[The Next Hop](#) (4:38) (Pdf)

[Interior Gateway Protocols \(IGPs\): RIP, RIPv2 & OSPF](#) (6:28) (Pdf)

[Exterior Gateway Protocols \(EGPs\) - BGP](#) (2:07) (Pdf)

[Default Routes](#) (3:21) (Pdf)

[Directly Connected Routes](#) (2:42) (Pdf)

[Transmission Speeds](#) (5:59) (Pdf)

[Network Address Translation \(NAT\)](#) (9:45) (Pdf)

[Installing & Configuring Routing on Windows Server 2016](#) (15:18) (Pdf)

[Cisco Packet Tracer Demonstration](#) (4:28)

[Understanding Routers Section Quiz](#)

Section 22: Windows Networking Services

[Section Introduction](#) (1:26)

Windows Server 2016 Virtualization Prerequisites

[How to a Install Window Server 2016 Virtual Machine](#) (18:54)

[How to Install Active Directory](#) (10:00)

[Installing & Configuring Dynamic Host Configuration Protocol \(DHCP\)](#) (24:23) [\(Pdf\)](#)

[Revisiting Automatic Private IP Addressing \(APIPA\)](#) (3:45)

[Exploring Domain Naming System \(DNS\)](#) (12:37)

[Introducing Remote Desktop Services \(RDS\)](#) (8:02)

[Revisiting Routing and Remote Access Service \(RRAS\)](#) (9:39)

[Windows Networking Services Section Quiz](#)

Section 23: Understanding the Name Resolution Process

[Section Introduction](#) (0:37)

[Revisiting DNS](#) (14:43) [\(Pdf\)](#)

[Name Resolution Process Overview](#) (1:58) [\(Pdf\)](#)

[NetBIOS vs Host Names](#) (8:03)

[Name Resolution Step 1: Host Cache & Hosts File](#) (9:53)

[Name Resolution Step 2: DNS Query](#) (2:34)

[Name Resolution Step 3: NetBIOS Cache](#) (4:57)

[Name Resolution Step 4: WINS Server Query](#) (4:13)

[Name Resolution Step 5: Local Subnet Broadcast](#) (2:57)

[Name Resolution Step 6: LMHosts File](#) (4:17)

[Name Resolution Section Quiz](#)

Section 24: Wireless Networking

[Section Introduction](#) (1:02)

[Wireless Networking Fundamentals](#) (17:59) [\(Pdf\)](#)

[Wireless Networking 802.11 Standards](#) (7:50) [\(Pdf\)](#)

[Signal Degradation](#) (6:02) [\(Pdf\)](#)

[Wireless Network Vulnerabilities & Security](#) (8:51) [\(Pdf\)](#)

[Wireless Networking Security Standards](#) (11:17) [\(Pdf\)](#)

[Wireless Networking Section Quiz](#)

Section 25: Wide Area Networking

[Section Introduction](#) (0:57)

[WAN Technologies Overview \(3:43\) \(Pdf\)](#)
[Circuit Switching vs. Packet Switching \(9:58\) \(Pdf\)](#)
[POTS/PSTN Dial-Up \(7:13\) \(Pdf\)](#)
[ISDN \(8:14\) \(Pdf\)](#)
[T-Carrier & E-Carrier Digital Signal Lines \(6:21\) \(Pdf\)](#)
[DSL \(8:25\) \(Pdf\)](#)
[Broadband Cable \(11:08\) \(Pdf\)](#)
[X.25, Frame Relay and ATM \(6:15\) \(Pdf\)](#)
[FDDI and SONET \(4:23\) \(Pdf\)](#)
[Wide Area Networking Section Quiz](#)

Section 26: Network Security

[Section Introduction \(1:06\)](#)
[Intranet, Extranet and the Internet \(7:21\) \(pdf\)](#)
[Perimeter Networks: DMZ \(6:32\) \(Pdf\)](#)
[Firewalls \(6:54\) \(Pdf\)](#)
[Windows Firewall Demonstration \(4:51\)](#)
[Proxy Servers \(5:35\) \(Pdf\)](#)
[Network Intrusion Detection & Prevention \(7:00\) \(pdf\)](#)
[VPN \(10:55\) \(Pdf\)](#)
[IPSec \(4:56\) \(Pdf\)](#)
[Microsoft Internet Security Zones \(4:41\)](#)
[Network Security Section Quiz](#)

Section 27: TCP/IP Command Line Tools

[Section Introduction \(0:59\)](#)
[Accessing & Using Command Prompt \(6:19\)](#)
[Telnet \(2:53\)](#)
[IPConfig \(3:59\) \(Pdf\)](#)
[Ping \(4:44\) \(Pdf\)](#)
[Tracert \(3:28\) \(Pdf\)](#)
[PathPing \(4:10\) \(pdf\)](#)
[Netstat \(4:21\) \(pdf\)](#)
[TCP/IP Command Line Tools Section Quiz](#)

Introduction to Programming Using HTML and CSS

Other Resources

[The Official Requirements:](#)

HTML:

[www.w3schools HTML](#)

[Semantic elements such as <header> or <details>](#)

[<track> \(also learn <audio> and <video>](#)

[Learn in 5 minutes](#)

HTML Forms:

[FieldSets](#)

CSS

[www.w3schools CSS:](#)

This document is formatted as:

Overarching idea about HTML/CSS

1. Specific ideas relating to the overarching idea
 - a. List of individual concepts related to specific idea
 - b. Basic explanation of concepts
 - i. Specific explanation of concepts

Understanding HTML Fundamentals

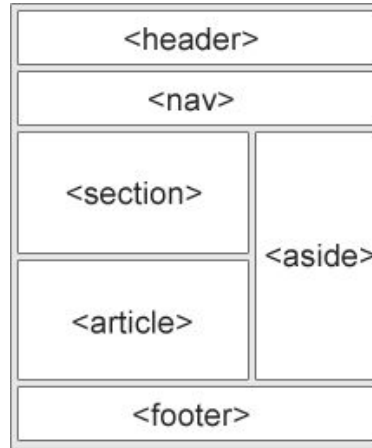
1. Construct markup that uses metadata elements
 - a. Script; noscript; style; link; meta tags, including encoding, keywords, viewport, and translate
 - b. Metadata, relations to other coding languages.
 - i. `<script>` = Javascript code, such as `alert()`; [Example](#)
 - ii. `<noscript>` = Message for when Javascript is disabled. (e.g. "Javascript is Disabled!") [Example](#)
 - iii. `<link>` = external resources. (e.g. `<link href="style.css" rel="stylesheet" type="text/css">`) [Example](#)
 - iv. `<meta>` = metadata. (e.g. `<meta charset="UTF-8">`) [Example](#)

2. Construct well-formed markup that conforms to industry best practices
 - a. DOCTYPE declaration; HTML; head; body; proper syntax, including closing tags and commonly used symbols; comments
 - b. Basic syntactic requirements for creating a document
 - i. `<!DOCTYPE html>` = html5 preamble. Always include.
 - ii. `<html>` = the tag that surrounds everything else.`</html>` [Example](#)
 - iii. `<head>` = where you put metadata & script/title tags, etc. [Example](#)
 - iv. `<body>` = where text/main page is. [Example](#)
 - v. ` `; non-breaking space
`<`; less than
`>`; greater than
`&`; ampersand
`"`; double quote
`'`; single quote
 - vi. `<!--` is a comment `-->`

Structure Documents Using HTML

1. Construct and analyze markup to structure content and organize data
 - a. Table tags; h1-h6; p; br; hr; div; span; ul; ol; li
 - b. Represent basic elements that will be included in (nearly) all documents
 - i. h1-h6; defines individual headings 1 through 6 in the document [Example](#)
 - ii. p; defines an individual paragraph in the document [Example](#)
 - iii. br; represents a literal line break [Example](#)
 - iv. hr; represents a thematic line break, usually between sections or articles [Example](#)
 - v. div; represents a generic grouping of html elements, usually use to differentiate style elements [Example](#)
 - vi. span; groups inline elements in a document, usually used to modify styling on text within paragraphs [Example](#)
 - vii. ul; represents an unordered (bulleted) list [Example](#)
 - viii. ol; represents an ordered (numbered, alphabetical, etc.) list [Example](#)
 - ix. li; represents an item in a list [Example](#)
2. Construct and analyze markup that uses HTML5 semantic elements
 - a. Semantic tags; header; nav; section; article; aside; footer; details; summary; figure; caption
 - b. Fundamental tags in defining the specifics of the content of the document
 - i. header - specifies text at the top (or “head”) of a document, often represents a title of the document. [Example](#)
 - ii. nav - specifies a block or set of navigation links. [Example](#)
 - iii. section - specifies a thematic grouping of content. [Example](#)

- iv. article - specifies independent, self-contained content. [Example](#)
 - Differentiating between article could be intended to be read intended to be read within the
- v. aside - usually specifies an element elements. [Example](#)
- vi. footer - specifies text at the bottom (or
- vii. details - specifies additional through the use of drop down text.
- viii. summary - defines the summary of the
- ix. figure - specifies self-contained caption. [Example](#)
- x. caption - defines a caption specifically



and section: an article should make sense on its own and separately from the rest of the document. Sections are context of the rest of the document.

horizontally adjacent to (or “aside” from) the main

“foot”) of a document

information that the user can show or hide, often

[Example](#)

details included in the details tag. [Example](#)

content, usually used to group an image with its

for tables. [Example](#)

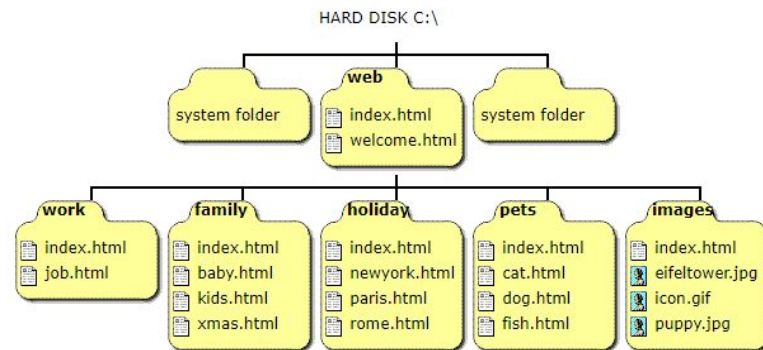
3. Construct and analyze markup that implements navigation

a. Image links; a; target; bookmark; relative versus absolute links; navigating simple folder hierarchies

b. Navigating and Structuring a HTML document.

- i. Image Links- Linking images through use of URL or File path on your computer.
- ii. <a> - <a> tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link's destination. [Example](#)
- iii. target - Used in the <a>. The target attribute specifies where to open the linked document. [Example](#)
- iv. bookmark - HTML bookmarks are used to allow readers to jump to specific parts of a Web page. After creating the id you would use the <a> command to link the bookmarks together. [Example](#)
- v. relative versus absolute links - An absolute file path is the full URL to an internet file. A relative file path points to a file relative to the current page. [Example](#)
- vi. navigating simple folder hierarchies - Make sure while linking pages that you are linking the correct file path based off of your file structure.

The diagram below represents the possible structure of how your web site could look on your PC. On your ISP's server the directory containing all your files (in this case called "web") would be your home (root) or domain directory.



In this example the folders each contain files relating to a specific subject. The images folder contains all the gif and jpg images. If you want to link to another file or image you need to give the location of the file correctly.

4. Construct and analyze markup that uses form elements
 - a. Form attributes; action; method; submission methods; accessibility; input types and restrictions; select; textarea; button; output; option; datalist; fieldset
 - b. Defines the usage of forms, often used for login functionality in web applications
 - i. action; specifies where to send form data after the form is submitted [Example](#)
 - ii. method; either "get" or "post", specifies the HTTP method to use when sending form-data [Example](#)
 1. Get is less secure, and so is generally used less
 2. Post is more secure, and is usually the go-to option
 - iii. Accessibility; ensuring that form resources are able to be requested consistently
 - iv. Input types and restrictions; validating input data to ensure that it is consistent with what the document expects to be input. [Example](#)
 - v. select; used to create a drop-down list from which the user can select one individual option (with the option tag) [Example](#)
 - vi. textarea; defines a block of text input of specified dimensions [Example](#)
 - vii. button; defines a simple button, usually with "onclick" settings [Example](#)
 - viii. output; defines the output for a function performed within the document [Example](#)
 - ix. option; used to specify an individual option in an input list (which was created with the select tag or datalist tag) [Example](#)
 - x. datalist; defines predefined options for an input tag, uses the option tag for individual options. [Example](#)
 - xi. fieldset; draws a box around included elements in order to group them together (usually for forms) [Example](#)

Present Multimedia Using HTML

1. Construct and analyze markup that displays images

- a. img and picture elements and their attributes
 - b. Adding videos, images, anything that isn't text.
 - i. img = images, as `` [Example](#)
 - ii. picture = container for image & source tags, changing the image as a certain variable changes: `<picture> <source srcset = "altpicture.png" media="(min-width: 600px)"> ...` [Example](#)
2. Describe the appropriate use of the img, svg, and canvas elements
- a. Using img,svg, and canvas elements in the html format
 - b. Implementing these commands for the correct use
 - i. `` = used for jpgs, pngs, etc. [Example](#)
 - ii. `<svg>` = img used for svg files. [Example](#)
 - iii. `<canvas>` = coupled with APIs to draw animations or graphics. [Example](#)

Construct and analyze markup that plays video and audio

- a. Video; audio; track; source; simple iframe implementations
- b. Implementing these commands to play video and audio.
 - i. `<video>` = embedded video, as `<video src="*.webm" poster="picture.jpg">Text/HTML for if video doesn't load.</video>` [Example](#)
 - ii. `<audio>` = audio file embed, as `<audio src="*.mp3"> Text/HTML for if it doesn't load</audio>` [Example](#)
 - iii. `<track>` = subtitles, as `<track src="*.vtt" kind="subtitles" srclang="language" label="Displayed Text, like 'English'.">` [Example](#)
 - iv. `<source>` = multiple resources, for browser support, as `<source src="*.extension" type="mediatype/extension">` or `<source src="video.mov" type="video/quicktime">` [Example](#)
 - v. `<iframe>` = embedded html page, as `<iframe width="" height="" src="https://"> </iframe>` [Example](#)

Style Web Pages Using CSS

- 1. Construct and analyze styles that position content
 - a. Positioning, including float, relative, absolute, max-width, overflow, height, width, and align; inline versus block; visibility; box model, including margins and padding
 - b. Understanding CSS block-style positioning of elements.
 - i. Position: relative/absolute; places the element according to the normal flow of the page/according to the window. Uses left/right/top/bottom: px; [Example](#)
 - ii. Display: block/inline; Determines if the element appears inline or as a page-wide element. [Example](#)
 - iii. Visibility: visible/hidden; shows/hides the element, but still affects page layout as if it was visible. [Example](#)
 - iv. padding: px; distance between content (like text) and inner part of the border. [Example](#)
 - v. border-width: px; thickness of the border. [Example](#)
 - vi. Margin: px; distance between border and other elements. [Example](#)

2. Construct and analyze styles that format text

- a. Font-family; color; font-style; font-size; font-weight; link colors; text formatting, including text alignment, text decoration, and indentation
- b. Using CSS to determine fonts, colors, and positioning.
 - i. Font-family- font family of a text is set with the font-family property. The font-family property should hold several font names as a "fallback" system. If the browser does not support the first font, it tries the next font, and so on. [Example](#)
 - ii. color - Helps set the colors of backgrounds, fonts, borders, etc. Uses RGB, Hex, RGBA, etc. [Example](#)
 - iii. font-style - Optional. Defines how the font should be styled. Default is "normal" [Example](#)
 - iv. font-size - font-size property sets the size of the text. [Example](#)
 - v. font-weight - font-weight property specifies the weight of a font (i.e. thickness) [Example](#)
 - vi. link colors - background-color property can be used to specify a background color for links: [Example](#)
 - vii. text formatting - Formatting the text such as positioning and other added elements. [Example](#)
 - viii. text alignment - text-align property is used to set the horizontal alignment of a text. A text can be left or right aligned, centered, or justified [Example](#)
 - ix. text decoration - text-decoration property is used to set or remove decorations from text. The value text-decoration: none; is often used to remove underlines from links [Example](#)
 - x. Indentation - text-indent property is used to specify the indentation of the first line of a text [Example](#)

3. Construct and analyze styles that format backgrounds and borders

- a. Border-color; border-style; border-width; backgrounds; divs; colors
- b. Understanding how to change colors/styles of borders and backgrounds.
 - c. border-color - Sets the color of the border [Example](#)
 - d. border-style - border-style property specifies what kind of border to display. [Example](#)
 - e. border-width - border-width property specifies the width of the four borders. The width can be set as a specific size (in px, pt, cm, em, etc) or by using one of the three pre-defined values: thin, medium, or thick. [Example](#)
 - f. backgrounds - background-color property specifies the background color of an element
[Example Single Background](#)
[Example Multiple Background](#)

4. Analyze styles that implement a simple responsive layout

- a. Units of measure; responsive effects with CSS, including viewport and media query; percentages versus pixels; frameworks and templates; max width
- b. Ways of measuring and implementing settings for the CSS layout.
 - i. Units of measure - Includes units of measure; responsive effects with CSS, including viewport and media query; percentages versus pixels; frameworks and templates; max width. em - Relative to the font-size of the element, ex - Relative to the x-height of the current font, ch - Relative to width of the "0" (zero), rem - Relative to font-size of the root element, vw -

- Relative to 1% of the width of the viewport, vh - Relative to 1% of the height of the viewport, vmin - Relative to 1% of viewport's smaller dimension, vmax - Relative to 1% of viewport's larger dimension, % - Percentage [Example](#)
- ii. responsive effects with CSS - Responsive web design makes your web page look good on all devices. Responsive web design uses only HTML and CSS. Responsive web design is not a program or a JavaScript. [Example](#)
 - iii. including viewport and media query - The viewport is the user's visible area of a web page. Designers take control over the viewport, through the <meta> tag. [Example](#)
 - iv. percentages versus pixels - pixels use a absolute length of measurement while percentages are relative and scalable depending on the device. [Example](#)
 - v. frameworks and templates - Existing code or template that you can adjust for your needs. [Example](#)
 - vi. max width - max-width will improve the browser's handling of small windows. This is important when making a site usable on small devices. [Example](#)

[HTML5 Application Development Fundamentals](#)

[Manage the application life cycle \(20-25%\)](#)

- Understand the platform fundamentals
 - Packaging and the runtime environment: app package, app container, credentials/permission sets, host process, leverage existing HTML5 skills and content for slate/tablet applications
- Manage the state of an application
 - Manage session state, app state, and persist state information; understand states of an application; understand the differences between local and session storage
- Debug and test an HTML5-based, touch-enabled application
 - Touch gestures; understand which gestures you test on a device

Preparation resources

- [Learn HTML5 in five minutes!](#)
- [A history \(API\) lesson](#)

[Build the user interface \(UI\) by using HTML5 \(25-30%\)](#)

- Choose and configure HTML5 tags to display text content
- Choose and configure HTML5 tags to display graphics

- When, why, and how to use Canvas; when, why, and how to use scalable vector graphics (SVG)
- Choose and configure HTML5 tags to play media
 - Video and audio tags
- Choose and configure HTML5 tags to organize content and forms
 - Tables, lists, sections; semantic HTML
- Choose and configure HTML5 tags for input and validation

Preparation resources

- [Creating user interface elements with Canvas or SVG](#)
- [Designing accessibility with HTML5](#)
- [Video element | video object](#)

Format the user interface by using Cascading Style Sheets (CSS) (20-25%)

- Understand the core CSS concepts
 - Separate presentation from content (create content with HTML and style content with CSS); manage content flow (inline versus block flow); manage positioning of individual elements(float versus absolute positioning); manage content overflow (scrolling, visible, and hidden); basic CSS styling
- Arrange UI content by using CSS
 - Use flexible box and grid layouts to establish content alignment, direction, and orientation; proportional scaling and use of "free scale" for elements within a flexible box or grid; order and arrange content; concepts for using flex box for simple layouts and grid for complex layouts; grid content properties for rows and columns; use application templates
- Manage the flow of text content by using CSS
 - Regions and using regions to flow text content between multiple sections (content source, content container, dynamic flow, flow-into, flow-from, msRegionUpdate, msRegionOverflow, msGetRegionContent); columns and hyphenation and using these CSS settings to optimize the readability of text; use "positioned floats" to create text flow around a floating object
- Manage the graphical interface by using CSS
 - Graphics effects (rounded corners, shadows, transparency, background gradients, typography, and Web Open Font Format); two-dimensional (2-D) and three-dimensional (3-D) transformations (translate, scale, rotate, skew, and transitions and animations); SVG filter effects; Canvas 3-D perspective

Preparation resources

- [CSS](#)
- [HTML/CSS for Windows Store apps](#)
- [CSS3 Effects, transitions, and animations](#)

Code by using JavaScript (30-35%)

- Manage and maintain JavaScript
 - Create and use functions; jQuery and other third-party libraries
- Update the UI by using JavaScript

- Locate/access elements; listen and respond to events; show and hide elements; update the content of elements; add elements
- Code animations by using JavaScript
 - Use animation; manipulate the canvas; work with images, shapes, and other graphics
- Access data access by using JavaScript
 - Send and receive data; transmit complex objects and parsing; load and save files; App Cache; datatypes; forms; cookies; localStorage
- Respond to the touch interface
 - Gestures, how to capture and respond to gestures
- Code additional HTML5 APIs
 - GeoLocation, Web Workers, WebSocket; File API
- Access device and operating system resources
 - In- memory resources, such as contact lists and calendar; hardware capabilities, such as GPS, accelerometer, and camera

Preparation resources

- [HTML 5 JavaScript](#)
- [Using requestAnimationFrame for power efficient animations sample \(Windows 8.1\)](#)
- [Application Cache API \("AppCache"\)](#)

Introduction to Programming Using Block-Based Languages (Touch Develop)

For this MTA, I would recommend completing a full Code.org programming lesson.

Apply Strategies to Solve Computational Problems (15-20%)

- Identify basic algorithmic steps to solve simple problems
 - Decompose simple problems into steps, sequence processes in the appropriate order, describe storyboards, resolve challenges and errors related to logic or pseudocode

- Decompose a computational problem into sub-problems
 - Describe computer programs that use logical subdivisions, describe solutions that use programmable strategies such as objects, functions, and parameters in the pseudo code provided, identify situations when code can be reviewed

Design Algorithms (25-30%)

- Create Algorithms
 - Differentiate problems as easy or hard for computers to solve, apply the concept of iteration, create simple algorithms
 - [Big O notation](#) is a common method to determine problem complexity.
- Analyze game play to identify the algorithmic sequences
 - Analyze a game and create a sequence of instructions for playing it, identify an event, create the code for an event in block-based editors, explain the “on every frame” code and event handlers
 - Try out the [code.org event loop](#)
- Create and analyze algorithms used to implement animation and movement in code
 - Describe animation that uses a series of individual frames, resolve errors in algorithms, create algorithms that can be translated into pseudocode or block-based code, use code to command items on the screen or device
- Explain sequence, selection, and iteration
 - Define loops, identify the control variable, predict the output of loop, random number, and control variable constructs, identify conditional statements; choose the appropriate Boolean logic for specific results
 - Try code.org random function (maybe make a dice rolling program)

Work with Data Representation in Block-Based Programming Languages (15-20%)

- Represent data in text, sounds, pictures, and numbers
 - Create code to add and position objects, such as sprites on a screen or device, identify data examples as text, sound, pictures, or numbers, change the parameters of “set frame grid” to work with different sprite sheets, explain the role of cloud variables, explain the impact of variable scope, including cloud

variables, global variables, and local or temporary variables, design, create, and populate a table or two-dimensional array, describe multiple uses for data

- Employ simple data structures to solve computational problems
 - Declare and use variables in a program, use input and variables to calculate new information, describe arrays, lists, and collections, explain the differences between variables and arrays
- Describe how data is accessed in apps and games
 - Distinguish among variable data types, add, remove, and insert elements in a collection, iterate through collections using for loops and for-each loops, describe how random numbers are used
 - Try code.org and javascript array usages

Solve Computational Problems by Using Modeling and Simulation (5-10%)

- Solve computational problems by using computer and non-computer methods
 - Use algorithms and Boolean logic, use games and apps to simulate practical tasks such as converting currencies
- Represent events observed in the physical world by coding simulation and modeling programs
 - Create programs and apps that mimic random occurrences, create programs and apps that demonstrate fundamentals of physics such as gravity, acceleration, and bounce

Code Programs in Block-Based Programming (15-20%)

- Implement solutions using code
 - Identify the basic coding elements of the programming environment, create code for conditional statements, create conditional statements using AND, OR, and NOT, create loops, use counting variables
 - Learn boolean logic, and if statements
- Use libraries and built-in functions to facilitate programming solutions
 - Code by using ready-made functions related to objects including the game board, wall, obstacle, sprite, string, textbox, collection, and turtle
- Make connections between elements of mathematics and computer science

- Plot points on a gaming coordinate system, analyze a game to discover how random numbers are used, use the random range function to generate random numbers, evaluate random numbers by using conditionals

Assess Personal Security in Internet Communications (5-10%)

- Explain the basic components of Internet communication
 - Explain how information travels across the Internet, define cloud computing, describe the Internet of Things, describe the roles of cloud computing and cloud storage and their uses
 - The Internet of Things is a collection of simple devices connected to the internet, such as fridges and sinks. It has industrial uses such as valves, drones and robots.
- Explain the principles of security
 - Describe how personal information can be used in inappropriate ways, describe how to prevent someone from gaining access to an online account, describe steps that websites take to keep passwords secure, describe the Caesar Cipher
- Implement encryption and authentication strategies
 - Encode and decode messages using Unicode, evaluate passwords based on security criteria, describe how hashed passwords enhance Internet security
 - Hashed passwords are more secure because the server doesn't have to store your password.

Examine the Software Development Process (5-10%)

- Plan and create programs
 - Analyze problems in relation to your audience and identify which apps or games can be part of the solution and how they can be used, describe user-experience principles, gather user input, use code or text to create instructions for using a program
- Describe software development processes used to solve problems
 - Plan project tasks and delegate responsibilities, describe a cycle of create, evaluate, and revise
- Analyze and evaluate completed programs
 - Evaluate for readability and usability, give and receive feedback, evaluate feedback and revise the program accordingly

Software Development Fundamentals

I would **highly** recommend you know what big-o notation is before taking the test. Many of my notes are in big-o-notation.






Understanding core programming (15-20%)

- Understand computer storage and data types
 - 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
 - [Crash course on memory](#)
 - [Technical view of memory](#)
- Understand computer decision structures
 - 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
 - See python guide to learn this
- Identify the appropriate method for handling repetition
 - [For loops, While loops, Do...While loops, and recursion](#)
 - If you know how and when to write all of these in one language, you should be good.
- Understand error handling
 - Structured exception handling
 - See python or javascript try, catch/except,

Preparation resources

- [Rediscover the lost art of memory optimization in your managed code](#)

Flow Chart Basics

Name	Symbol	Use in Flowchart
Oval or Ellipse		the beginning or end of a process
Flow Line		the direction of flow in the process
Parallelogram		either an input operation (e.g., doing research) or an output operation (e.g., printing a rough draft out)
Rectangle		a process is to be carried out (e.g., creating the topic sentence of a paper)
Diamond		a decision must be made, which can result in two or more directions the process can take (e.g., deciding whether to revise a paper again or turn it in)

- [Control flow in Visual Basic](#)
- [Exceptions and exception handling \(C# programming guide\)](#)

Understanding object-oriented programming (20-25%)

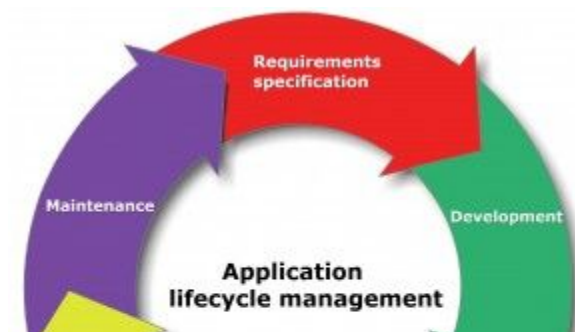
- Understand the fundamentals of classes
 - Properties, methods, events, and constructors; how to create a class; how to use classes in code
 - See python guide for more; it's fundamentally the same in any language
- Understand inheritance
 - Inheriting the functionality of a base class into a derived class
 - [Basic inheritance](#) is where a Dog is-a animal, and has the "is-a" relation.
- Understand polymorphism
 - Extending the functionality in a class after inheriting from a base class, overriding methods in the derived class
 - [Polymorphism](#) is where one object can take many shapes.
- Understand encapsulation
 - Creating classes that hide their implementation details while still allowing access to the required functionality through the interface, access modifiers
 - This is referring to things such as getter and setter functions, [private variables](#).

Preparation resources

- [Learn the principles of object-oriented programming](#)
- [Object-oriented programming](#)
- [Programming in C#: \(01\) OOP, managed languages and C#](#)

Understanding general software development (15-20%)

- Understand application lifecycle management
 - Phases of application lifecycle management, software testing



- The life of an application commonly cycles between many phases, such as development, testing, deployment, maintenance, etc.
- Interpret application specifications
 - Reading application specifications and translating them into prototypes, code, select appropriate application type, and components
 - Based on requirements, write code that fulfills them, similar to the Computer Science A exam
- Understand algorithms and data structures
 - Arrays, stacks, queues, linked lists, and sorting algorithms; performance implications of various data structures; choosing the right data structure
 - Arrays are a data structure where every location can be accessed in $O(1)$ time.
 - [Stacks commonly include LIFO and FIFO](#) where FIFO is like a line and LIFO is like a stack of plates.
 - [Linked lists](#) are a data structure where elements can be found in $O(n)$ time and each elements links to its next neighbor.
 - [Sorting algorithms](#) include [insertion sort](#), [selection sort](#), [bubble sort](#), etc.

Preparation resources

- [Application Lifecycle Management with Visual Studio and Team Foundation Server](#)
- [Agile software development with Microsoft Visual Studio ALM](#)
- [An extensive examination of data structures using C# 2.0](#)

[Understanding web applications \(15-20%\)](#)

- Understand web page development
 - HTML, Cascading Style Sheets (CSS), JavaScript
 - See respective MTA guides. Generally speaking HTML is the flow and structure of a website, CSS is the visuals, and JavaScript is the functions.
- Understand Microsoft ASP.NET web application development
 - Page life cycle, event model, state management, client-side versus server-side programming

- Client-side programming runs on the client, server side programming needs security and runs on the server
- Understand web hosting
 - Creating virtual directories and websites, deploying web applications, understanding the role of Internet Information Services
 - [IIS is like](#) the windows version of Apache, but also an SFTP server
 - IIS has websites, which each hold many applications.
- Understand web services
 - Web services that will be consumed by client applications, accessing web services from a client application, SOAP and Web Service Definition Language (WSDL)
 - [WSDL can define a web service](#) based on its possible inputs and expected outputs.
 - [SOAP can be used for sending data](#) over HTTP and has to be encoded in XML

Preparation resources

- [Modern web development](#)
- [Developing web applications with ASP.NET](#)
- [Accessing raw SOAP messages in ASP.NET Web Services](#)

[Understanding desktop applications \(15-20%\)](#)

- Understand Windows apps
 - UI design guideline categories, characteristics and capabilities of Store Apps, identify gestures
 - [UI guidelines](#)
- Understand console-based applications
 - Characteristics and capabilities of console-based applications
 - Console-based applications have no GUI, but rather textual terminal input and output
- Understand Windows Services
 - Characteristics and capabilities of Windows Services

- [Intro to windows services](#)

Preparation resources

- [Windows Forms application basics \(Visual Basic\)](#)
- [Building console applications](#)
- [Introduction to Windows service applications](#)

Understanding databases (15-20%)

- Understand relational database management systems
 - Characteristics and capabilities of database products, database design, Entity Relationship Diagrams (ERDs), normalization concepts
 - [ERDs](#) show the relation between things, and can be used in workflows, databases, and more.
- Understand database query methods
 - Structured query language (SQL), creating and accessing stored procedures, updating data and selecting data
 - [SQL is a language](#) for selecting and editing databases through tables
- Understand database connection methods
 - Connecting to various types of data stores, such as flat file; XML file; in-memory object; resource optimization
 - [A “flat file”](#) would be commonly a CSV, where rows of the file are rows of the table
 - [An XML file](#) is the format HTML uses and can be good for document-style data
 - [Resource optimization](#) would be referring to sorting, caching, etc.

Preparation resources

- [Database design methodologies for Microsoft SQL Server \(Visual Basic\)](#)
- [Querying the database](#)

- [Connecting to data in Visual Studio overview](#)

Introduction to Programming Using Python

After studying, try programming something simple to see what you don't know, like a terminal todo list that can save or load into files, or a terminal calendar with events.

Perform Operations using Data Types and Operators (20-25%)

- Evaluate an expression to identify the data type Python will assign to each variable
 - Identify str, int, float, and bool data types
 - [W3 schools](#)
 - “str” refers to text, and is stored as an array of characters. Python strings are immutable, meaning they cannot be changed in-place but rather are remade each time you change it.
 - “int” refers to an integer, such as 5 or -16, but NOT 5.1
 - “float” refers to a float, which can be (within wide bounds) any number, such as 15, 1.515, 3.141, 10000
 - “bool” refers to a boolean, which is a type of value that can only be true or false, such as a light switch.
 - [making variables](#)
 - Same as any other language, but dynamically typed so you don't need to specify what objects are as you would in Java or C++.
 - Perform data and data type operations
 - Convert from one data type to another type; construct data structures; perform indexing and [slicing](#) operations
 - [Arrays w3](#)
 - [Python len\(\)](#) gives you the length of an array or other data structure, such as a string

- Slicing in python is where you have an array and you can access elements between x and y by using array[x:y]. As usual, it's inclusive, exclusive. To get every element, use array[:] or array[0:array.length]
 - [Converting](#) is as easy as the type as a function. Ex "5" == str(5)
 - [Indexing is referring](#) to values in an array. To get the first element of the "dog" array, use dog[0]. To get the last element use dog[-1] (negative values start from the end).
- Determine the sequence of execution based on operator precedence
 - Assignment; Comparison; Logical; Arithmetic; [Identity \(is\)](#); Containment (in)
 - Make sure you know [the difference between "x == y" and "x is y"](#). The difference is that "x==y" means they have the same value, but "x is y" refers to x pointing/referring to the same value as y.
 - [In refers to something within an array](#). If I have the array "dog = ["bob","joe","jim"]", then "'bob' in dog" would be true.
- Select the appropriate operator to achieve the intended result
 - Assignment; Comparison; Logical; Arithmetic; Identity (is); Containment (in)
 - Same as previous one; see above for usage

[Control Flow with Decisions and Loops \(25-30%\)](#)

- Construct and analyze code segments that use branching statements
 - if; elif; else; nested and compound conditional expressions
 - [w3 python if else](#)
 - The same as Javascript, but remember [not to use {brackets}](#)
 - There are no "switch" statements.
- Construct and analyze code segments that perform iteration
 - while; for; break; continue; pass; nested loops and loops that include compound conditional expressions
 - [w3schools](#) - These are the same as Javascript
 - [Pass is a code segment](#) that does nothing. It exists because if you have an empty except block or

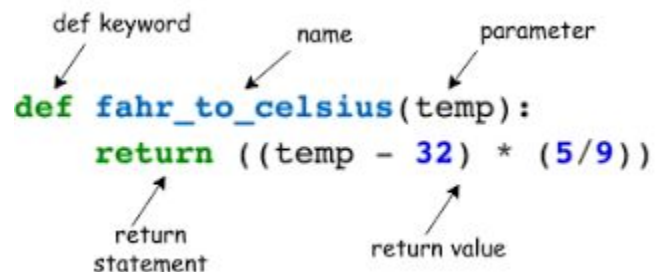
[Perform Input and Output Operations \(20-25%\)](#)

- Construct and analyze code segments that perform file input and output operations

- Open; close; read; write; append; check existence; delete; with statement
 - [w3 schools](#)
 - `f = open("file.txt","w")` # could also be `r` for read or `r+` for read write
 - `f.read()` for entire file
 - `f.write("text")`
 - `f.readLine()`
- Construct and analyze code segments that perform console input and output operations
 - Read input from console; print formatted text; use of command line arguments
 - This is [referring to sys.argv](#), which is an array of arguments (like Java or C++)

Document and Structure Code (15-20%)

- Document code segments using comments and documentation strings
 - Use indentation, white space, comments, and documentation strings; generate documentation by using [pydoc](#)
 - When you use `help(function)`, that is provided through `pydoc`.
 - Write it [like so](#)
- Construct and analyze code segments that include function definitions
 - Call signatures; default values; return; def; pass
 - A function signature is all of the variables it asks for as input
 - Default values are where you have a function which could run without input, such as how `"def ring(pitch=5)"` could be ran as `"ring(15)"` for a pitch of 15 or as `"ring()"` for a default input of pitch 5.
 - `"def"` is how you define a function, and make sure you know the required syntax



- pass is a "function" that does nothing, useful in empty "except" blocks or unimplemented functions.

Perform Troubleshooting and Error Handling (5-10%)

- Analyze, detect, and fix code segments that have errors
 - Syntax errors; logic errors; runtime errors
 - Identify the error using the compiler/output, and it could tell you the line number
- Analyze and construct code segments that handle exceptions
 - Try; except; else; finally; raise
 - Make sure you know why else was listed here. In python, if a loop breaks because of the condition, or a try doesn't throw anything, it calls the else after it.

Perform Operations Using Modules and Tools (1-5%)

- Perform basic operations using built-in modules
 - Math; datetime; io; sys; os; os.path; random
 - Import datetime
 - Import time
 - time.time() gets you the current epoch time
 - Make sure you know what epoch time is
 - datetime.date(year,month,day)
- Solve complex computing problems by using built-in modules
 - Math; datetime; random
 - See above

while - else clause

The optional **else** clause runs only if the loop exits normally (not by break)

```
>>> x = 0
>>> while x <= 6 :
    x = x + 1
    print x
else :
    y = x
>>> print y
```

- while condition :
 Statements
- else :
 Statements

Database Fundamentals

Create database objects (20-25%)

- Choose data types
 - Understand what data types are, why they are important, and how they affect storage requirements
 - [Every TSQL data type](#)
- Understand tables and how to create them
 - Purpose of tables; create tables in a database by using proper ANSI SQL syntax
 - [W3schools CREATE TABLE](#)
- Create views
 - Understand when to use views and how to create a view by using T-SQL or a graphical designer
 - [Views are a virtual table](#) made by a query
 - Views can be a subset of a table for purposes of security
 - Views can be used to conveniently group traits in one virtual table.
- Create stored procedures and functions
 - Select, insert, update, or delete data
 - [W3 schools](#)

Preparation resources

- [Publishing data and database objects](#)
- [CREATE TABLE \(Transact-SQL\)](#)
- [How to: Create views \(Visual Database Tools\)](#)

[Manipulate data \(25-30%\)](#)

- Select data
 - Utilize SELECT queries to extract data from one table, extract data by using joins, combine result sets by using UNION and INTERSECT
 - SQL [union](#) by default only unions distinct (different) values, so you can use SQL "UNION ALL" to
- Insert data
 - Understand how data is inserted into a database, how to use INSERT statements

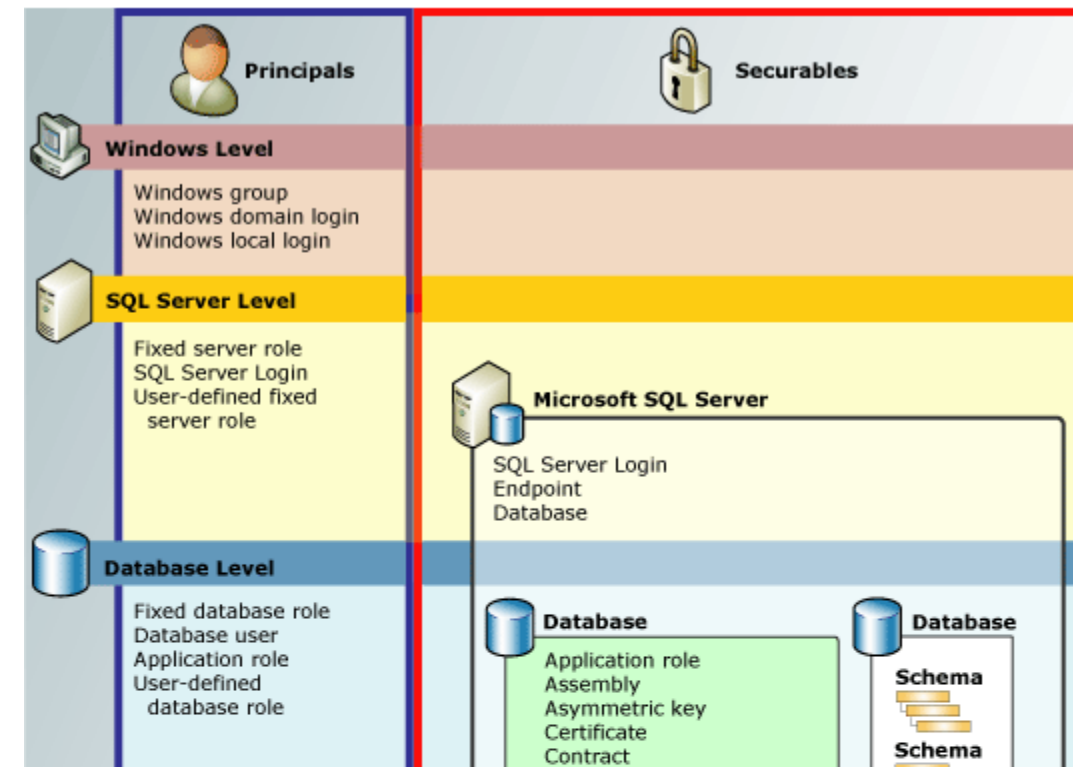
- [INSERT INTO users \(name,job\) VALUES \(\("Bob","Worker"\),\("Jill","Manager"\)\)](#)
- Update data
 - Understand how data is updated in a database and how to write the updated data to the database by using the appropriate UPDATE statements, update by using a table
 - [UPDATE users SET name="Bob" WHERE name="Robert"](#)
- Delete data
 - Delete data from single or multiple tables, ensure data and referential integrity by using transactions
 - [SQL Transactions and ACID](#)
 - Atomicity - ensures that all operations within the work unit are completed successfully. Otherwise, the transaction is aborted at the point of failure and all the previous operations are rolled back to their former state.
 - Consistency - ensures that the database properly changes states upon a successfully committed transaction.
 - Isolation - enables transactions to operate independently of and transparent to each other.
 - Durability - ensures that the result or effect of a committed transaction persists in case of a system failure.
 - commit, rollback, savepoint, set transaction
 - "rollback to <savepoint>" rolls back as if you never ran the commands and made the edits since the savepoint, as if saving in a game.
 - "release <savepoint>" deletes it

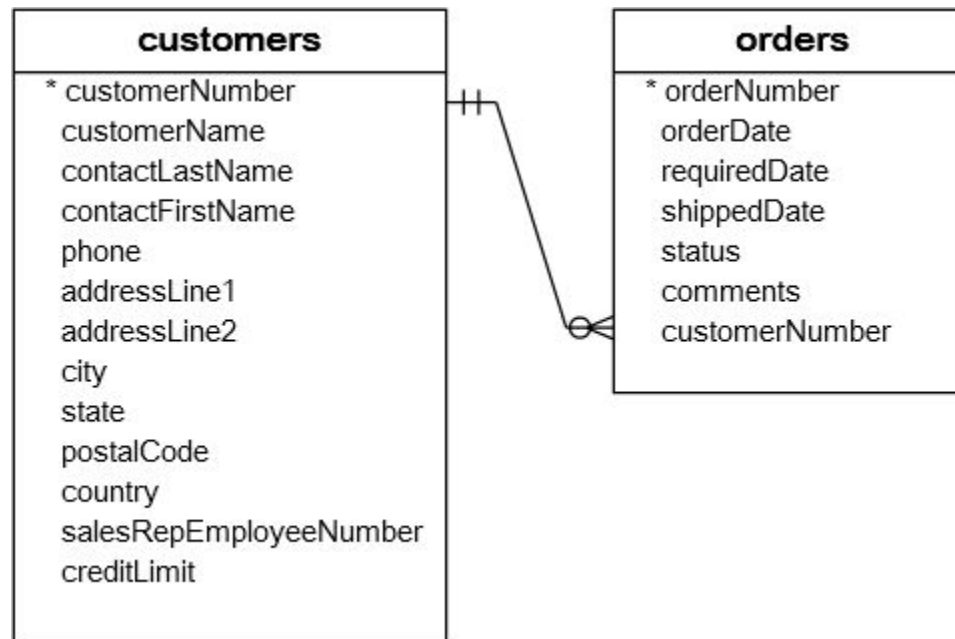
Preparation resources

- [Selecting data using the SqlDataSource control](#)
- [Walkthrough: Retrieving, updating, inserting, and deleting data with the LinqDataSource and DetailsView controls](#)
- [How to: Delete rows from the database \(LINQ to SQL\)](#)

[Understand data storage \(15-20%\)](#)

- Understand normalization
 - Understand the reasons for normalization, the five most common levels of normalization, how to normalize a database to third normal form
 - I would strongly recommend you reading [this](#), but the summary is:
 - UNF is UnNormalized Form and can be any database.
 - 1NF is Normalized Form 1 and has atomic values, meaning a column in a row would not have multiple things.
 - 2NF means that all values in the row corresponds to the entire key
 - 3NF means no transitive dependencies, such as not having, inside of a book table, values genreID and genreName. Instead of this, have a genre table, linking genre to genreName.
 - For something to be (n)NF it has to be (n-1)NF, so 3NF has to also be 2NF, which has to be 1NF
- Understand primary, foreign, and composite keys
 - Understand the reason for keys in a database, choose appropriate primary keys, select appropriate data type for keys, select appropriate fields for composite keys, understand the relationship between foreign and primary keys
 - A primary key is a single, unique, key that links all the data in the row.
 - [Composite keys](#) are two keys, that when combined, guarantee the row to be unique. An example is having the keys "room number" and "seat number". One of these alone wouldn't tell you your seat (also wouldn't differentiate your seat from others), but both combined would.
 - A foreign key is a key to another separate table, such as referring to a specific customer in an order table.





- Understand indexes
 - Understand clustered and non-clustered indexes and their purpose in a database
 - [Clustered](#) indexes are what the table is sorted by (like sorting by the "student-id" key for a table of students). Non-clustered tables are not sorted.

Preparation resources

- [Working with keys](#)
- [How to: Handle composite keys in queries \(LINQ to SQL\)](#)
- [Working with indexes](#)

[Administer a database \(10-15%\)](#)

- Understand database security concepts

- Understand the need to secure a database, what objects can be secured, what objects should be secured, user accounts, and roles
 - Have physical security, firewalls, etc.
 - Microsoft calls objects that can be secured "[Securables](#)" and these are the valuable objects in your table
- Understand database backups and restore
 - Understand various backup types, such as full and incremental, importance of backups, how to restore a database
 - [Incremental vs differential](#)

Preparation resources

- [Security considerations for a SQL Server installation](#)
- [SQL Server security \(ADO.NET\)](#)
- [Backing up and restoring databases in SQL Server](#)

Windows Server Administration Fundamentals

This test requires you to know the names of everything, too, like everything in active directory.

[Understanding server installation \(10-15%\)](#)

- Understand device drivers
 - Installation, removal, disabling, update/upgrade, rollback, troubleshooting, Plug & Play, IRQ, interrupts, driver signing, managing through Group Policy
 - [Drivers](#) are in charge of telling the OS how to use hardware such as keyboards, controllers, video cards, and printers.
 - An [Interrupt ReQuest](#) is sent to the CPU to interrupt whatever its doing, for time sensitive things such as USB keystrokes
- Understand services

- Which statuses a service can be in, startup types, recovery options, delayed startup, Run As settings for a service, stopping or pausing a service, service accounts, dependencies
 - Services on "automatic" start immediately when the computer starts up, while "automatic (delayed)" starts later in the startup process.
- Understand server installation options
 - Choose the correct operating system version options; Server core vs. Desktop Experience, Nano Server installation, interactive installs; automated install using WDS; VHD/VHDX installation source, perform unattended installs; perform upgrades, clean installs, and migrations
 - [Server Core](#) has no GUI, while desktop experience does. [Interactive installs](#) require you to answer many questions. Unattended installs are where you prematurely answer questions into an answer file, so you don't have to watch the installation. They are good for if you are installing to many computers. A VHD is a virtual hard disk, which can be used for installing OSes into HyperV.
 - [Nano Server](#) is a windows server meant to be used only remotely (headless). It has no local logon capability, only supports 64 bit applications. It uses less disk space, installs quicker, and needs less updates.

Preparation resources

- [Installing devices and their drivers](#)
- [Server Core installation option getting started guide](#)
- [Installing Windows Server 2008 R2](#)

[Understanding server roles \(25-30%\)](#)

- Identify application servers
 - Mail servers, database servers, collaboration servers, monitoring servers, threat management
 - A mail server can be for public or private mail
- Understand Web services

- IIS, WWW, and FTP, installing from Server Manager, separate worker processes, adding components, sites, ports, SSL, certificates
 - Test each on the server in the back room (ask Pauley)
- Understand remote access
 - Remote assistance, remote administration tools, Remote Desktop Services, multipoint services, licensing, RD Gateway, VPN, application virtualization, multiple ports
 - Windows [MultiPoint servers](#) are when multiple windows OSes are running on one computer. It was replaced from an independent SKU into a role in windows server 2016.
 - [Remote Desktop Gateway](#) is useful for when you have multiple RD devices inside of one network.
 - Using multiple ports for RDP is an alternative to RDG where you just allocate a port number for each device, but the port numbers can be harder to track/remember.
- Understand the file and print services
 - Local printers, network printers, printer pools, web printing, web management, driver deployment, file, folder, and share permissions vs. rights, auditing, print job management
 - Printer pooling is when you have multiple printers to share the workload in a round-robin style, and uses multiple printers of the same brand and type.
 - [Auditing](#) in windows is to investigate or diagnose something through logged events
- Understand server virtualization
 - Virtual memory, virtual networks, snapshots and saved states, physical to virtual conversions, virtual to physical conversions, VHD and VHDX formats, nested virtualization
 - [VHDX](#) is an updated version of VHD, with a max storage of 64TB instead of 2TB, and is more modern.
 - VHDX is resilient to power failure events. It is not supported by windows 8.
 - [Nested Virtualization](#) is a VM inside of a VM, and is supported by HyperV (only for HyperV virtual machines inside of HyperV virtual machines)

Preparation resources

- [Changes in functionality from Windows Server 2003 with SP1 to Windows Server 2008](#)

- [What's new in routing and remote access in Windows Server 2008](#)

[Understanding Active Directory \(20-25%\)](#)

This is easily the hardest part of the thing, Windows has their own name for everything... [Here's a guide](#)

- Understand accounts and groups
 - Domain accounts, local accounts, user profiles, computer accounts, group types, default groups, group scopes, group nesting, understand AGDLP and AGUDLP processes to help implement nesting
 - [AGUDLP](#) is for making Role-Based access controls in windows.
 - Local accounts are stored only on one device, while domain accounts are controlled by a domain server. [Domain accounts](#) are great for places like businesses or schools where there are many computers you could be authorized to sign into.
 - There are many default groups, such as Backup Operators (privileged to make backups), Guests (without access to much), and Administrators (high access)
- Understand organizational units and containers
 - Purpose of OUs, purpose of containers, delegation, default containers, uses for different container objects, default hidden and visible containers
 - Organizational units are like folders in Active Directory, that can help to sort and categorize files.
- Understand Active Directory infrastructure
 - Domain controllers. forests, child domains, operation master roles, domain vs. workgroup, trust relationships, functional levels, deprecated functional levels, namespace, sites, replication, schema, Passport
 - The [AD Schema](#) is the "blueprint" for how the Active Directory is sorted.
 - [Windows Passport is](#) basically a password manager for windows websites.
 - [Youtube video](#) on forests, domains, child domains, etc
- Understand group policy
 - Group policy processing, Group Policy Management Console, computer policies, user policies, local policies

- Group policy is an advanced form of windows settings

Preparation resources

- [Active Directory Services](#)
- [Designing OU structures that work](#)

[Understanding storage \(10-15%\)](#)

- Identify storage technologies and their typical usage scenarios
 - Advantages and disadvantages of different storage topologies, local storage, network storage, Fibre Channel, iSCSI hardware
 - See A+ info
 - Network Storage is where files are stored off your computer, such as on a central file server. This can be much easier for backups (because you only have to backup the central server)
 - Fibre Channel is a type of SAN (Storage Area Network) using fiber optic cables. It needs dedicated network hardware.
 - Local Storage is where the files your accessing are on/in your computer. This is the most common setup and usually faster.
 - [iSCSI](#) is a type of SAN that doesn't need dedicated network hardware, and can run on existing networks, such as ethernet.
- Understand RAID redundancy
 - [RAID 0, RAID 1, RAID 5, RAID 10](#) and combinations, hardware and software RAID
 - RAID 0 is striping, where the data is faster to read.
 - RAID 1 is mirroring, where the data is safely stored on more disks, so if (n-1) out of the (n) disks fail, you can recover all the data. (at least one not-failed disk)
 - RAID 5 is striping, where the data is stored with parity and redundancy, so it can read faster, and can recover even if one disk fails.

- RAID10 is both raid 0 and 1. You have a RAID 0 set up for fast reading, and then duplicate it using RAID 1 so it is redundant.
 - Hardware RAID is where your RAID is managed by a hardware device, and can be faster for nearly every RAID type. Software RAID is usually managed by the OS (but can be in the BIOS or UEFI. If so, use that).
- Understand disk types
 - Solid State Drive (SSD) and Hard Disk Drive (HDD) types and comparisons, ATA basic disk, dynamic disk, mount points, file systems, mounting a virtual hard disk, distributed file systems
 - [SSD and HDD](#) are forms of persistent storage, but SSD is faster.
 - Mount points are where the physical disk is logically assigned, such as C:\ or D:\. It can also be in other drives, such as C:\backups
 - VHD files can be mounted in disk management (win+x+k). You can create VHDs and attach VHDs in this.

Preparation resources

- [Planning for disks and storage](#)
- [Windows Server 2008-file and storage solutions](#)
- [Overview of disk management](#)

[Understanding server performance management \(10-15%\)](#)

- Identify major server hardware components
 - Memory, disk, processor, network, 32-bit and 64-bit architecture, removable drives, graphic cards, cooling, power usage, ports
 - [Basic info](#)
 - Refer to A+ requirements
- Understand performance monitoring
 - Methodology, procedures, effect of network, CPU, memory and disk, creating a baseline, Performance Monitor, Resource Monitor, Task Manager, performance counters, Data Collector Sets
 - Refer to A+ requirements

- Understand logs and alerts, Event Viewer
 - Purpose of performance logs and alerts
 - To tell the server owner in case of an attack (ex. DDOS would show high IO), hardware flaw (ex. failed disks), or too highly utilized hardware (ex. CPU always at 100%)

Preparation resources

- [Performance monitoring getting started guide](#)
- [Computer hardware and Windows Server 2008](#)

[Understanding server maintenance \(15-20%\)](#)

- Identify steps in the startup process
 - BIOS, UEFI, TPM, bootsector, bootloader, MBR, boot.ini, POST, Safe Mode
 - [TPM](#) only uses safe hardware and boots safe windows versions to try to prevent rootkits
 - [The first thing that runs on your computer when it starts up is BIOS](#) and the newer version is UEFI
 - The boot sector is the part of a disk read to find the bootloader/OS
 - The bootloader is stored in (or referred to by) the boot sector, and loads the OS.
 - Master Boot Record is the BIOS-version of a partition table
 - GUID is the UEFI-version of a partition table
 - [POST](#) is Power On Self Test, and is recursively ran to test each part in your computer before running the OS.
 - Safe Mode is a mode in Windows that only loads basic drivers and no startup programs, and is useful for debugging.
 - Safe Mode with Networking loads networking
 - Safe Mode with Command Prompt opens a command prompt window for you
 - boot.ini is a text file that contains many boot configuration options
- Understand business continuity

- Backup and restore, disaster recovery planning, clustering, AD restore, folder redirection, data redundancy, uninterruptible power supply (UPS)
 - It is VITAL to [backup your Active Directory](#), since this holds all of the data and settings for your organization. AD backups are stored in VHD files, meaning that you can easily read files in them.
 - Business continuity is the planning to overcome incidents and resume normal operations. For example, if you lost power, you would need [a UPS](#) to keep continuity.
 - Backing up and restoring is useful and necessary to make sure you don't lose important data
 - Disaster recovery planning is planning for events such as natural events (tornadoes, floods, etc) and how you would go about dealing with them.
 - Clustering is to have multiple servers to achieve one action. This can be done for redundancy and for efficiency.
 - [A UPS is](#) basically a big battery that can keep your computer on during short power outages. For longer power outages, you would likely need a generator.
- Understand updates
 - Software, driver, operating systems, applications, Windows Update, Windows Server Update Service (WSUS)
 - WSUS is windows software for bulk or quicker windows updates. It is also a service that can be ran on windows servers for providing update files.
 - I hope you know what Windows Update does...
- Understand troubleshooting methodology
 - Processes, procedures, best practices; systematic vs. specific approach, Performance Monitor, Event Viewer, Resource Monitor, Information Technology Infrastructure Library, central logging, event filtering, default logs
 - [Information Technology Infrastructure Library](#), is a set of detailed practices for IT service management (ITSM) that focuses on aligning IT services with the needs of business. So the act of aligning what you do with what the business needs.
 - Systematic troubleshooting is more general, such as seeing a problem and restarting the computer.
 - Specific troubleshooting is more, well, specific. An example is knowing to go into safe mode to revert drivers.

- You could diagnose the problem:
 - Using Performance Monitor / Resource Monitor, what is taking all your resources
 - In Event Viewer, what started the chain reaction problem, when did the attacker sign in, etc.
- Central logging is important for tracking problems, and for making getting problems or issues easier.
 - A common protocol for this is Server Network Management Protocol (161)
- Event filtering is to find the event you are looking for, such as how event viewer has categories and can filter by time.

Preparation resources

- [Boot process and BCDEdit](#)
- [Active Directory backup and restore in Windows Server 2008](#)
- [Troubleshoot Windows Server 2008](#)

Mobility and Devices Fundamentals

Understand device configurations (20-25%)

- Configure device options
 - Describe Microsoft account, configure Control Panel, configure system options
 - Windows has offline accounts (only one device) and online accounts (usable by many devices)
- Configure desktop settings
 - Configure the Start menu, profiles, display settings, shortcuts, and group configurations and capabilities
 - To make a shortcut, right click something and then click "make shortcut"
 - To change display settings, search "display settings". From there, you can change things such as resolution.
- Configure drive encryption
 - Configure BitLocker, prepare for file encryption

- BitLocker is the program Windows uses for encryption. For file encryption, it is recommended to only encrypt sensitive data, since full disk encryption slows down everything. This can be done with folder encryption or encrypted partitions.
- Configure updates
 - Describe Windows Update, describe app updates, describe device system updates
 - [Windows App Updates](#) go through Windows Store

Understand data access and management (20-25%)

- Describe cloud storage services
 - Describe OneDrive, Microsoft Azure storage, OneNote, Outlook, and Office 365
 - OneDrive is like Google Drive
 - [Microsoft Azure](#) is like OneDrive but for server things
 - OneNote is like if MS Paint and MS Word had an ugly baby
 - Outlook is an online EMail service
 - Office 365 is a suite of all office tools
- Describe local storage
 - Describe storage spaces and storage pools
 - [Storage Space](#) is a simplified version of RAID. It is made up of storage pools (a pool of storage devices). It is well-paired with ReFS for redundancy.
- Describe file systems
 - Describe FAT, FAT32, exFAT, NTFS, and Resilient File System Overview (ReFS)
 - FAT stands for File Allocation Table and is a type of file system:
 - Fat32 has a max file size of 4GB
 - exFAT is a newer version of FAT with max file and volume sizes in petabytes
 - NTFS is the default windows file system. It has built-in features such as EFS and archiving, and file compression
 - Resilient File System has checksums of all files, so it can tell if they are corrupted. If they are, it uses the other device in the storage pool to seamless recover the data.
- Describe file and print sharing

- Describe NTFS and share permissions, HomeGroup, print drivers, and effective permissions; create public, basic, and advanced shares; map drives
 - HomeGroup is a feature that *used* to be in windows.
 - To join it, it requires you to be on a "private" network
 - You could easily share files and printers
 - To join an existing homegroup, you would type in a randomly generated password.
 - Effective permissions are when a folder gives permissions to its children. An example is a read-only folder would give its children read only attributes. This can be disabled (properties->security->advanced->disable inheritance)
- Describe data encryption
 - Describe encrypting file system (EFS); describe how EFS-encrypted folders impact moving and copying files; describe BitLocker To Go, virtual private network (VPN), public key, and private key; certificate services; and SSL
 - [Bitlocker To Go](#) is for mobile devices such as flash drives, where they are encrypted.
 - Encrypting File System is to encrypt part of a drive. To use it, simply right click a file and check the "encrypt" box.
 - A VPN is an encrypted tunnel connecting two devices/networks. It can provide security from public networks, or can extend a logical private network across multiple physical locations

[Understand device security \(20-25%\)](#)

- Describe the differences between the Internet, an intranet, and an extranet
 - Describe uses of private networks, public networks, host firewalls, network firewalls, and clouds
 - Private networks are networks such as a home or office. Public networks are networks you would find in places like coffee shops or airports, and are deemed as less secure.
 - Host firewalls run on end nodes such as your computer, and block unwanted information from being received or sent.
 - Network firewalls are vital because they use dedicated hardware to protect many computers
 - "Cloud" is a general term for external server service:
 - Leasing servers over the internet would be leasing cloud servers

- Leasing storage space from businesses like AWS or google drive would be cloud storage
- Describe user authentication
 - Describe Multifactor authentication, describe smart cards, describe Windows Rights Management Services, describe biometrics and password procedures
 - There are many [ways to authenticate](#), such as:
 - Something you know - password, pin, username
 - Something you are - fingerprint, retina scan
 - Something you do - signature, walk structure
 - Something you have - ID card, hardware key
 - Somewhere you are - in your house, at work
 - Multifactor authentication is using multiple types of authentication, such as a password (know) and card (have).
 - An example of single factor would be a username (know) and password (know).
 - [Smart cards](#) are cards that hold your password on them, so you can't authenticate without them.
 - Some hold a mathematical function on them, so they can't be read for the password and duplicated as easily
- Describe permissions
 - Configure file and share permissions; describe the behavior when moving or copying files from one location to another; describe basic and advanced permissions; describe taking ownership, delegating, and resultant permissions
 - Taking ownership is where you have full access of a file, such as the ability to delete or move it. With ownership you can transfer permissions and ownership.
 - The modify permission is like ownership, but without the ability to change permissions
- Describe malware
 - Describe computer viruses, Trojan horses, spyware, and adware; describe antivirus and antimalware strategies
 - Viruses self-replicate without user input
 - Trojan Horses pretend to be safe useful software, but secretly are contain malware

- Spyware listens to what you're doing, gathering personal information such as passwords or blackmail
- Adware spams you with ads for profit
- You can install antivirus/antimalware software, but the best protection is end user awareness (not to download untrusted files)

Understand cloud services (20-25%)

- Describe the types of cloud services
 - Describe productivity services, storage services, communications services, and search services
 - Cloud productivity and communication services would include work scheduling and task management software, such as [Any of these](#)
 - Cloud storage services are very common and offered by many technology companies, such as google drive, AWS storage, azure storage, etc.
 - [Cloud Search Services](#) are if you had a website or app that you wanted to be searchable.
- Describe Microsoft Intune
 - Describe Microsoft Intune capabilities, describe selective wipe, describe location settings
 - For this part, I'd recommend a 30-day free intune account, just to see the cool menus
 - [Selective wipe](#): So let's say your company own some windows-managed devices, and one gets stolen. Use selective wipe to wipe the company data from it.
 - It is confusing what this [Location Settings](#) is referring to, but it might be the Intune setting to disable the device's GPS location.
- Describe Microsoft Azure
 - Describe virtual machines; describe Azure services; storage tiers, describe disaster recovery, high availability, redundancy, and fault tolerance
 - Azure allows you to make virtual machines in the cloud.
 - Availability is how often a service is available. Most websites have something around 99.99999% availability.
 - Redundancy is having data duplicated across devices. Many "cloud" servers are redundant, commonly in many locations. An example is a server in each continent.

- Fault tolerance is where it is tolerant to problems. For example, backups would increase fault tolerance, RAID 1 would increase fault tolerance, and having two ISPs would increase fault tolerance.
- [Storage Tiers](#) are for when you're storing data:
 - **Hot** - Optimized for storing data that is accessed frequently.
 - **Cool** - Optimized for storing data that is infrequently accessed and stored for at least 30 days.
 - **Archive** - Optimized for storing data that is rarely accessed and stored for at least 180 days with flexible latency requirements (on the order of hours).

Understand enterprise mobility (20-25%)

- Describe identity services
 - Describe Windows Server Active Directory and Azure Active Directory, Microsoft Account, and federation services
 - [ADFS](#) is a system by windows where you, using active directory, can sign in to many things with only one authentication.
- Describe business data access
 - Describe Company Portal, describe Work Folders, Offline folders, describe Azure RemoteApp
 - A Windows company portal would be Office 365 (the online subscription)
 - OneDrive offline folders are downloaded to your device when you have internet so that you can access them offline.
 - **Azure RemoteApp literally ended two years ago. I don't know why it's on here.** [Learn about it here](#)
 - It was used to virtualize an app on windows servers
- Describe Bring Your Own Device (BYOD)
 - Describe device-centric to people-centric IT, describe desktop virtualization, describe Dynamic Access Control policies, describe Windows Rights Management
 - [People-centric](#) is IT where you give people ability to work more leniently, while device-centric is a more strict rule. The flaw of people-centric is company data could be more at risk and harder to account.

Cloud Fundamentals

This PDF has a good amount of knowledge. Most of my resources on Cloud Fundamentals are from it

Also, learn DNS

Understand the cloud (20-25%)

- Describe cloud principles and delivery mechanisms
 - Differentiate between on-premises IT service models; differentiate between subscription or pay-as-you-go versus upfront CapEx/OpEx funding model; use cloud services to expand capacity (elasticity of the cloud), scalability, redundancy, and availability; differentiate between cloud services that are configurable versus on-premises services that are customizable
 - [Capital Expense \(CapEx\)](#) is one time payments, such as new hardware
 - [Operational Expense\(OpEx\)](#) is expense with multiple payments, such as an employee's wage, a subscription service, etc.
- Describe cloud security requirements and policies
 - Describe how cloud services manage privacy, how compliance goals are met, how data is secured at rest or on-the-wire, and how data and operations transparency requirements are met
 - Data is secured at rest with disk/partition encryption (or bitlocker)
 - Data is secured on the wire with HTTPS, SSL, TLS, etc.
 - Operation transparency is commonly met through third party audits
- Describe how a cloud service stays up to date and available
 - Describe the service/feature improvement process; monitor service health, service maintenance, and future roadmap publishing; identify guarantees, service level agreements (SLA), and capping of liability of the cloud service provider
 - Service Level Agreements are a binding of responsibility between companies. It's like the legal version of promising you will do something.

- Describe the different types of cloud services
 - Differentiate between types of cloud services and their characteristics, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS); integrate the cloud with on-premises services in hybrid scenarios
 - [IBM description](#)

Enable Microsoft cloud services (20-25%)

- Identify the requirements and dependencies for using Office 365 and Microsoft Intune
 - Plan networking and domains, firewall rule, client requirements, bandwidth implications, and DNS
 - To use Office 365, you need a DNS domain (these can be free, such as a .tk domain)
 - Your firewall needs to be configured to allow internal office servers to be externally accessed
 - Your network plan could include setting up an Active Directory server and port forwarding to it.
- Select a cloud service plan
 - Understand the different options and plans available for Office 365 and Microsoft Intune
 - [Office 365 pricing](#)
 - [Intune pricing](#)
- Sign up for cloud services
 - Name your tenant, set up your first administrator, determine tenant location
 -
- Set up the initial configuration of cloud services
 - Register domains, verify domains, choose the domain purpose; identify required DNS record types
 - [How DNS works](#)
 - [DNS record types](#)

Administer Office 365 and Microsoft Intune (15-20%)

- Create users and groups, and assign services and licenses
 - Differentiate between cloud identities (Online identity, Synchronized identities, Federated identities), create and manage users and identities, delete and restore users (soft delete), create and manage groups, assign and revoke licenses, determine user locations

- There are different identities (where your account is stored):
 - Online is where it is stored and signed into on microsoft's servers
 - Synchronized is where you sign into onedrive online and it is synced with you also signing in on the local Active Directory.
 - Federated Identities is where it is **only** on your local Active directory, so be sure to have high availability.
 - Assign permissions in Office 365 and Microsoft Intune
 - Assign or revoke administrative roles; manage delegated admins; manage password policies, subscriptions, and licenses
 - Admin roles in Office 365 and Microsoft Intune: Global admin (full permissions), Billing admin (manage user licensing), Service admin, User management admin, and Password admin.
 - If a third party manages your Office 365, then you would be a delegated admin.
 - Monitor service health in Office 365 and Microsoft Intune
 - Monitor the Service Health dashboard, subscribe to RSS feeds, monitor the maintenance schedule, monitor the message center, log service support requests, configure alerts
 - The Service Health Dashboard can tell you if the server is scheduled for maintenance.

Use and configure Microsoft cloud services (20-25%)

- Configure Exchange Online
 - Manage recipients (mailboxes, shared mailboxes, resources, contacts, groups), manage anti-spam and antivirus settings
 - By default there is [spam protection](#)
- Configure SharePoint Online, including OneDrive
 - Create SharePoint team sites, configure external sharing, set up social features using newsfeeds or Yammer, apply themes, set storage and resource limits
 - Team sites are good for child departments. [Yammer](#) is a company messaging service
- Configure Skype for Business Online
 - Manage Skype for Business user options, manage external communication settings, configure dial-in settings and meeting invitation options, configure Skype for Business Online DNS

- [Dial-in conferencing](#) is where users enter a conference by a short passphrase
- Configure Microsoft Intune
 - Install Microsoft Intune client management software, create and deploy policies, automate installs, identify software requirements, set up notifications, identify mobile device management policies
 - Intune says it works on any non-deprecated OS

Support cloud users (15-20%)

- Resolve sign-in and Office application installation issues
 - Troubleshoot sign-in issues, forgotten passwords, connection problems, difficulty activating Office applications, and difficulty connecting mobile devices to Office 365 or Microsoft Intune; choose between 32-bit and 64-bit; identify when an Office repair is required; identify operating system requirements for Office 365 ProPlus; browser requirements and specific versions of Internet Explorer
 - 64-bit is recommended for 64-bit devices, while 32-bit is recommended for 32-bit devices. If you are unsure, get 32-bit (or on windows 10, do WIN+x+y to see what architecture you have)
- Resolve email and calendar issues
 - Troubleshoot issues receiving and sending email, troubleshoot issues accessing a delegated mailbox
 - A user can track emails using Delivery Reports. Delivery reports allow a user to check the delivery status of an email message
 - An administrator use Message Trace to track emails. Message Trace allows an administrator to locate missing emails.
- Resolve SharePoint and OneDrive issues
 - Identify storage limits, troubleshoot “Open with Explorer” not working and OneDrive not syncing, recover deleted files
 - OneDrive not syncing is likely an internet problem, but it could also be that you aren't signed in.
 - One drive has a 1TB limit per user for <5 users, and if more, a 5TB limit/user. If you need more, contact Microsoft for 25TB/user
- Resolve Skype for Business issues

- Troubleshoot Skype for Business sign-in issues, troubleshoot connection issues to Skype for Business Online, troubleshoot communicating to Skype consumer users and users in other companies using Skype for Business
 - Make sure the account is created, they are using the right username/password, etc.