



Harvard Extension School
HARVARD DIVISION OF CONTINUING EDUCATION

CSCI E-94

Fundamentals of Cloud Computing and OpenAI with Microsoft Azure

Spring Term 2025

Course Information

CRN: 25152

Section Number: 1

Format: Flexible Attendance Web Conference

Credit Status: Undergraduate, Graduate, Noncredit

Credit Hours: 4

Class Meetings: Thursdays, January 30-May 17, 6:30pm-8:30pm

Course Description: This course offers a comprehensive introduction to cloud and serverless computing, focusing on Microsoft Azure's key services and Azure's artificial intelligence (AI) platform. We contrast the challenges and benefits of cloud computing, serverless cloud computing, and traditional self-managed cloud and on-premises solutions. Students learn the fundamental architecture and design patterns necessary to build geographically distributed, highly available, and scalable solutions using key services in the Microsoft Azure platform. Students learn about the OpenAI offerings in Azure and how to responsibly leverage them in their cloud-native solutions, with hands-on experience in prompt engineering, fine tuning, and embedding. We cover the costs and benefits of each and how to pragmatically apply them. Students engage in hands-on learning architecting secure, scalable, geo-redundant, and cost-effective infrastructure and deploying that infrastructure to Microsoft Azure using infrastructure as code via the Bicep language. Students learn approaches for building solutions that gracefully degrade when non-essential functionality is unavailable. Students learn to implement defense in depth using network segmentation (VNETs) and additional best practices. A wide range of Microsoft Azure Services are covered including Azure Front Door, Azure App Services, Azure App Configuration, KeyVault, Azure SQL, Azure API management;

serverless services including Azure Functions, and Azure Logic Applications; Microsoft Entra ID (formerly Azure Active Directory) for enterprise identity management; Azure B2C for low-cost identity management on consumer-centric software-as-a-service offerings; and Azure Storage, Azure Service Bus, Azure Event Grid, Azure Event Hub, Azure Cosmos database, and Azure Cognitive Search. In addition to Azure services and guidance, the course covers implementing processes to streamline development, such as continuous integration, continuous deployment (CI/CD), and automated testing. Students also learn how to test their applications and infrastructure at scale using Azure Load Testing. Coverage includes always-up architecture and deployment strategies, rollback strategies, A/B testing, testing in production, monitoring, distributed tracing, alerting, performance tuning, snapshot debugging in production, and health analysis using Application Insights and Azure Monitor. Additionally, students learn strategies and architecture for ensuring data sovereignty concerns are addressed in their solutions.

Prerequisites: Basic C#, C++, Python, or Java development skills. CSCI E-10a or the equivalent. This course involves a substantial amount of programming in C# and cross platform.NET (6 or above).

Instructor Information & Office Hours

Joseph Ficara

Email: jficara@fas.harvard.edu

Section Meetings

Sections are live-streamed optional weekly discussion groups run by our teaching assistants (TAs). Student questions regarding lecture material and assignments will be addressed during this time. Periodically, TAs may provide supplementary material during the section. Details will be announced in class and available on the website and in Microsoft Teams.

Note: The first several sections will introduce the C# programming language and .NET 8.

Thursdays beginning Feb. 6th, 2025

Location: Zoom conference link.

The section zoom link will be posted in the sections channel in Microsoft Teams as well as on the course site

Thursdays held from 5:30 PM - 6:30 PM EST

Course Goals / Learning Outcomes

Upon completion of this course, you will be able to

- **Cloud Architecture, Computing & Security**
 - Understand and apply fundamental cloud architecture patterns and best practices to build highly scalable, secure, cloud-native applications
 - Build cloud-native applications from scratch using Microsoft .NET 9, as well as apply this knowledge to other technologies.
 - Architect and implement REST interfaces that support industry best practices and conventions.
 - Understand Relational versus NoSQL database technologies and what they are best suited for
 - Understand techniques for converting monolithic applications to macro & microservice applications and when it is and is not appropriate to do so.
 - Understand the fundamentals of macro and microservice architectures and how they can be used to facilitate separate deployments, scalability, graceful degradation, and reduced friction in larger engineering teams
 - Understand defense in depth with network segmentation and virtual networks (VNETs).
 - Understand how serverless computing and, Platform as a Service, PaaS, can be effectively used to create high-density cloud-native applications that control cost and allow for just-in-time scale.
 - Build applications that leverage key Azure PaaS and serverless offerings that are appropriate to the problem to be solved.
- **Azure AI Offerings and AI related content**
 - Explore OpenAI offerings on Azure for cloud-native solutions.

- Acquire skills for utilizing OpenAI models on Azure in various scenarios.
- Understand how to apply Azure OpenAI services to real-world applications.
- Learn best practices for integrating and optimizing OpenAI offerings within cloud-native solutions on Azure.
- Understand the fundamentals of Microsoft Semantic Kernel and how it integrates large language models (LLMs) into AI applications.
- Develop and orchestrate AI workflows using kernel functions, plugins, for real-world use cases.
- Apply Semantic Kernel in building scalable AI solutions, leveraging domain-specific logic alongside LLMs to enhance enterprise applications.
- How to utilize the Agent Framework to perform tasks autonomously or semi-autonomously
- **DevOps and Production Readiness**
 - Setup continuous integration and continuous development, CICD, and how CICD can be used to reduce friction in development and deployment.
 - Engage in hands-on experience architecting secure, scalable, georedundant, and cost-effective infrastructure using Bicep language.
 - Understand and implement feature flags to control the release of functionality to users.
 - Setup monitoring and alerting to facilitate proactive issue mitigation.
 - Architect and implement rollback strategies for always up applications.

Mode of Attendance & Participation Policy

Class sessions are conducted live via Zoom to foster engagement through active participation, discussions, and dialogue. Your attendance is highly recommended to benefit from real-time interaction. Recognizing the need for flexibility, our course offers an alternative for those who cannot attend the scheduled times—recordings of each lecture are available for you to view at your convenience. To further support your learning, we offer a dedicated Q&A platform through Microsoft Teams, where you can ask questions anytime. Moreover, we hold weekly sessions and provide access to instructor office hours for personalized assistance and additional support.

If you plan to attend the course when it is held, please arrive on time. You should attend Zoom meetings with a functional web-camera and microphone, prepared with materials needed, to engage thoughtfully, and with your camera on. You may turn off your camera for occasional interruptions or momentarily for privacy.

You will also need the most up-to-date Zoom client installed on your computer to join class. Please participate from a safe and appropriate environment with appropriate clothing for class. Participating while traveling or in a car is not permitted. In addition, please do not join class via mobile phone or web browser.

Please be sure to review important information on [Student Policies and Conduct](#).

Assignments & Grading

Summary:

Your final grade is the average of the 6 assignments with the highest scores out of the 7 assignments assigned.

Details:

It is likely that in at least one homework assignment, you will score less than 100%. We will automatically drop the homework assignment with the lowest grade. If you elect not to submit one of the assignments, then that homework assignment will count as your lowest graded homework assignment and will be dropped.

Since Azure Development is ever evolving at internet speed, some Web assignments may be modified. I reserve the right to change them at any time, up to the end of class on the night on which they are assigned.

Each assignment has a specific due date. The student must post her/his assignment to the canvas website's drop box by 5:00 PM EST, the day it is due. There is a fifteen-minute grace period for late assignments (i.e., assignments turned in at 5:16 PM EST or later will be counted as one day late).

Penalties for Late Assignments

1. 1.5 points per day late up to a maximum of 21 points.

The drop box date/time stamp determines the time of submission.

2. Assignments turned in more than 14 days late will receive a grade of 0%
3. 10 to 100 Points or expulsion for submitting an assignment that contains a virus depending on the damage done to the TAs machine.
4. 5 Points for not including your name, email address, assignment title, and number.
5. Assignments not received by May 15th, 2025, at 11:59 PM EST will be assigned a grade of 0%

Extra Credit

It is possible to get some extra points for turning an assignment in early. We will grant you one point for each day an assignment is turned in early, up to a maximum of five points. To be considered early, the assignment's submission must be turned in by 5:00 PM EST before the day it is due. The 15-minute grace period applies here as well.

For example: If the assignment is due on 2/13/2025 and it's turned in on 2/13/2025 at 5:15 PM EST, it is eligible for one point of extra credit. If it is turned in on 2/13/2025 at 5:16 PM EST, it is not eligible for the one point of extra credit.

Resubmits

A resubmit is only permitted if requested by the TA. If a resubmit occurs before the due date, all extra credit points for early submission are forfeit; if a resubmit occurs after the due date, the late penalty policy will apply.

The last day an assignment can be turned in:

Any work not received by May 15th, 2025, at 11:59 PM EST, will be assigned a grade of 0%. Extensions beyond May 15th, 2025, will only be granted in cases of life-or severe health-threatening emergencies. Any other cause, including but not limited to business problems, travel, and computer breakdowns, does not qualify for an extension.

If Harvard Extension cancels a class

If the Harvard Extension School officially cancels a class, the assignment to be given at that class will be assigned at the next class, and the due date will be moved accordingly. Assignments due on the night of the canceled class are still due at their original date & time.

Grade Definitions

Grading criteria (grading percentages)

Grade	A	A-	B+	B	B-	C+	C	C-	D+	D	D-	E
Points	100-92	91-90	89-87	86-83	82-80	79-77	76-73	72-70	69-67	66-63	62-60	Less than 60

A and A– Earned by work whose superior quality indicates a full mastery of the subject—and in the case of A, work of extraordinary distinction. There is no grade of A+.

B+, B, and B– Earned by work that indicates a strong comprehension of the course material, a good command of the skills needed to work with the course materials, and the student's full engagement with the course requirements and activities.

C+, C, and C– Earned by work that indicates an adequate and satisfactory comprehension of the course material and the skills needed to work with the course materials, and that indicates that the student has met the basic requirements for completing assigned work and participating in class activities.

D+, D, and D– Earned by work that is unsatisfactory but that indicates some minimal command of the course materials and some minimal participation in class activities that is worthy of course credit.

E Earned by work that is unsatisfactory and unworthy of course credit. This grade may also be assigned to students who do not submit required work in courses from which they have not officially withdrawn by the withdrawal deadline. Zero or E grades are assigned to students for missing work. These grades are included in the calculation of the final grade.

Graduate Credit Requirements

This course is available for both undergraduate and graduate credit. Graduate students will be required to complete additional advanced components for each assignment. Undergraduate students have the option to attempt these advanced elements for extra credit, but they are not mandatory.

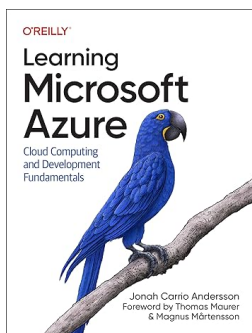
Coursework submission policy

Submissions shall be placed in a single zip file that preserves the directory structure. The zip file shall contain:

1. All assets such as configuration files, images, initialization scripts, and resources that are part of the application must be included in the assignment submission. The TA must be able to build and deploy your homework assignment to Azure and experience the full functionality of your application.
2. A text file named ProjectNotes.txt that includes:
 1. The homework assignment tile and number
 2. Your name and email address
 3. Any notes for the TA needed to install, setup, login to, and operate your homework assignment must be included in the assignment submission.

If you have any questions regarding these guidelines, please ask a TA or the instructor.

Course Materials



Optional Book: Learning Microsoft Azure: Cloud Computing and Development Fundamentals

ISBN: 978-1098113322

Authors: Jonah Carrio Andersson

Publisher: O'Reilly Media

Publication Date: 2023-12-26

Edition: 1st Edition



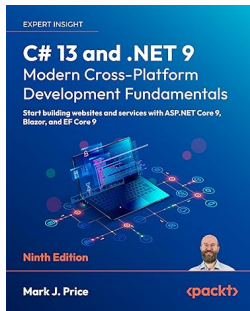
Optional Book: Azure for Developers - Second Edition

ISBN: 9781803240091

Authors: Kamil Mrzygłod

Publication Date: 2022-08-30

Optional Book: C# 13 and .NET 9 – Modern Cross-Platform Development Fundamentals: Start building websites and



services with ASP.NET Core 9, Blazor, and EF Core 9

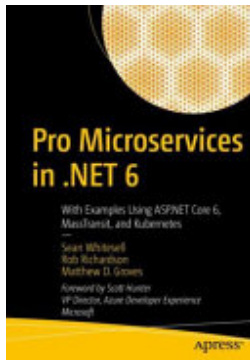
ISBN: 978-1835881224

Authors: Mark J. Price

Publisher: Packt Publishing - ebooks Account

Publication Date: 2024-12-10

Edition: 9th



Optional Book: Pro Microservices in .NET 6

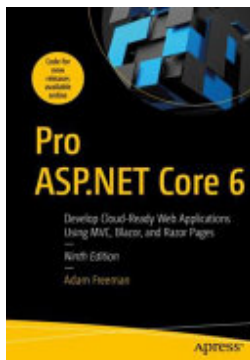
ISBN: 9781484278321

Authors: Sean Whitesell, Rob Richardson, Matthew D. Groves

Publisher: Apress

Publication Date: 2022-01-16

This book is optional, but recommended if you are new to .NET or .NET 6



Optional Book: Pro ASP.NET Core 6

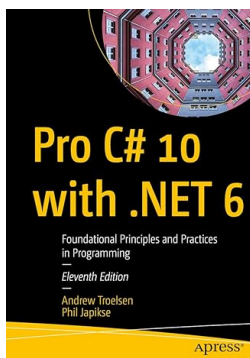
ISBN: 9781484279564

Authors: Adam Freeman

Publisher: Apress

Publication Date: 2022-03-11

This book is optional but recommended if you are new to .NET



Optional Book: Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming

ISBN: 9781484278680

Authors: Andrew Troelsen, Philip Japikse

Publisher: Apress

Publication Date: 2022-07-31

This book is optional but recommended if you are new to C#

You are responsible for understanding Harvard Extension School policies on [Academic Integrity](#) and how to use sources responsibly. Violations of academic integrity are taken very seriously. Visit [Using Sources Effectively and Responsibly](#) and the [Harvard Guide to Using Sources](#) to review important information on academic citation rules.

AI Technologies. The Extension School's [Academic Integrity Policy](#) prohibits students from representing work as their own that they did not write, code, or create. It is never permissible to submit work generated by machine learning and AI technologies (such as ChatGPT) without proper attribution. Alternatively, your instructor may prohibit all use of AI technologies in their course.

Writing Code. While it may be common practice in non-academic settings to adapt code examples found online or in texts, this is not the case in academia. In particular, you should never copy code produced as coursework by other students, whether in the current term or a previous term; nor may you provide work for other students to use. Copying code from another student or any other source is a form of academic dishonesty, as is deriving a program substantially from the work of another.

Writing code is similar to academic writing in that when you use or adapt code developed by someone else as part of your assigned coursework, you must cite your source. Paraphrasing without proper citation is just as dishonest with programming as it is with prose. A program can be considered plagiarized even though no single line is identical to any line of the source.

Accessibility Services Policy

The Division of Continuing Education (DCE) is committed to providing an accessible academic community. The [Accessibility Services Office \(ASO\)](#) is responsible for providing accommodations to students with disabilities. Students must request accommodations or adjustments through the ASO. Instructors cannot grant accommodation requests without prior ASO approval. It is imperative to be in touch with the ASO as soon as possible to avoid delays in the provision of accommodation.

DCE takes student privacy seriously. Any medical documentation should be provided directly to the ASO if a substantial accommodation is required. If you miss class due to a short-term illness, notify your instructor and/or TA but do not include a doctor's note. Course staff will not request, accept, or review doctor's notes or other medical documentation. For more information, email accessibility@extension.harvard.edu.

Publishing or Distributing Course Materials Policy

Students may not post, publish, sell, or otherwise distribute course materials without the written permission of the course instructor. Such materials include, but are not limited to, the following: lecture notes, lecture slides, video, or audio recordings, assignments, problem sets, examinations, other students' work, and answer keys. Students who sell, post, publish, or distribute course materials without written permission, whether for the purposes of soliciting answers or otherwise, may be subject to disciplinary action, up to and including requirement to withdraw. Further, students may not make video or audio recordings of class sessions for their own use without written permission of the instructor.

Canvas Access After End of Term

The Canvas website for this course will remain available to enrolled students for a limited time after the course concludes. **You are encouraged to download coursework and materials you wish to keep *before* the term ends.** See [Course Formats & Required Technology](#) for additional information on Canvas access.

Class Meeting Schedule

January 30: Class 1

- Course Overview
- Introduction
 - Visual Studio 2022 & Automated Testing
 - Azure App Services
 - .NET Web API
 - Integrating AI into your applications
- **Assignment 1 assigned and is due on February 13th at 5:00 PM EST.**

February 6: Class 2

- REST Fundamentals
- Azure Monitor / Application Insights

- Introduction to Bicep and infrastructure as code

February 13: Class 3

- Azure SQL Fundamentals
- Entity Framework Core Fundamentals
- **Assignment 2 assigned and is due on February 27th at 5:00 PM EST.**

February 20: Class 4

- Introduction to Network Segmentation and VNETS for defense in depth
- Azure Storage Fundamentals Part I

February 27: Class 5

- Azure Storage Fundamentals Part II
- **Assignment 3 assigned and is due on March 13th at 5:00 PM EST.**

March 6: Class 6

- Serverless using Azure Functions

March 13: Class 7

- WebJobs
- Serverless using Logic Apps
- **Assignment 4 assigned and is due on April 3rd at 5:00 PM EST.**

March 20: *Spring Break* - NO CLASS

- NO CLASS

March 27: Class 8

- Azure AD & OAuth Fundamentals
- Azure Web Apps & .NET Core MVC Fundamentals

April 3: Class 9

- API Management
- **Assignment 5 assigned and is due on April 17th at 5:00 PM EST.**

April 10: Class 10

- Azure Service Bus
- Microservices using .NET Core and Azure

April 17: Class 11

- Azure Cosmos DB
- Azure Cognitive Search
- **Assignment 6 assigned and is due on May 1st at 5:00 PM EST.**

April 24: Class 12

- Introduction to Azure OpenAI and Basic Concepts
- Semantic Kernel - Part I - Integrate AI Models into your applications

May 1: Class 13

- Working with OpenAI Models on Azure, RAG Overview and Hands-on Implementation
- **Assignment 7 assigned and is due on May 15th at 5:00 PM EST.**

May 8: Class 14

- Semantic Kernel - Part II Agent Framework

May 15: (Final class meeting)

- Prompt Engineering, Model Selection, Fine Tuning and AI Considerations beyond engineering