MS.NET TECHNOLOGIES MCQ BANK
Total No. Of MCQs : 722
Name : Shubham Shivaji Patil

1. Lecture: Introduction to .Net Framework and CLR

1.1. Introduction to the .Net Framework

Q1. What is the .NET Framework?
a) A programming language developed by Microsoft
b) A collection of programming libraries and tools for building Windows applications
c) An operating system developed by Microsoft
d) A hardware architecture used in supercomputers

Answer: b) A collection of programming libraries and tools for building Windows applications

Q2. What are the key components of the .NET Framework?
a) IDE, CLR, IL, HTTP
b) CLR, BCL, CIL, JIT
c) C#, VB.NET, CIL, ASP.NET
d) BCL, CLR, FCL, HTTP

Answer: b) CLR, BCL, CIL, JIT

Q3. Which programming languages are commonly used with the .NET Framework?
a) Python and Java
b) C++ and Ruby
c) C# and VB.NET
d) PHP and JavaScript

Answer: c) C# and VB.NET

Q4. In which year was the first version of the .NET Framework released?
a) 2000
b) 2001
c) 2002
d) 2003

Answer: b) 2001

Q5. What is the purpose of the Common Language Runtime (CLR) in the .NET Framework?
a) To compile source code into machine code
b) To provide a development environment for building applications
c) To manage memory and resources for .NET applications
d) To ensure cross-platform compatibility of .NET applications

Answer: c) To manage memory and resources for .NET applications

1.2. Intermediate Language (IL)

Q6. What is Intermediate Language (IL)?
a) A low-level assembly language specific to the .NET Framework
b) A high-level language used for application development
c) A scripting language for web development in .NET
d) A markup language for creating user interfaces

Answer: a) A low-level assembly language specific to the .NET Framework

Q7. Which part of the .NET Framework is responsible for converting IL code into machine code?
a) Just-In-Time (JIT) Compiler
b) Intermediate Language Compiler (ILC)
c) Common Language Specification (CLS)
d) Common Type System (CTS)

Answer: a) Just-In-Time (JIT) Compiler

Q8. IL code is platform-specific. (True/False)

Answer: False

Q9. Which of the following is true about Intermediate Language (IL)?
a) IL code is directly executed by the CPU
b) IL code is human-readable and writable
c) IL code is specific to Windows operating system
d) IL code is always interpreted by the .NET runtime

Answer: d) IL code is always interpreted by the .NET runtime

Q10. What is the benefit of using IL in the .NET Framework?
a) Improved performance compared to machine code
b) Platform independence and easy portability
c) Easier debugging of IL code compared to other languages
d) Ability to directly execute IL code on any hardware architecture

Answer: b) Platform independence and easy portability

1.3. Assemblies and their structure, EXEs/DLLs

Q11. What is an assembly in the .NET Framework?
a) A collection of source code files
b) A compiled unit of code that contains metadata and IL code
c) A group of classes from the Base Class Library (BCL)
d) An executable file written in C# or VB.NET

Answer: b) A compiled unit of code that contains metadata and IL code

Q12. Which type of files can an assembly contain?
a) Only EXE files
b) Only DLL files
c) Both EXE and DLL files
d) Only source code files

Answer: c) Both EXE and DLL files

Q13. What is the purpose of the Global Assembly Cache (GAC) in the .NET Framework?
a) To store globally accessible assemblies for sharing across multiple applications
b) To store the .NET Framework runtime files
c) To provide a cache for temporary storage of IL code
d) To store configuration files for .NET applications

Answer: a) To store globally accessible assemblies for sharing across multiple applications

Q14. How are private assemblies deployed in .NET applications?
a) They are installed in the Global Assembly Cache (GAC).
b) They are copied to the application's directory or a subdirectory.
c) They are embedded within the main executable (EXE) file.
d) They are automatically downloaded from the internet when needed.

Answer: b) They are copied to the application's directory or a subdirectory.

Q15. What is the purpose of metadata in the .NET Framework assembly?
a) To store the source code of the application
b) To provide information about the assembly's structure and types
c) To optimize the execution of IL code
d) To enable hardware-specific optimizations

Answer: b) To provide information about the assembly's structure and types

1.4. CLR and its functions

Q16. What does CLR stand for in the .NET Framework?
a) Common Language Runtime
b) Compiled Language Resolver
c) Code Language Registry
d) Core Library Resolver

Answer: a) Common Language Runtime

Q17. Which of the following is NOT a function of the Common Language Runtime (CLR)?
a) Memory management and garbage collection
b) Code verification and type safety
c) Just-In-Time (JIT) compilation of IL code
d) Interpreting source code during runtime

Answer: d) Interpreting source code during runtime

Q18. What is Just-In-Time (JIT) compilation in the .NET Framework?
a) Pre-compiling IL code to machine code before execution
b) Translating IL code to JavaScript for web applications
c) Converting source code to IL code during the build process
d) Dynamically compiling IL code into machine code during runtime

Answer: d) Dynamically compiling IL code into machine code during runtime

Q19. What is the role of the Common Type System (CTS) in the .NET Framework?
a) To ensure that all .NET languages can interoperate seamlessly
b) To manage memory allocation for objects and variables
c) To provide a set of common types for basic data structures
d) To handle security permissions for .NET applications

Answer: a) To ensure that all .NET languages can interoperate seamlessly

Q20. How does the CLR handle exceptions in .NET applications?
a) By preventing exceptions from occurring in the code
b) By automatically recovering from exceptions and resuming execution
c) By throwing exceptions and stopping program execution
d) By catching and handling exceptions through try-catch blocks

Answer: d) By catching and handling exceptions through try-catch blocks

1.4.1. JIT Compilation

Q21. What does JIT Compilation stand for in the context of the .NET Framework?
a) Just-In-Time Compilation
b) Java Integrated Tools Compilation
c) Just-In-Time Conversion
d) Java Intermediate Type Compilation

Answer: a) Just-In-Time Compilation

Q22. What is the main advantage of JIT Compilation in the .NET Framework?
a) It produces smaller executable files.
b) It enables cross-platform

compatibility of .NET applications.
c) It allows code to be optimized for the target machine's architecture.
d) It eliminates the need for a separate runtime environment.

Answer: c) It allows code to be optimized for the target machine's architecture.

Q23. When does JIT Compilation occur in the .NET Framework?
a) During the installation of the .NET Framework on the system
b) During the build process of the application
c) When the application is launched and the IL code is executed
d) During the development phase of the application

Answer: c) When the application is launched and the IL code is executed

Q24. What is the role of the JIT compiler in the .NET Framework?
a) To convert C# or VB.NET source code to machine code
b) To convert IL code to machine code just before execution
c) To convert JavaScript code to IL code for better performance
d) To interpret IL code and execute it directly

Answer: b) To convert IL code to machine code just before execution

Q25. Which of the following is true about JIT Compilation?
a) It compiles all the IL code in an assembly at once.
b) It converts IL code into platform-specific assembly language.
c) It generates machine code for all the methods in an assembly during build time.
d) It can be disabled to improve application performance.

Answer: b) It converts IL code into platform-specific assembly language.

1.4.2. Memory Management

Q26. What is memory management in the context of the .NET Framework?
a) The process of allocating memory to variables and objects during runtime
b) The process of converting IL code into machine code
c) The process of organizing files and folders in the Global Assembly Cache (GAC)
d) The process of optimizing code for better performance

Answer: a) The process of allocating memory to variables and objects during runtime

Q27. How does the .NET Framework manage memory for objects that are no longer in use?
a) By automatically deallocating memory using JIT Compilation
b) By using the Common Type System (CTS) to optimize memory usage
c) By employing the Common Language Runtime (CLR) to perform garbage collection
d) By explicitly releasing memory using the "delete" keyword

Answer: c) By employing the Common Language Runtime (CLR) to perform garbage collection

Q28. What is Garbage Collection (GC) in the .NET Framework?
a) The process of reclaiming memory occupied by unreferenced objects
b) The process of freeing memory during compile-time
c) The process of converting IL code to machine code
d) The process of organizing assemblies in the Global Assembly Cache (GAC)

Answer: a) The process of reclaiming memory occupied by unreferenced objects

Q29. How does Garbage Collection (GC) work in the .NET Framework?
a) It deallocates memory as soon as an object is no longer needed.
b) It runs at regular intervals and identifies and removes unused objects.
c) It relies on the developer to manually release memory when needed.
d) It requires the programmer to explicitly call a GC function.

Answer: b) It runs at regular intervals and identifies and removes unused objects.

Q30. Which of the following statements is true about Garbage Collection (GC) in the .NET Framework?
a) Garbage Collection can be disabled if the application requires manual memory management.
b) Garbage Collection is always running continuously during the application's execution.
c) Developers have direct control over when Garbage Collection runs.
d) Garbage Collection can improve performance by automatically allocating memory.

Answer: d) Garbage Collection can improve performance by automatically allocating memory.

1.4.2.1. Garbage Collection

Q31. What is the primary purpose of Garbage Collection (GC) in the .NET Framework?
a) To prevent memory leaks in .NET applications
b) To speed up the execution of IL code
c) To ensure that all objects are deallocated during compile-time
d) To manage hardware resources efficiently

Answer: a) To prevent memory leaks in .NET applications

Q32. When does Garbage Collection (GC) occur in the .NET Framework?
a) Only when the application is first launched
b) During the build process of the application
c) At regular intervals during the application's execution
d) Only when the application is closed

Answer: c) At regular intervals during the application's execution

Q33. Which of the following is true about the generations used in Garbage Collection (GC)?
a) Objects are always placed in Generation 0 before promotion to higher generations.
b) Generation 2 contains short-lived objects, and Generation 0 contains long-lived objects.
c) Newly created objects are always placed in Generation 2.
d) Garbage Collection starts with Generation 2 and works its way down to Generation 0.

Answer: d) Garbage Collection starts with Generation 2 and works its way down to Generation 0.

Q34. How does Garbage Collection (GC) handle unreachable objects that are not yet deallocated?
a) It immediately deallocates the unreachable objects.
b) It promotes them to higher generations for later deallocation.
c) It throws an exception to alert the programmer about the issue.
d) It leaves them in memory for better performance.

Answer: b) It promotes them to higher generations for later deallocation.

Q35. What can be done to improve the performance of Garbage Collection (GC) in .NET applications?
a) Increasing the frequency of GC execution
b) Reducing the amount of memory allocated for the application
c) Writing code with more objects to encourage GC to run more often
d) Designing the application to minimize object references and memory churn

Answer: d) Designing the application to minimize object references and memory churn

1.4.2.2. AppDomain Management

Q36. What is an AppDomain in the .NET Framework?
a) A virtual machine that runs .NET applications
b) An isolated environment for running .NET applications securely
c) A directory where assemblies are stored on the file system
d) A collection of classes and methods in the .NET Base Class Library (BCL)

Answer: b) An isolated environment for running .NET applications securely

Q37. Why is AppDomain Management important in the .NET Framework?
a) It allows the .NET runtime to handle memory management more efficiently.
b) It enables the .NET Framework to run applications on multiple operating systems.
c) It prevents different .NET applications from interfering with each other.
d) It simplifies the process of deploying .NET applications on web servers.

Answer: c) It prevents different .NET applications from interfering with each other.

Q38. What is the primary benefit of using multiple AppDomains in a single process?
a) Improved security and isolation between application components
b) Faster execution of IL code due to parallel processing
c) Reduced memory consumption for the .NET runtime
d) Simplified debugging and error handling

Answer: a) Improved security and isolation between application components

Q39. How can you unload an AppDomain in the .NET Framework?
a) By manually calling the Unload() method of the AppDomain class
b) By restarting the entire application process
c) By disabling Garbage Collection for that specific AppDomain
d) By deleting the application's executable file from the file system

Answer: a) By manually calling the Unload() method of the AppDomain class

Q40. Which of the following is true about AppDomain Management in the .NET Framework?
a) AppDomains cannot communicate with each other.
b) AppDomains always share the same memory space.
c) AppDomains provide a lightweight form of process isolation.
d) AppDomains are automatically created for each .NET application.

Answer: c) AppDomains provide a lightweight form of process isolation.

1.4.3. CLS, CTS

Q41. What do CLS and CTS stand for in the context of the .NET Framework?
a) Common Language Specification and Common Type Specification
b) Common Language Standard and Common Type Standard
c) Common Language Specification and Common Type System
d) Common Language Structure and Common Type Structure

Answer: c) Common Language Specification and Common Type System

Q42. What is the Common Language Specification (CLS) in the .NET Framework?
a) A set of guidelines for writing cross-platform applications
b) A specification for interoperability between different .NET languages
c) A standard for organizing files and folders in the Global Assembly Cache (GAC)
d) A collection of common data types used in all .NET applications

Answer: b) A specification for interoperability between different .NET languages

Q43. Which of the following statements is true about the Common Type System (CTS)?
a) It provides a set of common data types used in all .NET applications.
b) It defines the rules for writing cross-platform applications.
c) It is specific to a particular programming language, such as C# or VB.NET.
d) It is responsible for converting IL code into machine code.

Answer: a) It provides a set of common data types used in all .NET applications.

Q44. Why is the Common Type System (CTS) essential for .NET applications?
a) It ensures that all .NET applications use the same programming language.
b) It allows .NET applications to interoperate with COM components.
c) It guarantees that all .NET applications are compatible with Windows operating systems.
d) It ensures that different .NET languages can seamlessly work together.

Answer: d) It ensures that different .NET languages can seamlessly work together.

Q45. Which of the following is an advantage of adhering to the Common Language Specification (CLS) rules in .NET development?
a) It results in smaller executable files.
b) It improves the performance of IL code.
c) It guarantees platform-specific optimization.
d) It enhances the compatibility of .NET components across languages.

Answer: d) It enhances the compatibility of .NET components across languages.

1.4.4. Security

Q46. What is Code Access Security (CAS) in the .NET Framework?
a) A mechanism to prevent unauthorized code from performing harmful actions
b) A set of cryptographic algorithms for securing data transmission
c) A feature that encrypts IL code to protect intellectual property
d) A method of protecting against SQL injection attacks

Answer: a) A mechanism to prevent unauthorized code from performing harmful actions

Q47. Which aspect of .NET Framework security prevents unauthorized code from performing harmful actions?
a) Garbage Collection
b) Just-In-Time Compilation
c) Code Access Security (CAS)
d) Intermediate Language (IL) verification

Answer: c) Code Access Security (CAS)

Q48. What is the purpose of the Evidence-based Security model in the .NET Framework?
a) To allow any code to run with full trust by default
b) To ensure that all code runs with minimum permissions
c) To grant administrative privileges to all .NET applications
d) To provide dynamic code permissions based on the source of the code

Answer: d) To provide dynamic code permissions based on the source of the code

Q49. Which of the following statements is true about .NET Framework security?
a) Code Access Security (CAS) is only applicable to web applications.
b) .NET applications always run with full trust by default.
c) Security permissions cannot be customized in the .NET Framework.
d) Code running in the same AppDomain has identical permissions.

Answer: d) Code running in the same AppDomain has identical permissions.

Q50. How can you grant additional permissions to a specific .NET application?
a) By modifying the Global Assembly Cache (GAC) settings
b) By changing the permissions of the application's executable file
c) By using the Code Access Security Policy tool (caspol.exe)
d) By altering the .NET Framework's security settings in the Control Panel

Answer: c) By using the Code Access Security Policy tool (caspol.exe)

2. Lecture: .Net Framework Versions and Visual Studio Basics

2.1. .Net Framework, .Net Core, Mono, Xamarin differences

Q1. Which of the following is a key difference between .NET Framework and .NET Core?
a) .NET Core supports only Windows, while .NET Framework is cross-platform.
b) .NET Core includes the full Base Class Library (BCL), whereas .NET Framework has a limited BCL.
c) .NET Core is primarily used for building desktop applications, while .NET Framework is for web applications.
d) .NET Core applications require a separate runtime installation, while .NET Framework applications do not.

Answer: a) .NET Core supports only Windows, while .NET Framework is cross-platform.

Q2. What is the main advantage of using Mono in the .NET ecosystem?
a) Mono allows developers to write applications in multiple programming languages.
b) Mono enables cross-platform development for .NET applications.
c) Mono provides a comprehensive set of tools for web development.
d) Mono enhances the performance of .NET applications on Windows machines.

Answer: b) Mono enables cross-platform development for .NET applications.

Q3. Xamarin is mainly used for developing what type of applications?
a) Web applications
b) Desktop applications
c) Mobile applications
d) Embedded systems

Answer: c) Mobile applications

Q4. Which of the following is a cross-platform UI toolkit provided by Xamarin?
a) WinForms
b) WPF (Windows Presentation Foundation)
c) ASP.NET
d) Xamarin.Forms

Answer: d) Xamarin.Forms

Q5. What is the primary advantage of using Xamarin for mobile app development?
a) It allows developers to write separate codebases for iOS and Android.
b) It provides a built-in emulator for testing apps on different devices.
c) It enables developers to use platform-specific languages for app development.
d) It allows sharing a significant portion of code between iOS and Android apps.

Answer: d) It allows sharing a significant portion of code between iOS and Android apps.

Q6. Which of the following is NOT a difference between .NET Framework and .NET Core?
a) .NET Framework includes ASP.NET WebForms, while .NET Core does not.
b) .NET Core is an open-source framework, while .NET Framework is not.
c) .NET Core has a smaller memory footprint compared to .NET Framework.
d) .NET Framework supports cross-platform development, while .NET Core is limited to Windows.

Answer: d) .NET Framework supports cross-platform development, while .NET Core is limited to Windows.

Q7. What is the primary advantage of using Xamarin over other cross-platform mobile development frameworks?
a) It provides better performance compared to native app development.
b) It allows developers to write a single codebase for multiple platforms.

c) It offers a wide range of UI components for web app development.
d) It supports only Android app development.

Answer: b) It allows developers to write a single codebase for multiple platforms.

Q8. Which of the following platforms does Mono NOT support?
a) Windows
b) macOS
c) Linux
d) iOS

Answer: d) iOS

2.2. Versions of the Framework

Q9. Which version of the .NET Framework was the first to be released?
a) .NET Framework 1.0
b) .NET Framework 2.0
c) .NET Framework 3.0
d) .NET Framework 4.0

Answer: a) .NET Framework 1.0

Q10. Which of the following statements is true about the .NET Core versioning scheme?
a) The version number always starts with 1.0.
b) The major version number represents significant updates and changes.
c) The minor version number indicates bug fixes and minor enhancements.
d) The patch version number indicates backward-incompatible changes.

Answer: b) The major version number represents significant updates and changes.

Q11. In which version of the .NET Framework was ASP.NET Core introduced?
a) .NET Framework 2.0
b) .NET Framework 3.5
c) .NET Framework 4.0
d) .NET Framework 5.0

Answer: d) .NET Framework 5.0

Q12. Which version of .NET Core introduced support for Windows Presentation Foundation (WPF) applications?
a) .NET Core 1.0
b) .NET Core 2.0
c) .NET Core 3.0
d) .NET Core 5.0

Answer: c) .NET Core 3.0

Q13. What is the most recent version of .NET Core as of the year 2023?
a) .NET Core 3.1
b) .NET Core 5.0
c) .NET 6.0
d) .NET Core 4.0

Answer: c) .NET 6.0

Q14. In which version of .NET Framework did the Entity Framework feature get introduced?
a) .NET Framework 1.0
b) .NET Framework 2.0
c) .NET Framework 3.0

d) .NET Framework 3.5

Answer: d) .NET Framework 3.5

Q15. Which version of the .NET Framework introduced support for Windows Communication Foundation (WCF)?
a) .NET Framework 2.0
b) .NET Framework 3.0
c) .NET Framework 3.5
d) .NET Framework 4.0

Answer: b) .NET Framework 3.0

2.3. Managed and Unmanaged Code

Q16. What is managed code in the .NET Framework?
a) Code written in any programming language other than C#
b) Code that is executed directly by the CPU without any runtime environment
c) Code that is executed by the Common Language Runtime (CLR)
d) Code that requires manual memory management

Answer: c) Code that is executed by the Common Language Runtime (CLR)

Q17. Which of the following languages can be used to write managed code in the .NET Framework?
a) C++ and Assembly
b) C# and VB.NET
c) JavaScript and Python
d) Ruby and PHP

Answer: b) C# and VB.NET

Q18. What are the advantages of using managed code in .NET applications?
a) Faster execution and better performance
b) Platform independence and automatic memory management
c) Direct interaction with hardware resources
d) Reduced dependency on the Common Language Runtime (CLR)

Answer: b) Platform independence and automatic memory management

Q19. What is unmanaged code in the context of the .NET Framework?
a) Code that is written in an unstructured manner
b) Code that is directly executed by the CPU without any runtime environment
c) Code that relies on the Common Language Runtime (CLR) for execution
d) Code that requires manual memory management

Answer: b) Code that is directly executed by the CPU without any runtime environment

Q20. Which of the following languages is typically associated with unmanaged code in the .NET Framework?
a) C#
b) VB.NET
c) C++
d) Python

Answer: c) C++

Q21. What is the primary advantage of using managed code in

.NET applications?
a) Improved performance compared to unmanaged code

b) Easier debugging and error handling
c) Automatic memory management and garbage collection
d) Ability to directly interact with hardware resources

Answer: c) Automatic memory management and garbage collection

Q22. In the context of the .NET Framework, what is the main difference between managed and unmanaged code in terms of memory management?
a) Managed code relies on the operating system for memory management, while unmanaged code handles memory directly.
b) Managed code requires explicit memory deallocation, while unmanaged code uses automatic garbage collection.
c) Managed code uses reference counting for memory management, while unmanaged code uses heap and stack allocation.
d) Managed code does not require the programmer to handle memory management, while unmanaged code requires manual memory allocation and deallocation.

Answer: d) Managed code does not require the programmer to handle memory management, while unmanaged code requires manual memory allocation and deallocation.

2.4. Introduction to Visual Studio

Q23. What is Visual Studio in the context of .NET development?
a) A web-based code editor
b) A programming language
c) An integrated development environment (IDE)
d) A version control system

Answer: c) An integrated development environment (IDE)

Q24. Which programming languages are commonly supported by Visual Studio?
a) C# and JavaScript
b) Python and Ruby
c) Java and C++
d) PHP and HTML

Answer: a) C# and JavaScript

Q25. What are some of the features provided by Visual Studio for .NET development?
a) Code debugging, code refactoring, and integrated testing tools
b) Graphic design tools for creating user interfaces
c) Built-in web server for hosting web applications
d) Code completion and code hinting for all programming languages

Answer: a) Code debugging, code refactoring, and integrated testing tools

Q26. How does Visual Studio help developers manage their projects?
a) It automatically generates project documentation and reports.
b) It provides version control features to manage code changes.
c) It offers cloud-based storage for project files.
d) It automatically deploys projects to production servers.

Answer: b) It provides version control features to manage code changes.

Q27. Which of the following components are part of Visual Studio IDE?
a) Just-In-Time (JIT) compiler and Common Language Runtime (CLR)
b) .NET Framework and Base Class Library (BCL)
c) Solution Explorer, Code Editor, and Toolbox
d) NuGet Package Manager and IIS Express

Answer: c) Solution Explorer, Code Editor, and Toolbox

Q28. What is the primary function of the Solution Explorer in Visual Studio?
a) To manage the installed NuGet packages
b) To organize and navigate project files and folders
c) To configure the project's properties and settings
d) To preview the application's user interface

Answer: b) To organize and navigate project files and folders

2.5. Using ILDASM

Q29. What does ILDASM stand for in .NET development?
a) Intermediate Language Decompiler and Assembler
b) Intermediate Language Disassembler
c) Integrated Language Debugging and Analysis System
d) Intermediate Language Development and Management Tool

Answer: b) Intermediate Language Disassembler

Q30. What is the purpose of ILDASM in the .NET Framework?
a) To convert IL code to machine code
b) To decompile .NET assemblies into human-readable IL code
c) To generate platform-specific binaries from IL code
d) To optimize IL code for better performance

Answer: b) To decompile .NET assemblies into human-readable IL code

Q31. Which file types can be analyzed using ILDASM?
a) Only .exe files
b) Only .dll files
c) Both .exe and .dll files
d) Only source code files

Answer: c) Both .exe and .dll files

Q32. What is the advantage of using ILDASM for .NET developers?
a) It helps in debugging runtime errors.
b) It allows inspecting the IL code generated from source code.
c) It enables direct conversion of IL code to native machine code.
d) It provides a user-friendly interface for writing IL code.

Answer: b) It allows inspecting the IL code generated from source code.

Q33. Which of the following statements is true about the ILDASM tool?
a) It is used for code optimization during the build process.
b) It allows developers to directly modify the IL code.
c) It can generate source code from a compiled .NET assembly.
d) It is automatically installed with the .NET Framework.

Answer: c) It can generate source code from a compiled .NET assembly.

Q34. How can ILDASM be useful in the development process?
a) It provides a platform for testing and debugging .NET applications.
b) It allows developers to write and compile code directly in IL.
c) It aids in understanding the IL code generated by the compiler.
d) It optimizes the performance of IL code for better execution.

Answer: c) It aids in understanding the IL code generated by the compiler.

Q35. Which of the following actions can be performed using ILDASM?
a) Directly modifying the compiled machine code
b) Converting IL code to platform-specific assembly language
c) Decompiling .NET assemblies into C# or VB.NET source code
d) Executing IL code directly to test application behavior

Answer: c) Decompiling .NET assemblies into C# or VB.NET source code

3. Lecture: C# Basics and Class Libraries

Apologies for the misunderstanding. Here are five multiple-choice questions with answers for each topic and subtopic:

3.1. Console Applications and Class Libraries (Framework and .Net Core)

Q1. What is a class library in C#?
a) A program that runs in the console and interacts with the user.
b) A collection of classes and components that can be reused in multiple projects.
c) A special type of library that can only be used in .NET Core projects.
d) A library that is used to write console applications.

Answer: b) A collection of classes and components that can be reused in multiple projects.

Q2. Which of the following is true about .NET Core?
a) It is a software framework developed by Microsoft for building Windows applications.
b) It is an open-source, cross-platform framework that can be used to build applications for Windows, macOS, and Linux.
c) It can only be used to build web applications.
d) .NET Core applications cannot run on Linux.

Answer: b) It is an open-source, cross-platform framework that can be used to build applications for Windows, macOS, and Linux.

Q3. How do you create a new console application project in Visual Studio?
a) File > New > Project > Console Application
b) File > New > Project > Class Library
c) File > New > Project > Web Application
d) File > New > Project > Windows Forms Application

Answer: a) File > New > Project > Console Application

Q4. What is the purpose of the "Main" method in a console application?
a) It is the first method called when the program starts and serves as the entry point.
b) It is a reserved method that cannot be used in console applications.
c) It is used to print messages to the console.
d) It is used to create an instance of the console application class.

Answer: a) It is the first method called when the program starts and serves as the entry point.

Q5. How do you add a reference to a class library in a console application project?
a) By copying the library DLL file into the project's directory.
b) By using the "using" directive at the beginning of the source code file.
c) By adding a project reference to the class library in the Visual Studio project properties.
d) By creating a new instance of the class library in the console application.

Answer: c) By adding a project reference to the class library in the Visual Studio project properties.

3.2. C# Basics

Q6. Which of the following is NOT a valid C# identifier?
a) myVariable
b) _totalAmount
c) 123abc
d) firstName

Answer: c) 123abc

Q7. What is the correct way to declare a variable in C# with an initial value of 10?
a) int num = 10;
b) var num = 10;
c) number = 10;
d) int num => 10;

Answer: a) int num = 10;

Q8. What does the "var" keyword do in C#?
a) It declares a variable with an unknown data type.
b) It specifies that the variable is read-only.
c) It indicates a variable that can only be accessed within the current block of code.
d) It is used to declare global variables.

Answer: a) It declares a variable with an unknown data type.

Q9. What is the result of the following expression: 10 / 3 ?
a) 3
b) 3.33
c) 3.0
d) 4

Answer: c) 3.0

Q10. What is the purpose of the "if" statement in C#?
a) To declare a loop that will execute a block of code repeatedly.
b) To define a method or function in C#.
c) To conditionally execute a block of code based on a boolean expression.
d) To print messages to the console.

Answer: c) To conditionally execute a block of code based on a boolean expression.

3.3. Project References, using

Q11. In C#, how do you reference an external class library in your project?
a) By copying the library DLL file into the project's directory.
b) By using the "include" keyword followed by the library name in the project file.
c) By adding a project reference to the external library in the Visual Studio project properties.
d) By using the "using" directive at the beginning of the source code file.

Answer: c) By adding a project reference to the external library in the Visual Studio project properties.

Q12. Which of the following is true about using directives in C#?
a) Using directives are used to include external class libraries in the project.
b) Using directives are required for all classes and methods in C#.
c) Using directives must be defined at the beginning of every C# source code file.
d) Using directives are used to specify the namespace where a class is located.

Answer: d) Using directives are used to specify the namespace where a class is located.

Q13. How do you use the "using" directive to include a namespace in your C# code?
a) using MyNamespace;
b) use MyNamespace;
c) include MyNamespace;
d) import MyNamespace;

Answer: a) using MyNamespace;

Q14. What is the purpose of using a project reference in C#?
a) To include external class libraries in the project.

b) To reference a specific class within a namespace.
c) To add additional functionality to the project.
d) To establish a relationship between two projects, allowing them to share code.

Answer: d) To establish a relationship between two projects, allowing them to share code.

Q15. In C#, can a project reference be added to a project located in a different solution?
a) Yes, project references can be added to projects in different solutions.
b) No, project references can only be added to projects in the same solution.
c) Only .NET Core projects allow project references between different solutions.
d) Project references are not supported in C#.

Answer: b) No, project references can only be added to projects in the same solution.

3.4. Classes

Q16. In C#, what is a class?
a) A function that performs a specific task and returns a value.
b) A template or blueprint for creating objects that define their structure and behavior.
c) A variable that holds multiple values.
d) A collection of methods with the same name but different parameters.

Answer: b) A template or blueprint for creating objects that define their structure and behavior.

Q17. How do you define a class in C#?
a) By using the "method" keyword followed by the class name and method body.
b) By using the "class" keyword followed by the class name and class body.
c) By using the "new" keyword followed by the class name and constructor parameters.
d) By using the "var" keyword followed by the class name and variable initialization.

Answer: b) By using the "class" keyword followed by the class name and class body.

Q18. What is the default access modifier for a class in C# if no access modifier is specified?
a) private
b) protected
c) internal
d) public

Answer: c) internal

Q19. What is

the purpose of an object in C#?
a) To define the structure and behavior of a class.
b) To define a template or blueprint for creating classes.
c) To create instances of a class, allowing you to work with its properties and methods.
d) To perform a specific task and return a value.

Answer: c) To create instances of a class, allowing you to work with its properties and methods.

Q20. Which of the following is true about object-oriented programming (OOP) in C#?
a) It is a programming paradigm that does not support the concept of classes and objects.
b) It is a programming paradigm based on the use of objects and their interactions to design and build applications.
c) It is a programming paradigm that focuses on writing linear code without the use of functions or methods.
d) It is a programming paradigm that is only used in .NET Framework, not in .NET Core.

Answer: b) It is a programming paradigm based on the use of objects and their interactions to design and build applications.

3.5. Data Types in .Net and CTS equivalents

Q21. In C#, what is a data type?
a) It is a keyword that indicates the scope of a variable.
b) It is a function that performs a specific task and returns a value.
c) It is a value that represents different types of data, such as integers, strings, and floating-point numbers.
d) It is a collection of methods and properties that defines the behavior of a class.

Answer: c) It is a value that represents different types of data, such as integers, strings, and floating-point numbers.

Q22. How do you declare a variable of type "int" in C#?
a) int number;
b) int = number;
c) var number;
d) number => int;

Answer: a) int number;

Q23. What is the size of the "int" data type in C#?
a) 8 bits
b) 16 bits
c) 32 bits
d) 64 bits

Answer: c) 32 bits

Q24. Which of the following is a valid C# data type for storing decimal numbers?
a) float
b) double
c) decimal
d) real

Answer: c) decimal

Q25. In C#, what is the equivalent data type of "System.Int32" in the Common Type System (CTS)?
a) int
b) short
c) long
d) byte

Answer: a) int

3.6. Methods

Q26. What is a method in C#?
a) A special type of data type that represents a collection of values.
b) A function that performs a specific task and may return a value.
c) A reserved keyword that cannot be used as an identifier.
d) An instance of a class created from another class.

Answer: b) A function that performs a specific task and may return a value.

Q27. How do you define a method in C#?
a) By using the "method" keyword followed by the method name and method body.
b) By using the "func" keyword followed by the method name and method body.
c) By using the "void" keyword followed by the method name and method body.
d) By using the "new" keyword followed by the method name and method body.

Answer: c) By using the "void" keyword followed by the method name and method body.

Q28. What is the purpose of the "return" statement in a method?
a) To specify the method's return type.
b) To indicate that the method is complete and should terminate its execution.
c) To provide a value to the caller of the method.
d) The "return" statement is not used in C# methods.

Answer: c) To provide a value to the caller of the method.

Q29. What is the return type of a method that does not return any value in C#?
a) int
b) void
c) bool
d) string

Answer: b) void

Q30. How do you call a method in C#?
a) By using the "invoke" keyword followed by the method name and arguments.
b) By using the method name followed by parentheses and arguments.
c) By using the "call" keyword followed by the method name and arguments.
d) By using the "run" keyword followed by the method name and arguments.

Answer: b) By using the method name followed by parentheses and arguments.

3.6.1. Method Overloading

Q31. What is method overloading in C#?
a) A way to override a base class method with a new implementation in the derived class.
b) A way to provide multiple implementations of a method with the same name but different parameters.
c) A way to hide a base class method and prevent it from being accessed in the derived class.
d) A way to create multiple methods with the same name but different return types.

Answer: b) A way to provide multiple implementations of a method with the same name but different parameters.

Q32. How is method overloading achieved in C#?
a) By using the "override" keyword in the method's signature.
b) By using the "virtual" keyword in the method's signature.
c) By defining multiple methods with the same name but different parameter lists in the same class.
d) By defining multiple methods with the same name and parameter list in different classes.

Answer: c) By defining multiple methods with the same name but different parameter lists in the same class.

Q33. Which of the following is NOT a valid way to overload a method in C#?
a) By changing the return type of the method.
b) By changing the number of parameters in the method.
c) By changing the order of parameters in the method.
d) By changing the data type of parameters in the method.

Answer: a) By changing the return type of the method.

Q34. What is the benefit of method overloading in C#?
a) It allows you to hide a base class method and prevent it from being accessed in the derived class.
b) It allows you to define multiple methods with the same name, making the code more concise.
c) It allows you to provide multiple ways of calling the same method with different sets of parameters.

d) Method overloading has no benefit; it should be avoided in C#.

Answer: c) It allows you to provide multiple ways of calling the same method with different sets of parameters.

Q35. In C#, can you overload a method based on the return type only?
a) Yes, you can overload a method based on the return type only.
b) No, method overloading is not allowed in C# based on the return type.
c) Method overloading is not supported in C#.
d) Method overloading based on the return type depends on the version of C# being used.

Answer: b) No, method overloading is not allowed in C# based on the return type.

3.6.2. Optional Parameters

Q36. What are optional parameters in C#?
a) Parameters that are passed by reference to a method.
b) Parameters that are not required to be passed to a method when it is called.
c) Parameters that are required to be passed to a method when it is called.
d) Parameters that are declared as static in the

method signature.

Answer: b) Parameters that are not required to be passed to a method when it is called.

Q37. How do you define an optional parameter in C#?
a) By using the "optional" keyword in the parameter declaration.
b) By using the "default" keyword in the parameter declaration.
c) By using the "optional" keyword followed by the default value in the parameter declaration.
d) By using the "default" keyword followed by the optional value in the parameter declaration.

Answer: c) By using the "optional" keyword followed by the default value in the parameter declaration.

Q38. What is the purpose of optional parameters in C#?
a) To allow the method to be called with different numbers of arguments.
b) To specify that a method cannot be called without passing values for all parameters.
c) To allow the method to be called without passing values for certain parameters.
d) To define parameters that are constant and cannot be modified.

Answer: c) To allow the method to be called without passing values for certain parameters.

Q39. Which of the following is true about optional parameters in C#?
a) All parameters in a method can be declared as optional.
b) Only value-type parameters can be declared as optional.
c) Optional parameters must always be listed first in the method signature.
d) Optional parameters must have default values.

Answer: d) Optional parameters must have default values.

Q40. How do you call a method with optional parameters in C#?
a) By omitting the optional parameters in the method call.
b) By using the "optional" keyword followed by the parameter name and value in the method call.
c) By using the "default" keyword followed by the parameter name and value in the method call.
d) By using the "omit" keyword followed by the parameter name in the method call.

Answer: a) By omitting the optional parameters in the method call.

3.6.3. Named Parameters and Positional Parameters

Q41. What are named parameters in C#?

a) Parameters that are passed by reference to a method.
b) Parameters that are passed by position to a method.
c) Parameters that are explicitly identified by their names in a method call.
d) Parameters that are declared with specific names in the method signature.

Answer: c) Parameters that are explicitly identified by their names in a method call.

Q42. How do you use named parameters in C#?
a) By using the parameter name followed by the value in the method call.
b) By using the parameter name followed by the colon (:) and value in the method call.
c) By using the parameter name followed by the equal sign (=) and value in the method call.
d) By using the parameter name followed by the hyphen (-) and value in the method call.

Answer: c) By using the parameter name followed by the equal sign (=) and value in the method call.

Q43. What is the benefit of using named parameters in C#?
a) Named parameters provide better performance compared to positional parameters.
b) Named parameters allow you to pass arguments to a method in any order, making the code more flexible.
c) Named parameters reduce the number of required parameters in a method signature.
d) Named parameters are more memory-efficient than positional parameters.

Answer: b) Named parameters allow you to pass arguments to a method in any order, making the code more flexible.

Q44. Which of the following is true about using named parameters in C#?
a) All parameters in a method must be passed using named parameters.
b) Named parameters must be listed first in the method call.
c) Named parameters must be declared with specific names in the method signature.
d) Named parameters can be mixed with positional parameters in a method call.

Answer: d) Named parameters can be mixed with positional parameters in a method call.

Q45. How do you use positional parameters in C#?
a) By specifying the parameter values in the method call without identifying their names.
b) By using the parameter names followed by the values in the method call.
c) Positional parameters are not supported in C#.
d) By using the "positional" keyword followed by the parameter names in the method call.

Answer: a) By specifying the parameter values in the method call without identifying their names.

3.6.4. Using params

Q46. What is the purpose of the "params" keyword in C#?
a) To define parameters that are passed by position to a method.
b) To define parameters that are passed by reference to a method.
c) To specify that a method can accept a variable number of arguments of the same data type.
d) To specify that a method can accept a variable number of arguments of different data types.

Answer: c) To specify that a method can accept a variable number of arguments of the same data type.

Q47. How do you use the "params" keyword in C# method declaration?
a) By using the "params" keyword followed by the parameter name in the method signature.
b) By using the "params" keyword followed by the parameter data type and an array parameter name in the method signature.
c) By using the "params" keyword followed by the method name and an array parameter name in the method signature.
d) By using the "params" keyword followed by the method name and an array parameter name in the method body.

Answer: b) By using the "params" keyword followed by the parameter data type and an array parameter name in the method signature.

Q48. What is the benefit of using the "params" keyword in C#?
a) It allows you to pass an unlimited number of arguments to a method.
b) It allows you to pass arguments to a method in any order, making the code more flexible.
c) It reduces the number of required parameters in a method signature.
d) It improves the performance of the method by reducing memory usage.

Answer: a) It allows you to pass an unlimited number of arguments to a method.

Q49. Which of the following is true about using the "params" keyword in C#?
a) Only value-type parameters can be defined using the "params" keyword.
b) The "params" keyword can be used with optional parameters.
c) The "params" keyword can be used with named parameters.
d) The "params" keyword can only be used with arrays.

Answer: d) The "params" keyword can only be used with arrays.

Q50. How do you call a method with a "params" parameter in C#?
a) By using the parameter name followed by the value in the method call.
b) By using the "params" keyword followed by the parameter name and value in the method call.
c) By using the method name followed by parentheses and passing the arguments directly in the method call.
d) By using the "run" keyword followed by the method name and arguments.

Answer: c) By using the method name followed by parentheses and passing the arguments directly in the method call.

3.6.5. Local functions

Q51. What is a local function in C#?
a) A function that is declared within another method and can only be accessed within that method.
b) A function that is declared in a different class from where it is used.
c) A function that is declared as static and can be accessed without creating an instance of the class.
d) A function that

is declared with the "local" keyword and cannot be accessed from outside the current namespace.

Answer: a) A function that is declared within another method and can only be accessed within that method.

Q52. How do you declare a local function in C#?
a) By using the "local" keyword followed by the function name and function body.
b) By using the "func" keyword followed by the function name and function body.
c) By using the "void" keyword followed by the function name and function body.
d) By using the "new" keyword followed by the function name and function body.

Answer: a) By using the "local" keyword followed by the function name and function body.

Q53. What is the benefit of using local functions in C#?
a) Local functions allow you to create reusable code that can be called from other methods.
b) Local functions improve the performance of the program by reducing memory usage.
c) Local functions provide better encapsulation and organization of code within a method.
d) Local functions are required in C# and should be used in every method.

Answer: c) Local functions provide better encapsulation and organization of code within a method.

Q54. Which of the following is true about using local functions in C#?
a) Local functions cannot have parameters.

b) Local functions cannot have a return type.
c) Local functions can only be declared in the main method.
d) Local functions can access variables declared in the outer method.

Answer: d) Local functions can access variables declared in the outer method.

Q55. When should you use local functions in C#?
a) Local functions should be used for every method to improve performance.
b) Local functions should be used when a specific piece of code needs to be reused in multiple methods.
c) Local functions should be used for all methods that have parameters.
d) Local functions should be avoided in C# because they are not supported in .NET Core.

Answer: b) Local functions should be used when a specific piece of code needs to be reused in multiple methods.

3.7. Properties

Q56. What is a property in C#?
a) A special type of data type that represents a collection of values.
b) A function that performs a specific task and returns a value.
c) A value that represents different types of data, such as integers, strings, and floating-point numbers.
d) A class member that provides access to the internal state of an object.

Answer: d) A class member that provides access to the internal state of an object.

Q57. How do you define a property in C#?
a) By using the "property" keyword followed by the property name and get/set accessors.
b) By using the "property" keyword followed by the property name and the property value.
c) By using the "get" and "set" keywords followed by the property name and the property value.
d) By using the "get" and "set" keywords followed by the property name and the get/set accessors.

Answer: d) By using the "get" and "set" keywords followed by the property name and the get/set accessors.

Q58. What is the purpose of the "get" accessor in a property?
a) To set the value of the property.
b) To get the value of the property.
c) The "get" accessor is not used in properties.
d) To define the data type of the property.

Answer: b) To get the value of the property.

Q59. What is the purpose of the "set" accessor in a property?
a) To get the value of the property.
b) To define the data type of the property.
c) To set the value of the property.
d) The "set" accessor is not used in properties.

Answer: c) To set the value of the property.

Q60. How do you access a property in C#?
a) By using the property name followed by parentheses and arguments.
b) By using the property name followed by the "get" or "set" keyword and parentheses.
c) By using the property name followed by the "get" or "set" keyword and a dot (.) and the property value.
d) By using the "access" keyword followed by the property name.

Answer: b) By using the property name followed by the "get" or "set" keyword and parentheses.

3.7.1. get, set

Q61. What are the "get" and "set" accessors in a property?
a) The "get" accessor is used to set the value of the property, and the "set" accessor is used to get the value of the property.
b) The "get" accessor is used to get the value of the property, and the "set" accessor is used to set the value of the property.
c) Both "get" and "set" accessors are used to set the value of the property.
d) Both "get" and "set" accessors are used to get the value of the property.

Answer: b) The "get" accessor is used to get the value of the property, and the "set" accessor is used to set the value of the property.

Q62. How do you define the "get" and "set" accessors for a property in C#?
a) By using the "get" and "set" keywords followed by the property name and the property value.
b) By using the "get" and "set" keywords followed by the property name and the get/set accessors.
c) By using the "property" keyword followed by the property name and get/set accessors.
d) By using the "property" keyword followed by the property name and the property value.

Answer: b) By using the "get" and "set" keywords followed by the property name and the get/set accessors.

Q63. What is the purpose of the "get" accessor in a property?
a) To set the value of the property.
b) To get the value of the property.
c) The "get" accessor is not used in properties.
d) To define the data type of the property.

Answer: b) To get the value of the property.

Q64. What is the purpose of the "set" accessor in a property?
a) To get the value of the property.
b) To define the data type of the property.
c) To set the value of the property.
d) The "set" accessor is not used in properties.

Answer: c) To set the value of the property.

Q65. How do you access the "get" and "set" accessors of a property in C#?
a) By using the property name followed by parentheses and arguments.
b) By using the property name followed by the "get" or "set" keyword and parentheses.
c) By using the property name followed by the "get" or "set" keyword and a dot (.) and the property value.
d) By using the "access" keyword followed by the property name.

Answer: b) By using the property name followed by the "get" or "set" keyword and parentheses.

3.7.2. Readonly properties

Q66. What is a readonly property in C#?
a) A property that can only be accessed from within the class it is defined in

.
b) A property that can only be set once, typically during object initialization, and cannot be changed afterward.
c) A property that cannot be used in C#.
d) A property that is defined with the "readonly" keyword.

Answer: b) A property that can only be set once, typically during object initialization, and cannot be changed afterward.

Q67. How do you define a readonly property in C#?
a) By using the "readonly" keyword followed by the property name and the property value.
b) By using the "readonly" keyword followed by the property name and the get accessor.
c) By using the "get" and "set" keywords followed by the property name and the readonly keyword.
d) By using the "get" and "set" keywords followed by the property name and the readonly keyword.

Answer: d) By using the "get" and "set" keywords followed by the property name and the readonly keyword.

Q68. What is the purpose of a readonly property in C#?
a) To prevent other classes from accessing the property.
b) To define a property that can only be accessed from within the class it is defined in.
c) To define a property that can be set multiple times during the object's lifetime.
d) To create a property that can only be set once, typically during object initialization.

Answer: d) To create a property that can only be set once, typically during object initialization.

Q69. When should you use a readonly property in C#?
a) Readonly properties should be used for all properties to improve performance.
b) Readonly properties should be used when you want to allow the property value to be changed after object initialization.
c) Readonly properties should be used when you want to create a property that can only be set once.
d) Readonly properties should be used for all properties to prevent other classes from accessing the property.

Answer: c) Readonly properties should be used when you want to create a property that can only be set once.

Q70. How do you access the value of a readonly property in C#?
a) By using the property name followed by parentheses and arguments.
b) By using the property name followed by the "get" keyword and parentheses.
c) By using the property name followed by the "set" keyword and parentheses.
d) By using the property name followed by a dot (.) and the property value.

Answer: b) By using the property name followed by the "get" keyword and parentheses.

3.7.3. Using property accessors to create Readonly property

Q71. How can you create a readonly property in C# using property accessors?
a) By using the "readonly" keyword followed by the property name and the property value.
b) By using the "get" accessor only and not defining the "set" accessor for the property.
c) By using the "get" and "set" accessors, both with the "readonly" keyword.
d) By using the "get" accessor only and defining the "set" accessor with the "readonly" keyword.

Answer: b) By using the "get" accessor only and not defining the "set" accessor for the property.

Q72. What is the purpose of using property accessors to create a readonly property in C#?
a) Property accessors provide a way to define a property that can only be accessed from within the class it is defined in.
b) Property accessors allow you to set multiple values for a property during the object's lifetime.
c) Property accessors provide a way to create a property that can only be set once, typically during object initialization.
d) Property accessors are used to specify the data type of a property.

Answer: c) Property accessors provide a way to create a property that can only be set once, typically during object initialization.

Q73. How do you define the "get" accessor to create a readonly property in C#?
a) By using the "get" keyword followed by the property name and the property value.
b) By using the "get" keyword followed by the property name and a semicolon (;).

c) By using the "get" keyword followed by the property name and the "readonly" keyword.
d) By using the "get" keyword followed by the property name and the "get" keyword again.

Answer: b) By using the "get" keyword followed by the property name and a semicolon (;).

Q74. What is the purpose of the "get" accessor in a readonly property?
a) To get the value of the property.
b) To set the value of the property.
c) The "get" accessor is not used in readonly properties.
d) To define the data type of the property.

Answer: a) To get the value of the property.

Q75. How do you access the value of a readonly property in C#?
a) By using the property name followed by parentheses and arguments.
b) By using the property name followed by the "get" keyword and parentheses.
c) By using the property name followed by the "set" keyword and parentheses.
d) By using the property name followed by a dot (.) and the property value.

Answer: b) By using the property name followed by the "get" keyword and parentheses.

3.8. Constructors

Q76. What is a constructor in C#?
a) A function that performs a specific task and returns a value.
b) A special method that is called when an object is created from a class.
c) A method that is used to set the value of a property.
d) A keyword used to declare a variable in a class.

Answer: b) A special method that is called when an object is created from a class.

Q77. How do you define a constructor in C#?
a) By using

the "constructor" keyword followed by the constructor name and constructor body.
b) By using the "func" keyword followed by the constructor name and constructor body.
c) By using the "void" keyword followed by the constructor name and constructor body.
d) By using the class name followed by parentheses and constructor body.

Answer: d) By using the class name followed by parentheses and constructor body.

Q78. What is the purpose of a constructor in C#?
a) Constructors are used to create instances of a class and allocate memory for the object.
b) Constructors are used to set the value of properties in a class.
c) Constructors are used to define methods in a class.
d) Constructors are used to declare variables in a class.

Answer: a) Constructors are used to create instances of a class and allocate memory for the object.

Q79. Which of the following is true about constructors in C#?
a) A class can have multiple constructors with the same name but different parameters.
b) Constructors cannot take any parameters; they must be parameterless.
c) Constructors cannot be overloaded.
d) Constructors cannot be used in C#.

Answer: a) A class can have multiple constructors with the same name but different parameters.

Q80. How do you call a constructor in C#?
a) By using the "invoke" keyword followed by the constructor name and arguments.
b) By using the constructor name followed by parentheses and arguments.

c) By using the "call" keyword followed by the constructor name and arguments.
d) By using the "run" keyword followed by the constructor name and arguments.

Answer: b) By using the constructor name followed by parentheses and arguments.

3.9. Object Initializer

Q81. What is an object initializer in C#?
a) A method used to initialize objects in a class.
b) A special type of data type that represents a collection of values.
c) A way to create an object and set its properties or fields in a single statement.
d) A function that performs a specific task and returns a value.

Answer: c) A way to create an object and set its properties or fields in a single statement.

Q82. How do you use an object initializer in C#?
a) By using the "initialize" keyword followed by the object name and property assignments.
b) By using the object name followed by curly braces ({}) and property assignments.
c) By using the object name followed by parentheses and property assignments.
d) By using the "initialize" keyword followed by parentheses and property assignments.

Answer: b) By using the object name followed by curly braces ({}) and property assignments.

Q83. What is the benefit of using object initializers in C#?
a) Object initializers improve the performance of the program by reducing memory usage.
b) Object initializers allow you to create objects without using constructors.
c) Object initializers allow you to create and initialize objects in a single statement, which makes the code more concise.
d) Object initializers provide better encapsulation and organization of code within a class.

Answer: c) Object initializers allow you to create and initialize objects in a single statement, which makes the code more concise.

Q84. Which of the following is true about using object initializers in C#?
a) Object initializers can only be used with value types.
b) Object initializers can only be used with reference types.
c) Object initializers can be used with both value types and reference types.
d) Object initializers cannot be used in C#.

Answer: c) Object initializers can be used with both value types and reference types.

Q85. How do you use an object initializer to set properties of an object in C#?
a) By using the property name followed by the assignment operator (=) and the property value in curly braces ({}) after the object name.
b) By using the property name followed by the assignment operator (=) and the property value in parentheses after the object name.
c) By using the property name followed by the assignment operator (=) and the property value in parentheses after the object name.
d) By using the property name followed by the assignment operator (=) and the property value after the object name.

Answer: a) By using the property name followed by the assignment operator (=) and the property value in curly braces ({}) after the object name.

3.10. Destructors

Q86. What is a destructor in C#?
a) A method used to destroy objects and free up memory.
b) A special method that is called when an object is created from a class.
c) A way to release resources and perform cleanup operations before an object is destroyed.

d) A keyword used to declare a variable in a class.

Answer: c) A way to release resources and perform cleanup operations before an object is destroyed.

Q87. How do you define a destructor in C#?
a) By using the "destructor" keyword followed by the destructor name and destructor body.
b) By using the "void" keyword followed by the destructor name and destructor body.
c) By using the "~" symbol followed by the class name and destructor body.
d) By using the "new" keyword followed by the class name and destructor body.

Answer: c) By using the "~" symbol followed by the class name and destructor body.

Q88. What is the purpose of a destructor in C#?
a) Destructors are used to create instances of a class and allocate memory for the object.
b) Destructors are used to set the value of properties in a class.
c) Destructors are used to define methods in a class.
d) Destructors are used to release resources and perform cleanup operations before an object is destroyed.

Answer: d) Destructors are used to release resources and perform cleanup operations before an object is destroyed.

Q89. Which of the following is true about destructors in C#?
a) A class can have multiple destructors with different names.
b) Destructors cannot be used in C#.
c) Destructors cannot be overloaded.
d) Destructors are automatically called when an object is created from a class.

Answer: b) Destructors cannot be used in C#.

Q90. How do you release resources and perform cleanup operations in C# if destructors are not available?
a) By using the "release" keyword followed by the resource name in the class.
b) By using the "dispose" keyword followed by the resource name in the class.
c) By implementing the "IDisposable" interface and using the "Dispose" method.
d) By using the "cleanup" keyword followed by the resource name in the class.

Answer: c) By implementing the "IDisposable" interface and using the "Dispose" method.

4. Lecture: Advanced Class Concepts

4.1. Static Members of a Class

Q1. What are static members of a class in C#?
a) Members that are accessible only within the class where they are defined.
b) Members that can be accessed without creating an instance of the class.
c) Members that are declared using the "static" keyword and have a fixed value.
d) Members that are used to create objects from the class.

Answer: b) Members that can be accessed without creating an instance of the class.

Q2. Which of the following can be declared as static members in a C# class?
a) Fields
b) Properties
c) Methods
d) All of the above

Answer: d) All of the above

Q3. How do you declare a static field in C#?
a) By using the "field" keyword followed by the field name and data type.
b) By using the "static" keyword followed by the field name and data type.
c) By using the "var" keyword followed by the field name and data type.
d) By using the "const" keyword followed by the field name and data type.

Answer: b) By using the "static" keyword followed by the field name and data type.

Q4. What is the main difference between a static method and an instance method in C#?
a) Static methods can be called without creating an instance of the class, while instance methods require an instance of the class to be called.
b) Instance methods can be called without creating an instance of the class, while static methods require an instance of the class to be called.
c) Static methods are declared with the "static" keyword, while instance methods are declared with the "instance" keyword.
d) Static methods can only be used with value types, while instance methods can only be used with reference types.

Answer: a) Static methods can be called without creating an instance of the class, while instance methods require an instance of the class to be called.

Q5. How do you access a static member of a class in C#?
a) By using the member name followed by parentheses and arguments.
b) By using the member name followed by the "static" keyword and parentheses.
c) By using the class name followed by a dot (.) and the member name.
d) By using the class name followed by a semicolon (;) and the member name.

Answer: c) By using the class name followed by a dot (.) and the member name.

4.1.1. Fields

Q6. What are fields in C#?
a) Fields are methods in a class that perform specific tasks and return values.
b) Fields are special types of data that represent a collection of values.
c) Fields are variables in a class used to store data that represents the state of an object.
d) Fields are classes that contain static members.

Answer: c) Fields are variables in a class used to store data that represents the state of an object.

Q7. How do you declare a field in C#?
a) By using the "field" keyword followed by the field name and data type.
b) By using the "var" keyword followed by the field name and data type.
c) By using the "field" keyword followed by the field name and the "field" keyword again.
d) By using the data type followed by the field name and a semicolon (;).

Answer: d) By using the data type followed by the field name and a semicolon (;).

Q8. Which of the following access specifiers can be used with fields in C#?
a) public
b) private
c) protected
d) All of the above

Answer: d) All of the above

Q9. What is the default value of a numeric field declared in a class if not initialized explicitly?
a) 0
b) 1
c) null
d) The default value depends on the data type of the field.

Answer: a) 0

Q10. How do you access the value of a field in C#?
a) By using the field name followed by parentheses and arguments.
b) By using the field name followed by the "get" keyword and parentheses.
c) By using the field name followed by the "set" keyword and parentheses.
d) By using the field name followed by a dot (.) and the field value.

Answer: d) By using the field name followed by a dot (.) and the field value.

4.1.2. Methods

Q11. What are methods in C#?
a) Methods are special types of data that represent a collection of values.
b) Methods are variables in a class used to store data that represents the state of an object.
c) Methods are classes that contain static members.
d) Methods are functions in a class that perform specific tasks and return values.

Answer: d) Methods are functions in a class that perform specific tasks and return values.

Q12. How do you declare a method in C#?
a) By using the "method" keyword followed by the method name and method body.
b) By using the "void" keyword followed by the method name and method body.
c) By using the "func" keyword followed by the method name and method body.
d) By using the method name followed by parentheses and method body.

Answer: b) By using the "void" keyword followed by the method name and method body.

Q13. Which of the following access specifiers can be used with methods in C#?
a) public
b) private
c) protected
d) All of the above

Answer: d) All of the above

Q14. What is the purpose of the "void" keyword in a method declaration?

a) The "void" keyword specifies that the method has a return type of "void," meaning it does not return any value.
b) The "void" keyword specifies that the method is a static method.
c) The "void" keyword specifies that the method is an instance method.
d) The "void" keyword specifies that the method can be accessed from outside the class.

Answer: a) The "void" keyword specifies that the method has a return type of "void," meaning it does not return any value.

Q15. How do you call a method in C#?
a) By using the method name followed by parentheses and arguments.
b) By using the method name followed by the "invoke" keyword and parentheses.
c) By using the method name followed by the "call" keyword and parentheses.
d) By using the method name followed by a semicolon (;) and parentheses.

Answer: a) By using the method name followed by parentheses and arguments.

4.1.3. Properties

Q16. What are properties in C#?
a) Properties are variables in a class used to store data that represents the state of an object.
b) Properties are methods in a class that perform specific tasks and return values.
c) Properties are special types of data that represent a collection of values.
d) Properties are class members used to provide access to the internal state of an object.

Answer: d) Properties are class members used to provide access to the internal state of an object.

Q17. How do you declare a property in C#?
a) By using the "property" keyword

followed by the property name and property type.
b) By using the "var" keyword followed by the property name and property type.
c) By using the "get" and "set" keywords followed by the property name and property type.
d) By using the property name followed by parentheses and property type.

Answer: c) By using the "get" and "set" keywords followed by the property name and property type.

Q18. Which of the following access specifiers can be used with properties in C#?
a) public
b) private
c) protected
d) All of the above

Answer: d) All of the above

Q19. What is the purpose of the "get" accessor in a property?
a) The "get" accessor is used to set the value of the property.
b) The "get" accessor is used to define the data type of the property.
c) The "get" accessor is used to get the value of the property.
d) The "get" accessor is not used in properties.

Answer: c) The "get" accessor is used to get the value of the property.

Q20. How do you access the value of a property in C#?
a) By using the property name followed by parentheses and arguments.
b) By using the property name followed by the "get" keyword and parentheses.
c) By using the property name followed by the "set" keyword and parentheses.
d) By using the property name followed by a dot (.) and the property value.

Answer: b) By using the property name followed by the "get" keyword and parentheses.

4.1.4. Constructors

Q21. What are constructors in C#?
a) Constructors are special types of data that represent a collection of values.
b) Constructors are classes that contain static members.
c) Constructors are methods in a class used to perform specific tasks and return values.
d) Constructors are special methods that are called when an object is created from a class.

Answer: d) Constructors are special methods that are called when an object is created from a class.

Q22. How do you declare a constructor in C#?
a) By using the "constructor" keyword followed by the constructor name and constructor body.
b) By using the "void" keyword followed by the constructor name and constructor body.
c) By using the class name followed by parentheses and constructor body.
d) By using the class name followed by curly braces ({}) and constructor body.

Answer: c) By using the class name followed by parentheses and constructor body.

Q23. What is the purpose of a constructor in C#?
a) Constructors are used to create instances of a class and allocate memory for the object.
b) Constructors are used to set the value of properties in a class.
c) Constructors are used to define methods in a class.
d) Constructors are used to declare variables in a class.

Answer: a) Constructors are used to create instances of a class and allocate memory for the object.

Q24. Which of the following is true about constructors in C#?
a) A class can have multiple constructors with the same name but different parameters.
b) Constructors cannot take any parameters; they must be parameterless.
c) Constructors cannot be overloaded.
d) Constructors cannot be used in C#.

Answer: a) A class can have multiple constructors with the same name but different parameters.

Q25. How do you call a constructor in C#?
a) By using the constructor name followed by parentheses and arguments.
b) By using the constructor name followed by the "invoke" keyword and parentheses.
c) By using the constructor name followed by the "call" keyword and parentheses.
d) By using the constructor name followed by a semicolon (;) and parentheses.

Answer: a) By using the constructor name followed by parentheses and arguments.

4.2. Static Classes

Q26. What is a static class in C#?
a) A static class is a class that can be instantiated to create objects.
b) A static class is a class that can only contain static members and cannot be instantiated.
c) A static class is a class that can only contain non-static members and can be instantiated.
d) A static class is a class that can only be accessed from within the class it is defined in.

Answer: b) A static class is a class that can only contain static members and cannot be instantiated.

Q27. How do you declare a static class in C#?
a) By using the "static" keyword followed by the class name and class body.
b) By using the "class" keyword followed by the class name and class body.
c) By using the "void" keyword followed by the class name and class body.
d) By using the "new" keyword followed by the class name and class body.

Answer: a) By using the "static" keyword followed by the class name and class body.

Q28. What is the main advantage of using a static class in C#?
a) Static classes allow you to create multiple instances of the class.
b) Static classes provide better encapsulation and organization of code within a class.
c) Static classes cannot be used in C#.
d) Static classes allow you to define and access static members without creating an instance of the class.

Answer: d) Static classes allow you to define and access static members without creating an instance of the class.

Q29. Which of the following is true about static classes in C#?
a) Static classes can have instance members along with static members.
b) Static classes cannot have any members; they must be empty.
c) Static classes can be inherited by other classes.
d) Static classes can only contain properties but not methods.

Answer: b) Static classes cannot have any members; they must be empty.

Q30. How do you access a static member of a static class in C#?
a) By using the member name followed by parentheses and arguments.
b) By using the member name followed by the "static" keyword and parentheses.
c) By using the class name followed by a dot (.) and the member name.
d) By using the class name followed by a semicolon (;) and the member name.

Answer: c) By using the class name followed by a dot (.) and the member name.

4.3. Static local functions

Q31. What are static local functions in C#?
a) Static local functions are functions declared within a static class.
b) Static local functions are functions declared within another method or function.
c) Static local functions are functions that can be called without creating an instance of the class.
d) Static local functions are functions that can only be called from within the class they are declared in.

Answer: b) Static local functions are functions declared within another method or function.

Q32. How do you declare a static local function in C#?
a) By using the "function" keyword followed by the function name and function body.
b) By using the "static" keyword followed by the function name and function body.
c) By using the "void" keyword followed by the function name and function body.
d) By using the function name followed by parentheses and function body.

Answer: d) By using the function name followed by parentheses and function body.

Q33. What is the main advantage of using static local functions in C#?
a) Static local functions allow you to define and access static members without creating

an instance of the class.
b) Static local functions provide better encapsulation and organization of code within a method or function.
c) Static local functions can only be called from within the class they are declared in.
d) Static local functions cannot be used in C#.

Answer: b) Static local functions provide better encapsulation and organization of code within a method or function.

Q34. Which of the following is true about static local functions in C#?
a) Static local functions can have access specifiers (public, private, etc.).
b) Static local functions cannot have any parameters.
c) Static local functions can be declared in a static class only.

d) Static local functions can only be called from outside the class they are declared in.

Answer: b) Static local functions cannot have any parameters.

Q35. How do you call a static local function in C#?
a) By using the function name followed by parentheses and arguments.
b) By using the function name followed by the "invoke" keyword and parentheses.
c) By using the function name followed by the "call" keyword and parentheses.
d) By using the function name followed by a semicolon (;) and parentheses.

Answer: a) By using the function name followed by parentheses and arguments.

4.4. Inheritance

Q36. What is inheritance in C#?
a) Inheritance is a process where one class inherits the properties and behaviors of another class.
b) Inheritance is a process where one class is copied into another class.
c) Inheritance is a process where one class is replaced with another class.
d) Inheritance is a process where one class is hidden from another class.

Answer: a) Inheritance is a process where one class inherits the properties and behaviors of another class.

Q37. How do you declare inheritance in C#?
a) By using the "inherit" keyword followed by the base class name and the derived class name.
b) By using the "extend" keyword followed by the base class name and the derived class name.
c) By using the "inherits" keyword followed by the base class name and the derived class name.
d) By using the derived class name followed by a colon (:) and the base class name.

Answer: d) By using the derived class name followed by a colon (:) and the base class name.

Q38. What is a base class in C#?
a) The base class is the class that inherits properties and behaviors from another class.
b) The base class is the class that is derived from another class.
c) The base class is the class that is hidden from another class.
d) The base class is the class that contains static members.

Answer: b) The base class is the class that is derived from another class.

Q39. What is a derived class in C#?
a) The derived class is the class that inherits properties and behaviors from another class.
b) The derived class is the class that is hidden from another class.
c) The derived class is the class that contains static members.
d) The derived class is the class that is derived from another class.

Answer: d) The derived class is the class that is derived from another class.

Q40. What is the main advantage of using inheritance in C#?
a) Inheritance allows you to define and access static members without creating an instance of the class.
b) Inheritance provides better encapsulation and organization of code within a class.
c) Inheritance allows you to reuse code by inheriting properties and behaviors from a base class.
d) Inheritance allows you to create multiple instances of the class.

Answer: c) Inheritance allows you to reuse code by inheriting properties and behaviors from a base class.

4.4.1. Access Specifiers

Q41. What are access specifiers in C#?
a) Access specifiers are keywords used to define the visibility and accessibility of class members.

b) Access specifiers are used to declare properties in a class.
c) Access specifiers are used to define the data type of variables in a class.
d) Access specifiers are used to define methods in a class.

Answer: a) Access specifiers are keywords used to define the visibility and accessibility of class members.

Q42. How do you specify the access specifier for a class member in C#?
a) By using the "access" keyword followed by the access specifier and the member name.
b) By using the "specify" keyword followed by the access specifier and the member name.
c) By using the access specifier followed by the member name and a semicolon (;).
d) By using the member name followed by the access specifier and a semicolon (;).

Answer: c) By using the access specifier followed by the member name and a semicolon (;).

Q43. Which of the following access specifiers provides the widest accessibility to a class member in C#?
a) public
b) private
c) protected
d) internal

Answer: a) public

Q44. What is the default access specifier for class members in C# if not explicitly specified?
a) public
b) private
c) protected
d) internal

Answer: b) private

Q45. Which of the following access specifiers restricts the visibility of a class member to the same class and its derived classes?
a) public
b) private
c) protected
d) internal

Answer: c) protected

4.4.2. Constructors in a hierarchy

Q46. What happens when a derived class is created in C#?
a) The derived class inherits all the constructors from the base class.
b) The derived class hides all the constructors from the base class.
c) The derived class automatically creates its own constructors.
d) The derived class cannot have constructors.

Answer: a) The derived class inherits all the constructors from the base class.

Q47. Can a derived class have its own constructors in C#?
a) Yes, a derived class can have its own constructors in addition to the inherited constructors.
b) No, a derived class cannot have its own constructors; it can only use the inherited constructors.
c) Yes, a derived class can have its own constructors, but it must override the inherited constructors.
d) No, a derived class can only use the constructors defined in the base class.

Answer: a) Yes, a derived class can have its own

constructors in addition to the inherited constructors.

Q48. What is the purpose of a constructor in a derived class?
a) Constructors in a derived class are used to hide the constructors from the base class.
b) Constructors in a derived class are used to override the constructors from the base class.
c) Constructors in a derived class are used to define the data type of variables.
d) Constructors in a derived class are used to create instances of the derived class.

Answer: b) Constructors in a derived class are used to override the constructors from the base class.

Q49. How do you call the constructor of the base class from a derived class in C#?
a) By using the "base" keyword followed by parentheses and arguments.
b) By using the "this" keyword followed by parentheses and arguments.
c) By using the constructor name followed by the "base" keyword and parentheses.
d) By using the constructor name followed by the "this" keyword and parentheses.

Answer: a) By using the "base" keyword followed by parentheses and arguments.

Q50. When is the constructor of the base class called in a derived class in C#?
a) The constructor of the base class is called automatically when the derived class is created.
b) The constructor of the base class is called only when explicitly specified in the derived class.
c) The constructor of the base class is called when the derived class is deleted.
d) The constructor of the base class is never called in a derived class.

Answer: a) The constructor of the base class is called automatically when the derived class is created.

4.4.3. Overloading in derived class

Q51. What is method overloading in C#?
a) Method overloading is the process of creating multiple methods with the same name but different parameters in the same class.
b) Method overloading is the process of creating multiple methods with the same name and parameters in the same class.
c) Method overloading is the process of creating multiple methods with the same name and parameters in different classes.
d) Method overloading is the process of creating multiple methods with the same name but different parameters in different classes.

Answer: a) Method overloading is the process of creating multiple methods with the same name but different parameters in the same class.

Q52. Can a derived class overload methods from the base class in C#?
a) Yes, a derived class can overload methods from the base class with the same name and different parameters.
b) No, a derived class cannot overload methods from the base class.
c) Yes, a derived class can overload methods from the base class with the same name and same parameters.
d) Yes, a derived class can overload methods from the base class, but it must use the "override" keyword.

Answer: a) Yes, a derived class can overload methods from the base class with the same name and different parameters.

Q53. What is the benefit of method overloading in C#?
a) Method overloading allows you to create multiple methods with different names and the same parameters.
b) Method overloading allows you to create multiple methods with different names and different parameters.
c) Method overloading allows you to create multiple methods with the same name and different parameters, providing more flexibility in method calling.

d) Method overloading allows you to create multiple methods with the same name and same parameters, providing more flexibility in method calling.

Answer: c) Method overloading allows you to create multiple methods with the same name and different parameters, providing more flexibility in method calling.

Q54. Which of the following is true about method overloading in C#?
a) Method overloading is a feature only available in static methods.
b) Method overloading is a feature only available in instance methods.
c) Method overloading is a feature only available in constructors.
d) Method overloading is a feature available in both static and instance methods.

Answer: d) Method overloading is a feature available in both static and instance methods.

Q55. How does C# determine which overloaded method to call at runtime?
a) C# uses the method name to determine which overloaded method to call.
b) C# uses the return type of the method to determine which overloaded method to call.
c) C# uses the number and type of arguments passed to the method to determine which overloaded method to call.
d) C# uses the access specifier of the method to determine which overloaded method to call.

Answer: c) C# uses the number and type of arguments passed to the method to determine which overloaded method to call.

4.4.4. Hiding, using new

Q56. What is method hiding in C#?
a) Method hiding is a feature that allows you to hide methods from the base class in the derived class using the "hide" keyword.
b) Method hiding is a feature that allows you to override methods from the base class in the derived class using the "override" keyword.
c) Method hiding is a feature that allows you to hide methods from the base class in the derived class using the "new" keyword.
d) Method hiding is a feature that allows you to override methods from the base class in the derived class using the "base" keyword.

Answer: c) Method hiding is a feature that allows you to hide methods from the base class in the derived class using the "new" keyword.

Q57. How do you hide a method from the base class in a derived class using the "new" keyword in C#?
a) By using the "base" keyword followed by the method name and parentheses.
b) By using the "new" keyword followed by the method name and parentheses.
c) By using the method name followed by parentheses and parentheses.
d) By using the "override" keyword followed by the method name and parentheses.

Answer: b) By using the "new" keyword followed by the method name and parentheses.

Q58. What is the main difference between method hiding and method overriding in C#?
a) Method hiding allows you to hide methods from the base class in the derived class, while method overriding allows you to override methods from the base class in the derived class.
b) Method hiding allows you to override methods from the base class in the derived class, while method overriding allows you to hide methods from the base class in the derived class.
c) Method hiding is a feature available in instance methods only, while method overriding is a feature available in static methods only.
d) Method hiding is a feature available in static methods only, while method overriding is a feature available in instance methods only.

Answer: a) Method hiding allows you to hide methods from the base class in the derived class, while method overriding allows you to override methods from the base class in the derived class.

Q59. Which keyword is used to specify that a method in a derived class is hiding a method from the base class in C#?
a) base
b) override
c) new
d) this

Answer: c) new

Q60. What is the purpose of the "base" keyword in C#?
a) The "base" keyword is used to specify that a method in a derived class is hiding a method from the base class.
b) The "base" keyword is used to call a method from the base class in the derived class.
c) The "base" keyword is used to create a new instance of the base class in the derived class.
d) The "base" keyword is used

to override a method from the base class in the derived class.

Answer: b) The "base" keyword is used to call a method from the base class in the derived class.

4.4.5. override

Q61. What is method overriding in C#?
a) Method overriding is a feature that allows you to hide methods from the base class in the derived class using the "override" keyword.
b) Method overriding is a feature that allows you to override methods from the base class in the derived class using the "hide" keyword.
c) Method overriding is a feature that allows you to hide methods from the base class in the derived class using the "base" keyword.
d) Method overriding is a feature that allows you to override methods from the base class in the derived class using the "override" keyword.

Answer: d) Method overriding is a feature that allows you to override methods from the base class in the derived class using the "override" keyword.

Q62. How do you override a method from the base class in a derived class using the "override" keyword in C#?
a) By using the "base" keyword followed by the method name and parentheses.
b) By using the "override" keyword followed by the method name and parentheses.
c) By using the method name followed by parentheses and parentheses.
d) By using the "new" keyword followed by the method name and parentheses.

Answer: b) By using the "override" keyword followed by the method name and parentheses.

Q63. What is the main difference between method hiding and method overriding in C#?
a) Method hiding allows you to override methods from the base class in the derived class, while method overriding allows you to hide methods from the base class in the derived class.
b) Method hiding allows you to hide methods from the base class in the derived class, while method overriding allows you to override methods from the base class in the derived class.
c) Method hiding is a feature available in static methods only, while method overriding is a feature available in instance methods only.
d) Method hiding is a feature available in instance methods only, while method overriding is a feature available in static methods only.

Answer: b) Method hiding allows you to hide methods from the base class in the derived class, while method overriding allows you to override methods from the base class in the derived class.

Q64. Which keyword is used to specify that a method in a derived class is overriding a method from the base class in C#?
a) base

b) override
c) new
d) this

Answer: b) override

Q65. What is the purpose of the "base" keyword in C#?
a) The "base" keyword is used to call a method from the base class in the derived class.
b) The "base" keyword is used to create a new instance of the base class in the derived class.
c) The "base" keyword is used to specify that a method in a derived class is hiding a method from the base class.
d) The "base" keyword is used to override a method from the base class in the derived class.

Answer: a) The "base" keyword is used to call a method from the base class in the derived class.

4.4.6. sealed methods

Q66. What are sealed methods in C#?
a) Sealed methods are methods declared with the "sealed" keyword that cannot be inherited or overridden in derived classes.
b) Sealed methods are methods declared with the "override" keyword that cannot be hidden in derived classes.
c) Sealed methods are methods declared with the "sealed" keyword that can only be accessed from within the class they are defined in.
d) Sealed methods are methods declared with the "override" keyword that can only be accessed from within the class they are defined in.

Answer: a) Sealed methods are methods declared with the "sealed" keyword that cannot be inherited or overridden in derived classes.

Q67. How do you declare a sealed method in C#?
a) By using the "sealed" keyword followed by the method name and method body.
b) By using the "override" keyword followed by the method name and method body.
c) By using the "sealed" keyword followed by the method name and parentheses.
d) By using the "override" keyword followed by the method name and parentheses.

Answer: a) By using the "sealed" keyword followed by the method name and method body.

Q68. What is the main purpose of using sealed methods in C#?
a) Sealed methods prevent the method from being called from outside the class.
b) Sealed methods prevent the method from being accessed within the class.
c) Sealed methods prevent the method from being inherited or overridden in derived classes.
d) Sealed methods prevent the method from being hidden in derived classes.

Answer: c) Sealed methods prevent the method from being inherited or overridden in derived classes.

Q69. Which of the following is true about sealed methods in C#?
a) Sealed methods can be accessed from outside the class.
b) Sealed methods can be

hidden in derived classes.
c) Sealed methods can be inherited in derived classes.
d) Sealed methods can be overridden in derived classes.

Answer: a) Sealed methods can be accessed from outside the class.

4.5. Abstract Classes

Q70. What is an abstract class in C#?

a) An abstract class is a class that cannot be instantiated and may contain abstract methods that must be implemented by its derived classes.
b) An abstract class is a class that can be instantiated but cannot contain any methods.
c) An abstract class is a class that can only be accessed from within the class it is defined in.
d) An abstract class is a class that can only contain static members.

Answer: a) An abstract class is a class that cannot be instantiated and may contain abstract methods that must be implemented by its derived classes.

Q71. How do you declare an abstract class in C#?
a) By using the "abstract" keyword followed by the class name and class body.
b) By using the "class" keyword followed by the class name and class body.
c) By using the "void" keyword followed by the class name and class body.
d) By using the "new" keyword followed by the class name and class body.

Answer: a) By using the "abstract" keyword followed by the class name and class body.

Q72. What is the main purpose of using an abstract class in C#?
a) An abstract class allows you to define and access static members without creating an instance of the class.
b) An abstract class allows you to create multiple instances of the class.
c) An abstract class allows you to define abstract methods that must be implemented by its derived classes.
d) An abstract class allows you to define methods that cannot be overridden in derived classes.

Answer: c) An abstract class allows you to define abstract methods that must be implemented by its derived classes.

Q73. Which of the following is true about abstract classes in C#?
a) Abstract classes cannot contain any methods or members.
b) Abstract classes cannot be inherited by other classes.
c) Abstract classes can contain abstract methods as well as non-abstract methods.
d) Abstract classes cannot be used in C#.

Answer: c) Abstract classes can contain abstract methods as well as non-abstract methods.

Q74. How do you declare an abstract method in an abstract class in C#?
a) By using the "abstract" keyword followed by the method name and method body.
b) By using the "virtual" keyword followed by the method name and method body.
c) By using the "abstract" keyword followed by the method name and parentheses.
d) By using the "virtual" keyword followed by the method name and parentheses.

Answer: c) By using the "abstract" keyword followed by the method name and parentheses.

Q75. What is the purpose of an abstract method in an abstract class in C#?
a) Abstract methods define the data type of variables in the abstract class.
b) Abstract methods define methods that cannot be overridden in derived classes.
c) Abstract methods define methods that must be implemented by its derived classes.
d) Abstract methods define methods that can only be accessed from within the class.

Answer: c) Abstract methods define methods that must be implemented by its derived classes.

4.6. Abstract Methods

Q76. What is an abstract method in C#?
a) An abstract method is a method that cannot be called from outside the class.
b) An abstract method is a method that can only be accessed from within the class it is defined in.
c) An abstract method is a method that cannot contain any code and must be implemented by its derived classes.
d) An abstract method is a method that can only contain static members.

Answer: c) An abstract method is a method that cannot contain any code and must be implemented by its derived classes.

Q77. How do you declare an abstract method in C#?
a) By using the "abstract" keyword followed by the method name and method body.
b) By using the "virtual" keyword followed by the method name and method body.
c) By using the "abstract" keyword followed by the method name and parentheses.
d) By using the "virtual" keyword followed by the method name and parentheses.

Answer: c) By using the "abstract" keyword followed by the method name and parentheses.

Q78. Can an abstract method have a method body in C#?
a) Yes, an abstract method can have a method body containing code.
b) No, an abstract method cannot have a method body and must be implemented by its derived classes.
c) Yes, an abstract method can have a method body, but it cannot be called from outside the class.
d) No, an abstract method cannot have a method body and cannot be implemented by its derived classes.

Answer: b) No, an abstract method cannot have a method body and must be implemented by its derived classes.

Q79. Which of the following is true about abstract methods in C#?
a) Abstract methods can be called from outside the class.
b) Abstract methods can be accessed from within the class they are defined in.
c) Abstract methods cannot be overridden in derived classes.
d) Abstract methods must be implemented by the derived classes.

Answer: d) Abstract methods must be implemented by the derived classes.

Q80. What is the purpose of an abstract method in C#?
a) Abstract methods define the data type of variables in the abstract class.
b) Abstract methods define methods that cannot be overridden in derived classes.
c) Abstract methods define methods that can only be accessed from within the class.
d) Abstract methods define methods that must be implemented by the derived classes.

Answer: d) Abstract methods define methods that must be implemented by the derived classes.

4.7. Sealed Classes

Q81. What is a sealed class in C#?
a) A sealed class is a class that cannot be instantiated and may contain abstract methods that must be implemented by its derived classes.
b) A sealed class is a class that can be instantiated but cannot contain any methods.
c) A sealed class is a class that can only be accessed from within the class it is defined in.
d) A sealed class is a class that cannot be inherited and cannot have derived classes.

Answer: d) A sealed class is a class that cannot be inherited and cannot have derived classes.

Q82. How do you declare a sealed class in C#?
a) By using the "sealed" keyword followed by the class name and class body.
b) By using the "class" keyword followed by the class name and class body.
c) By using the "void" keyword followed by the class name and class body.
d) By using the "new" keyword followed by the class name and class body.

Answer: a) By using the "sealed" keyword followed by the class name and class body.

Q83. What is the main purpose of using a sealed class in C#?
a) A sealed class allows you to define and access static members without creating an instance of the class.

b) A sealed class allows you to create multiple instances of the class.
c) A sealed class allows you to prevent the class from being inherited and having derived classes.
d) A sealed class allows you to define abstract methods that must be implemented by its derived classes.

Answer: c) A sealed class allows you to prevent the class from being inherited and having derived classes.

Q84.

Which of the following is true about sealed classes in C#?
a) Sealed classes can be inherited by other classes.
b) Sealed classes can have derived classes.
c) Sealed classes can contain abstract methods.
d) Sealed classes can be instantiated.

Answer: d) Sealed classes can be instantiated.

Q85. Can you use the "abstract" keyword with a sealed class in C#?
a) Yes, you can use the "abstract" keyword with a sealed class.
b) No, you cannot use the "abstract" keyword with a sealed class.
c) Yes, you can use the "abstract" keyword with a sealed class, but it will make the class non-sealed.
d) No, you cannot use the "abstract" keyword with a sealed class, as it contradicts the purpose of sealing the class.

Answer: b) No, you cannot use the "abstract" keyword with a sealed class

.

5. Lecture: Interfaces and Operator Overloading

5.1. Interfaces

Q1. What is an interface in C#?
a) A blueprint for a class that defines the properties, methods, and events the class must implement.
b) A special type of class that can be directly instantiated.
c) A keyword used to define static methods in a class.
d) A keyword used to define constructors in a class.

Answer: a) A blueprint for a class that defines the properties, methods, and events the class must implement.

Q2. How do you implement an interface in a class in C#?
a) By using the "implements" keyword followed by the interface name.
b) By using the "implements" keyword followed by the class name.
c) By using the "extends" keyword followed by the interface name.
d) By using the "extends" keyword followed by the class name.

Answer: a) By using the "implements" keyword followed by the interface name.

Q3. What is the purpose of explicitly implementing an interface in C#?
a) To hide the interface methods from the class's public interface.
b) To provide a different implementation for the interface methods based on the context of the class.
c) To override the interface methods with the "override" keyword.
d) To define static methods in the class.

Answer: a) To hide the interface methods from the class's public interface.

Q4. Can a class implement multiple interfaces in C#?
a) No, a class can only implement one interface.
b) Yes, a class can implement multiple interfaces.
c) Yes, but only if the interfaces have the same method names.
d) No, interfaces cannot be implemented by classes.

Answer: b) Yes, a class can implement multiple interfaces.

Q5. What is the purpose of inheritance in interfaces in C#?
a) Inheritance in interfaces allows a class to inherit properties and methods from multiple interfaces.
b) Inheritance in interfaces allows a class to inherit properties and methods from a base class.
c) Inheritance in interfaces allows a class to override the implementation of methods in the interface.
d) Inheritance in interfaces is not allowed in C#.

Answer: a) Inheritance in interfaces allows a class to inherit properties and methods from multiple interfaces.

5.1.1. Implementing an interface

Q6. What does it mean to implement an interface in C#?
a) Implementing an interface means creating an instance of the interface.
b) Implementing an interface means providing the method implementations specified in the interface.
c) Implementing an interface means inheriting from the interface.
d) Implementing an interface means calling the methods defined in the interface.

Answer: b) Implementing an interface means providing the method implementations specified in the interface.

Q7. Which keyword is used to implement an interface in a class in C#?
a) implements

b) extends
c) implementsinterface
d) interface

Answer: a) implements

Q8. When would you use explicit interface implementation in C#?
a) When you want to hide the interface methods from the public interface of the class.
b) When you want to provide multiple implementations of the same interface.
c) When you want to make the interface methods accessible only through the interface reference.
d) When you want to override the interface methods with different access modifiers.

Answer: c) When you want to make the interface methods accessible only through the interface reference.

Q9. Can a class inherit from both a class and an interface in C#?
a) Yes, a class can inherit from both a class and an interface.
b) No, a class can only inherit from another class or an interface, but not both.
c) Yes, but only if the class implements the interface explicitly.
d) No, C# does not support multiple inheritance.

Answer: a) Yes, a class can inherit from both a class and an interface.

Q10. What does inheritance in interfaces allow in C#?
a) Inheritance in interfaces allows a class to inherit the fields and properties of the interface.
b) Inheritance in interfaces allows a class to inherit the implementation of methods from the interface.
c) Inheritance in interfaces allows a class to inherit from multiple interfaces.
d) Inheritance in interfaces allows a class to override the methods of the interface.

Answer: c) Inheritance in interfaces allows a class to inherit from multiple interfaces.

5.1.2. Explicitly implementing an interface

Q11. What is the main difference between explicitly implementing an interface and implicitly implementing an interface in C#?
a) There is no difference; both approaches result in the same behavior.
b) Explicitly implementing an interface allows you to provide different method implementations based on the context, while implicitly implementing an interface provides a single implementation.
c) Explicitly implementing an interface is more efficient than implicitly implementing an interface.
d) Explicitly implementing an interface is a requirement in C#.

Answer: b) Explicitly implementing an interface allows you to provide different method implementations based on the context, while implicitly implementing an interface provides a single implementation.

Q12. What are default interface methods in C#?
a) Default interface methods are methods that are automatically available in all classes that implement the interface.
b) Default interface methods are methods that must be implemented by classes that implement the interface.
c) Default interface methods are methods that are marked with the "default" keyword and have a predefined implementation.
d) Default interface methods are methods that are static and can be called without creating an instance of the implementing class.

Answer: c) Default interface methods are methods that are marked with the "default" keyword and have a predefined implementation.

Q13. Can a class override a default interface method in C#?
a) Yes, a class can override a default interface method.

b) No, a class cannot override a default interface method.
c) Yes, but only if the class implements multiple interfaces with the same default method name.
d) No, default interface methods are automatically available and cannot be overridden.

Answer: b) No, a class cannot override a default interface method.

Q14. What is the main advantage of default interface methods in C#?
a) Default interface methods allow you to define common method implementations that are automatically available to all implementing classes.
b) Default interface methods allow you to define methods that are only accessible through the interface reference.
c) Default interface methods allow you to create static methods in an interface.
d) Default interface methods allow you to define methods with different access modifiers in an interface.

Answer: a) Default interface methods allow you to define common method implementations that are automatically available to all implementing classes.

5.1.3. Inheritance in interfaces

Q15. What does inheritance in interfaces allow in C#?
a) Inheritance in interfaces allows a class to inherit the fields and properties of the interface.
b) Inheritance in interfaces allows a class to inherit the implementation of methods from the interface.
c) Inheritance in interfaces allows a class to inherit from multiple interfaces.
d) Inheritance in interfaces allows a class to override the methods of the interface.

Answer: c) Inheritance in interfaces allows a class to inherit from multiple interfaces.

Q16. Can a class inherit from both a class and an interface in C#?
a) Yes, a class can inherit from both a class and an interface.
b) No, a class can only inherit from

another class or an interface, but not both.
c) Yes, but only if the class implements the interface explicitly.
d) No, C# does not support multiple inheritance.

Answer: a) Yes, a class can inherit from both a class and an interface.

5.1.4. Default interface methods

Q17. What are default interface methods in C#?
a) Default interface methods are methods that are automatically available in all classes that implement the interface.
b) Default interface methods are methods that must be implemented by classes that implement the interface.
c) Default interface methods are methods that are marked with the "default" keyword and have a predefined implementation.
d) Default interface methods are methods that are static and can be called without creating an instance of the implementing class.

Answer: c) Default interface methods are methods that are marked with the "default" keyword and have a predefined implementation.

Q18. Can a class override a default interface method in C#?
a) Yes, a class can override a default interface method.
b) No, a class cannot override a default interface method.
c) Yes, but only if the class implements multiple interfaces with the same default method name.
d) No, default interface methods are automatically available and cannot be overridden.

Answer: b) No, a class cannot override a default interface method.

Q19. What is the main advantage of default interface methods in C#?
a) Default interface methods allow you to define common method implementations that are automatically available to all implementing classes.
b) Default interface methods allow you to define methods that are only accessible through the interface reference.
c) Default interface methods allow you to create static methods in an interface.
d) Default interface methods allow you to define methods with different access modifiers in an interface.

Answer: a) Default interface methods allow you to define common method implementations that are automatically available to all implementing classes.

5.2. Operator overloading

Q20. What is operator overloading in C#?
a) Operator overloading allows you to define multiple operators with the same name but different functionality.
b) Operator overloading allows you to create new operators in C#.
c) Operator overloading allows you to define additional methods for built-in operators in C#.
d) Operator overloading allows you to override the default behavior of operators for custom types.

Answer: d) Operator overloading allows you to override the default behavior of operators for custom types.

Q21. Which keyword is used to overload an operator in C#?
a) override
b) overload
c) operator
d) custom

Answer: c) operator

Q22. Which operator cannot be overloaded in C#?
a) Addition (+)
b) Subtraction (-)
c) Assignment (=)
d) Division (/)

Answer: c) Assignment (=)

Q23. What is the syntax for overloading the addition operator (+) in C#?
a) public int operator+(int a, int b) { /* implementation */ }
b) public int +(int a, int b) { /* implementation */ }
c) public int Add(int a, int b) { /* implementation */ }
d) public int Add(int a) { /* implementation */ }

Answer: a) public int operator+(int a, int b) { /* implementation */ }

Q24. What is the main purpose of operator overloading in C#?
a) Operator overloading allows you to define multiple methods with the same name but different functionality.
b) Operator overloading allows you to create new operators.
c) Operator overloading allows you to define additional methods for built-in operators to work with custom types.
d) Operator overloading allows you to override the default behavior of built-in operators.

Answer: c) Operator overloading allows you to define additional methods for built-in operators to work with custom types.

6. Lecture: Reference and Value Types, Arrays, and Indexers

6.1. Reference and Value Types

Q1. Which of the following data types is considered a reference type in C#?
a) int
b) double
c) string
d) char

Answer: c) string

Q2. What is the key difference between reference types and value types in C#?
a) Reference types are allocated on the stack, while value types are allocated on the heap.
b) Reference types are passed by value, while value types are passed by reference.
c) Reference types store the value directly, while value types store a reference to the value.
d) Reference types are immutable, while value types are mutable.

Answer: c) Reference types store the value directly, while value types store a reference to the value.

Q3. In C#, which of the following data types is considered a value type?
a) int
b) List<string>
c) object
d) string[]

Answer: a) int

Q4. What happens when you assign a value type variable to another variable in C#?
a) The values are copied, and both variables point to the same memory location.
b) The values are copied, and each variable has its own memory location.
c) The variables point to the same memory location, but the values are not copied.
d) Value types cannot be assigned to other variables.

Answer: b) The values are copied, and each variable has its own memory location.

Q5. Which of the following is an example of a reference type in C#?
a) float
b) double
c) decimal
d) string

Answer: d) string

6.2. Value Types

Q6. What is the difference between a struct and a class in C#?
a) A struct is a value type, while a class is a reference type.
b) A struct can have methods and properties, while a class can only have fields.
c) A struct can be null, while a class cannot be null.
d) A struct can inherit from other classes, while a class cannot inherit from a struct.

Answer: a) A struct is a value type, while a class is a reference type.

Q7. Which of the following is an example of a value type in C#?
a) StringBuilder
b) DateTime
c) List<int>
d) Math

Answer: b) DateTime

Q8. What happens when you pass a value type to a method as an argument in C#?
a) The method receives a copy of the original variable, and changes to the parameter do not affect the original variable.
b) The method receives a reference to the original variable, and changes to the parameter affect the original variable.
c) The method receives the memory address of the variable.
d) Value types cannot be passed as arguments to methods.

Answer: a) The method receives a copy of the original variable, and changes to the parameter do not affect the original variable.

6.2.1. struct

Q9. Which of the following statements about structs in C# is true?
a) Structs support inheritance from other structs or classes.
b) Structs can have default constructors.
c) Structs can be used as base classes for other structs.
d) Structs support garbage collection.

Answer: b) Structs can have default constructors.

Q10. In C#, when you create an instance of a struct using the "new" keyword, where is the struct instance allocated?
a) On the stack
b) On the heap
c) In a separate memory location called the "struct pool"
d) In the method that creates the struct instance

Answer: a) On the stack

Q11. Can structs in C# be inherited from other classes or structs?
a) Yes, structs can inherit from other structs or classes.
b) No, structs cannot be inherited in C#.
c) Structs can only inherit from other structs, not classes.
d) Structs can only inherit from classes, not other structs.

Answer: b) No, structs cannot be inherited in C#.

Q12. Which of the following access modifiers can be applied to struct members in C#?
a) public and protected
b) private and protected
c) public, private, and protected
d) public, private, protected, and internal

Answer: c) public, private, and protected

Q13. What happens if you don't specify any constructor for a struct in C#?
a) The struct cannot be instantiated.
b) The C# compiler automatically generates a parameterless default constructor.
c) It results in a compilation error.
d) The struct is allocated on the heap instead of the stack.

Answer: b) The C# compiler automatically generates a parameterless default constructor.

6.2.2. enum

Q14. What is the purpose of an enum in C#?

a) Enums allow you to define a set of named constants representing integral values.
b) Enums are used to define a set of methods and properties for a class.
c) Enums allow you to create instances of a class with predefined values.
d) Enums are used to define static classes with utility methods.

Answer: a) Enums allow you to define a set of named constants representing integral values.

Q15. How do you

declare an enum in C#?
a) enum Color { Red, Green, Blue }
b) enum { Red, Green, Blue }
c) enum Colors = { "Red", "Green", "Blue" }
d) enum Colors (Red, Green, Blue)

Answer: a) enum Color { Red, Green, Blue }

Q16. In C#, can an enum be used as the underlying type for another enum?
a) Yes, enums can be used as the underlying type for other enums.
b) No, enums cannot be used as the underlying type for other enums.
c) Only numeric data types can be used as the underlying type for an enum.
d) It depends on the version of C# being used.

Answer: a) Yes, enums can be used as the underlying type for other enums.

Q17. What is the default underlying type of an enum in C#?
a) int
b) byte
c) short
d) long

Answer: a) int

Q18. How can you explicitly specify the underlying type of an enum in C#?
a) By using the "underlying" keyword before the enum declaration.
b) By using the "base" keyword before the enum declaration.
c) By specifying the desired type in parentheses after the enum declaration.
d) By using the "typeof" keyword before the enum declaration.

Answer: c) By specifying the desired type in parentheses after the enum declaration.

6.3. out and ref

Q19. What are "out" and "ref" keywords used for in C#?
a) They are used to define constant values.
b) They are used to specify access modifiers for methods.
c) They are used to pass parameters by reference to methods.
d) They are used to create aliases for variables.

Answer: c) They are used to pass parameters by reference to methods.

Q20. What is the main difference between "out" and "ref" parameters in C#?
a) "out" parameters must be initialized before they are passed to a method, while "ref" parameters do not require initialization.
b) "out" parameters can be modified inside the method, while "ref" parameters cannot be modified.
c) "out" parameters are used for value types, while "ref" parameters are used for reference types.
d) There is no difference; "out" and "ref" parameters are interchangeable.

Answer: a) "out" parameters must be initialized before they are passed to a method, while "ref" parameters do not require initialization.

Q21. In C#, can you use "out" or "ref" keywords with methods that have a return value?
a) Yes, "out" and "ref" keywords can be used with methods that have a return value.
b) No, "out" and "ref" keywords cannot be used with methods that have a return value.
c) Only "out" keyword can be used with methods that have a return value.
d) Only "ref" keyword can be used with methods that have a return value.

Answer: b) No, "out" and "ref" keywords cannot be used with methods that have a return value.

Q22. What is the behavior of an "out" parameter inside a method in C#?
a) The method must assign a value to the "out" parameter before it returns.
b) The method cannot modify the value of the "out" parameter.
c) The "out" parameter must be initialized with a default value before calling the method.
d) The "out" parameter is treated like a regular method parameter.

Answer: a) The method must assign a value to the "out" parameter before it returns.

6.4. Nullable Types

Q23. What is the purpose of nullable types in C#?
a) Nullable types allow you to declare variables without assigning a value to them.
b) Nullable types allow you to create variables that can hold either a value or "null."
c) Nullable types allow you to define methods with optional parameters.
d) Nullable types allow you to create variables with unlimited range.

Answer: b) Nullable types allow you to create variables that can hold either a value or "null."

Q24. How do you declare a nullable type in C#?
a) int? num;
b) nullable int num;
c) int num = null;
d) int num = Nullable;

Answer: a) int? num;

Q25. What is the syntax for assigning a value to a nullable type in C#?
a) num = 10;
b) num = 10.0;
c) num = "10";
d) num = null;

Answer: a) num = 10;

Q26. How do you check if a nullable type has a value in C#?
a) Use the "HasValue" property.
b) Use the "IsSet" property.
c) Use the "IsNotNull" method.
d) Use the "Exists" method.

Answer: a) Use the "HasValue" property.

Q27. What happens if you try to assign "null" to a non-nullable value type variable in C#?
a) The C# compiler generates an error.
b) The variable is automatically converted to a nullable type.
c) The value "null" is stored as the default value for the data type.
d) The variable is assigned a default value based on the data type.

Answer: a) The C# compiler generates an error.

6.5. Nullable Reference Types

Q28. What is the purpose of nullable reference types in C#?
a) Nullable reference types allow you to define classes that cannot be assigned a null value.
b) Nullable reference types allow you to define classes that can be assigned a null value.
c) Nullable reference types allow you to create instances of classes without explicitly initializing them.
d) Nullable reference types allow you to define static classes with utility methods.

Answer: b) Nullable reference types allow you to define classes that can be assigned a null value.

Q29. In C#, how do you enable nullable reference types for a project?
a) By adding the "nullable" keyword to the class definition.
b) By enabling the "nullable" setting in the project properties.
c) By adding a "using System.Nullable;" statement at the beginning of the file.
d) Nullable reference types are enabled by default and do not require any additional configuration.

Answer: b) By enabling the "nullable" setting in the project properties.

Q30. What is the syntax for declaring a nullable reference type in C#?
a) string name;
b) string? name;
c) string! name;
d) string& name;

Answer: b) string? name;

Q31. How do you assign "null" to a nullable reference type in C#?
a) name = "";
b) name = "null";
c) name = null;
d) name = undefined;

Answer: c) name = null;

6.6. ?? and ??=

Q32. What is the purpose of the ?? operator in C#?
a) It is a logical OR operator.
b) It is a null-coalescing operator.
c) It is a ternary conditional operator.
d) It is a bitwise OR operator.

Answer: b) It is a null-coalescing operator.

Q33. What does the ?? operator do in C#?
a) It checks if the left operand is equal

to the right operand.
b) It assigns the value of the left operand to the right operand if the left operand is not null; otherwise, it assigns the value of the right operand.
c) It returns the value of the left operand if it is not null; otherwise, it returns the value of the right operand.
d) It performs a logical OR operation between the left and right operands.

Answer: c) It returns the value of the left operand if it is not null; otherwise, it returns the value of the right operand.

Q34. What is the syntax for using the ?? operator in C#?
a) var result = a ?? b;
b) var result = a ? b : c;
c) var result = a ??= b;

d) var result = a | b;

Answer: a) var result = a ?? b;

Q35. What does the ??= operator do in C#?
a) It checks if the left operand is equal to the right operand.
b) It assigns the value of the left operand to the right operand if the left operand is not null; otherwise, it assigns the value of the right operand.
c) It returns the value of the left operand if it is not null; otherwise, it returns the value of the right operand.
d) It assigns the value of the right operand to the left operand if the left operand is null; otherwise, it keeps the value of the left operand.

Answer: d) It assigns the value of the right operand to the left operand if the left operand is null; otherwise, it keeps the value of the left operand.

6.7. Working with Arrays (single, multidim, jagged), Array Class members

Q36. What is the maximum number of dimensions that an array can have in C#?
a) 2
b) 3
c) 4
d) There is no limit on the number of dimensions.

Answer: d) There is no limit on the number of dimensions.

Q37. What is the default value for elements in a numeric array in C#?
a) 0
b) 1
c) -1
d) null

Answer: a) 0

Q38. How do you declare a single-dimensional array in C#?
a) int[] numbers;
b) array numbers[];
c) numbers[] int;
d) int[1] numbers;

Answer: a) int[] numbers;

Q39. How do you initialize an array with values at the time of declaration in C#?
a) int[] numbers = new int[] { 1, 2, 3 };
b) int numbers = { 1, 2, 3 };
c) int[] numbers = (1, 2, 3);
d) int numbers = new int[3] { 1, 2, 3 };

Answer: a) int[] numbers = new int[] { 1, 2, 3 };

Q40. How do you access the length of an array in C#?
a) numbers.length
b) numbers.Length
c) length(numbers)
d) Length(numbers)

Answer: b) numbers.Length

6.8. Indices and Ranges

Q41. What is the purpose of indices in C#?
a) Indices are used to access elements of a collection by their position.
b) Indices are used to define access modifiers for methods.
c) Indices are used to specify the length of an array.
d) Indices are used to perform arithmetic operations on numeric data types.

Answer: a) Indices are used to access elements of a collection by their position.

Q42. How do you create an index in C#?
a) By using the "index" keyword before a variable declaration.
b) By using the "new" keyword and specifying the index position.
c) By using the "[" and "]" brackets with the desired index value.
d) Indices are created automatically by the C# compiler.

Answer: c) By using the "[" and "]" brackets with the desired index value.

Q43. What is the syntax for accessing the first element of an array using an index in C#?
a) numbers[1]
b) numbers[0]
c) numbers[-1]
d) numbers[0th]

Answer: b) numbers[0]

Q44. What is the purpose of ranges in C#?
a) Ranges are used to access elements of a collection by their position.
b) Ranges are used to define access modifiers for methods.
c) Ranges are used to specify the length of an array.
d) Ranges are used to create subsets of a collection.

Answer: d) Ranges are used to create subsets of a collection.

Q45. How do you create a range in C#?
a) By using the "range" keyword before specifying the start and end positions.
b) By using the ".." operator with the start and end positions.
c) By using the "new" keyword and specifying the range positions.
d) Ranges are created automatically by the C# compiler.

Answer: b) By using the ".." operator with the start and end positions.

6.9. Indexers

Q46. What is an indexer in C#?
a) An indexer is a special type of property that allows you to access elements of a collection using square brackets.
b) An indexer is a method used to retrieve data from a database.
c) An indexer is a reserved keyword used for indexing arrays.
d) An indexer is a type of loop used for iterating over collections.

Answer: a) An indexer is a special type of property that allows you to access elements of a collection using square brackets.

Q47. How do you define an indexer in C#?
a) By using the "indexer" keyword before a method declaration.
b) By using the "this" keyword and specifying the index parameter.
c) By using the "new" keyword and specifying the index position.
d) Indexers are defined automatically for arrays in C#.

Answer: b) By using the "this" keyword and specifying the index parameter.

Q48. What is the return type of an indexer in C#?
a) void
b) int
c) The data type of the elements in the collection.
d) The same data type as the index parameter.

Answer: c) The data type of the elements in the collection.

Q49. How do you access the elements of a collection using an indexer in C#?
a) By calling the indexer method and passing the index as an argument.
b) By using the collection name followed by square brackets and the index value.
c) By using the "get" keyword and specifying the index position.
d) Indexers cannot be used to access elements of a collection.

Answer: b) By using the collection name followed by square brackets and the index value.

Q50. Can a C# class have multiple indexers?
a) Yes, a class can have multiple indexers with different data types.
b) No, a class can have only one indexer.
c) Yes, a class can have multiple indexers, but they must all have the same data type.
d) Indexers are only applicable to arrays, not classes.

Answer: a) Yes, a class can have multiple indexers with different data types.

7. Lecture: Generics and Collections

Sure! Here are 5 multiple-choice questions with answers for each point and subpoint:

7.1. Generic classes

Q1. What is the main purpose of using generic classes in C#?
a) To create classes that can only work with a specific data type.
b) To create classes that can work with multiple data types without the need for casting or boxing/unboxing.
c) To create classes that can only be used with value types.
d) To create classes that can only be used with reference types.

Answer: b) To create classes that can work with multiple data types without the need for casting or boxing/unboxing.

Q2. How do you declare a generic class in C#?
a) class MyClass<T> {}
b) class<T> MyClass {}
c) generic class MyClass {}
d) class MyClass {} <T>

Answer: a) class MyClass<T> {}

Q3. In a generic class definition, what does "T" represent?
a) "T" is a reserved keyword that stands for "type."
b) "T" is a placeholder for the data type that will be specified when creating an instance of the class.
c) "T" is a predefined data type that represents any numeric value.
d) "T" is an abbreviation for "Template," indicating that the class is a template for other classes.

Answer: b) "T" is a placeholder for the data type that will be specified when creating an instance of the class.

Q4. How many different data types can a single instance of a generic class in C# work with?
a) Only one data type.
b) Two data types, one for read-only operations and another for write operations.
c) As many data types as needed, depending on how the class is designed.
d) Generic classes can only work with value types.

Answer: c) As many data types as needed, depending on how the class is designed.

Q5. What is the benefit of using generic classes over non-generic classes in C#?
a) Generic classes allow for better performance because they avoid boxing/unboxing operations.
b) Non-generic classes have better type safety compared to generic classes.
c) Generic classes are easier to implement and require less coding.
d) Non-generic classes are more flexible and can work with any data type.

Answer: a) Generic classes allow for better performance because they avoid boxing/unboxing operations.

7.2. Generic methods

Q6. What is a generic method in C#?
a) A method that can only be used with a specific data type.
b) A method that can work with multiple data types without the need for casting or boxing/unboxing.
c) A method that can only be used with reference types.
d) A method that can only be used with value types.

Answer: b) A method that can work with multiple data types without the need for casting or boxing/unboxing.

Q7. How do you declare a generic method in C#?
a) void MyMethod<T>() {}
b) void<T> MyMethod() {}
c) generic void MyMethod() {}
d) void MyMethod {} <T>

Answer: a) void MyMethod<T>() {}

Q8. In a generic method definition, what does "T" represent?
a) "T" is a reserved keyword that stands for "type."
b) "T" is a placeholder for the data type that will be specified when calling the method.
c) "T" is a predefined data type that represents any numeric value.
d) "T" is an abbreviation for "Template," indicating that the method is a template for other methods.

Answer: b) "T" is a placeholder for the data type that will be specified when calling the method.

Q9. What is the benefit of using generic methods over non-generic methods in C#?
a) Generic methods allow for better performance because they avoid boxing/unboxing operations.
b) Non-generic methods have better type safety compared to generic methods.
c) Generic methods are easier to implement and require less coding.
d) Non-generic methods are more flexible and can work with any data type.

Answer: a) Generic methods allow for better performance because they avoid boxing/unboxing operations.

Q10. Can a generic method be used in a non-generic class in C#?
a) Yes, a generic method can be used in a non-generic class.
b) No, a generic method can only be used in a generic class.
c) Generic methods can only be used in static classes.
d) Non-generic classes do not support methods in C#.

Answer: a) Yes, a generic method can be used in a non-generic class.

7.3. Generic Constraints

Q11. What are generic constraints in C#?
a) Restrictions applied to generic types to ensure they can only be used with specific data types.
b) Rules that limit the number of type parameters a generic class or method can have.
c) Requirements for a class to be considered generic in C#.
d) Guidelines for naming generic types in C#.

Answer: a) Restrictions applied to generic types to ensure they can only be used with specific data types.

Q12. How do you specify a generic constraint in C#?
a) By using the "constrain" keyword followed by the data type.
b) By using the "where" keyword followed by the data type.
c) By using the "is" keyword followed by the data type.
d) Generic types do not support constraints in C#.

Answer: b) By using the "where" keyword followed by the data type.

Q13. What is the purpose of using a generic constraint in C#?
a) To allow a generic type to work with any data type.
b) To limit the generic type to a specific data type or a set of data types.
c) To ensure that the generic type can only work with reference types.
d) To improve the performance of generic types.

Answer: b) To limit the generic type to a specific data type or a set of data types.

Q14. How many generic constraints can be specified for a single type parameter in C#?
a) Only one constraint is allowed per type parameter.
b) Two or more constraints can be specified for a type parameter.
c) Constraints cannot be applied to type parameters in C#.
d) Constraints can only be applied to value types.

Answer: b) Two or more constraints can be specified for a type parameter.

Q15. Which of the following is a valid generic constraint in C#?
a) where T : class, struct
b) where T : struct, new()
c) where T : T
d) where T : System.Collections.IEnumerable

Answer: b) where T : struct, new()

7.4. Collections – generic and non-generic

Q16. What is a collection in C#?
a) A collection is a group of related methods.
b) A collection is a class that can only store integers.
c) A collection is a data structure that can hold multiple elements of the same or different data types.
d) A collection is a reserved keyword in C#.

Answer: c) A collection is a data structure that can hold multiple elements of the same or different data types.

Q17. How are collections in C# different from arrays?
a) Collections have a fixed size, while arrays can dynamically resize.
b) Collections can hold elements of different data types, while arrays can only hold elements of the same data

type.
c) Arrays have a fixed size, while collections can dynamically resize.
d) There is no difference; arrays and collections are interchangeable.

Answer: c) Arrays have a fixed size, while collections can dynamically resize.

Q18. Which of the following is an example of a non-generic collection in C#?
a) List<T>
b) Dictionary<TKey, TValue>
c) ArrayList
d) Queue<T>

Answer: c) ArrayList

Q19. What is the advantage of using generic collections over non-generic collections in C#?
a) Generic collections are more memory-efficient than non-generic collections.
b) Generic collections provide better type safety, as they can only store elements of a specific data type.
c) Non-generic collections can hold more elements than generic collections.
d) Generic collections are faster in terms of element access and manipulation.

Answer: b) Generic collections provide better type safety, as they can only store elements of a specific data type.

Q20. Which of the following is a valid generic collection in C#?
a) Stack<T>

b) Hashtable
c) LinkedList<T>
d) SortedList

Answer: a) Stack<T>

7.5. Collection Examples based on ICollection, IList, IDictionary (both generic and non-generic)

Q21. What is the purpose of the ICollection interface in C#?
a) The ICollection interface is used to define a generic collection in C#.
b) The ICollection interface provides a way to iterate over a collection using the foreach loop.
c) The ICollection interface is used to define a non-generic collection in C#.
d) The ICollection interface provides a way to store elements in a sorted order.

Answer: c) The ICollection interface is used to define a non-generic collection in C#.

Q22. How is the IList interface different from the ICollection interface in C#?
a) The IList interface is used for generic collections, while the ICollection interface is used for non-generic collections.
b) The IList interface provides methods for adding, removing, and accessing elements by index, while the ICollection interface does not.
c) The IList interface provides methods for sorting elements in the collection, while the ICollection interface does not.
d) There is no difference; IList and ICollection interfaces are interchangeable.

Answer: b) The IList interface provides methods for adding, removing, and accessing elements by index, while the ICollection interface does not.

Q23. Which of the following is a valid example of a generic collection that implements the ICollection interface in C#?
a) Dictionary<TKey, TValue>
b) Queue<T>
c) HashSet<T>
d) ArrayList

Answer: c) HashSet<T>

Q24. What is the purpose of the IDictionary interface in C#?
a) The IDictionary interface is used to define a generic dictionary collection in C#.
b) The IDictionary interface provides a way to store elements in a sorted order.
c) The IDictionary interface is used to define a non-generic dictionary collection in C#.
d) The IDictionary interface provides a way to access elements in a collection using a key.

Answer: d) The IDictionary interface provides a way to access elements in a collection using a key.

Q25. How is the IDictionary interface different from the ICollection interface in C#?
a) The IDictionary interface provides methods for adding, removing, and accessing elements by key, while the ICollection interface does not.
b) The ICollection interface is used for generic collections, while the IDictionary interface is used for non-generic collections.
c) The IDictionary interface provides methods for sorting elements in the collection, while the ICollection interface does not.
d) There is no difference; IDictionary and ICollection interfaces are interchangeable.

Answer: a) The IDictionary interface provides methods for adding, removing, and accessing elements by key, while the ICollection interface does not.

7.6. Iterating collections using foreach

Q26. What is the purpose of the foreach loop in C#?

a) The foreach loop is used to iterate over elements in a collection one by one.
b) The foreach loop is used to perform arithmetic operations on numeric data types.
c) The foreach loop is used to define access modifiers for methods.
d) The foreach loop is used to create aliases for variables.

Answer: a) The foreach loop is used to iterate over elements in a collection one by one.

Q27. How do you use the foreach loop to iterate over a collection in C#?
a) foreach (var item in collection) {}
b) foreach (collection as var item) {}
c) foreach (item in collection) {}
d) foreach (item : collection) {}

Answer: a) foreach (var item in collection) {}

Q28. Which of the following collections can be used with the foreach loop in C#?
a) ArrayList
b) Dictionary<TKey, TValue>
c) Stack<T>
d) All of the above

Answer: d) All of the above

Q29. In the foreach loop, what is the type of the "item" variable?
a) The "item" variable is of type object.
b) The "item" variable is of type int.
c) The "item" variable is of the same data type as the elements in the collection.
d) The "item" variable is of type string.

Answer: c) The "item" variable is of the same data type as the elements in the collection.

Q30. Can you modify the elements of a collection while using the foreach loop in C#?
a) Yes, you can modify the elements of a collection while using the foreach loop.
b) No, the foreach loop is read-only and does not allow modification of elements.
c) Modification is allowed, but it requires casting the "item" variable to the appropriate data type.
d) Only non-generic collections can be modified using the foreach loop.

Answer: b) No, the foreach loop is read-only and does not allow modification of elements.

7.7. Using Tuples to pass multiple values to a function

Q31. What is a Tuple in C#?
a) A Tuple is a data structure that can hold a single value.
b) A Tuple is a special type of collection that can only store integers.
c) A Tuple is a class used to store multiple values of different data types as a single unit.
d) Tuples are not supported in C#.

Answer: c) A Tuple is a class used to store multiple values of different data types as a single unit.

Q32. How do you create a Tuple in C#?
a) Tuple myTuple = new Tuple();
b) Tuple<int, string> myTuple = new Tuple<int, string>(1, "Hello");
c) Tuple myTuple = (1, "Hello");
d) Tuples are created automatically when assigning multiple values to a variable.

Answer: b) Tuple<int, string> myTuple = new Tuple<int, string>(1, "Hello");

Q33. How many values can a Tuple store in C#?
a) A Tuple can store only one value.
b) A Tuple can store up to three values.

c) A Tuple can store up to seven values.
d) A Tuple can store any number of values.

Answer: c) A Tuple can store up to seven

values.

Q34. What is the advantage of using Tuples in C#?
a) Tuples provide better type safety compared to using separate variables to store multiple values.
b) Tuples make it easier to pass multiple values as a single argument to a method or return multiple values from a method.
c) Tuples allow for better performance in memory management.
d) Tuples can only store values of the same data type.

Answer: b) Tuples make it easier to pass multiple values as a single argument to a method or return multiple values from a method.

Q35. How do you access the values stored in a Tuple in C#?
a) By using the "get" keyword followed by the index of the value.
b) By using the "Value" property of the Tuple with the index of the value.
c) By using the "." operator with the index of the value.
d) By using the "Item" property of the Tuple with the index of the value.

Answer: d) By using the "Item" property of the Tuple with the index of the value.

8. Lecture: Delegates, Lambdas, and Error Handling

8.1. Delegates

Q1. What is a delegate in C#?
a) A delegate is a keyword used to define a method.
b) A delegate is a data type used to represent a reference to a method with a specific signature.
c) A delegate is a special type of class used for exception handling.
d) A delegate is a reserved keyword used for defining events.

Answer: b) A delegate is a data type used to represent a reference to a method with a specific signature.

Q2. How do you declare a delegate in C#?
a) delegate MyDelegate;
b) MyDelegate delegate;
c) delegate void MyDelegate();
d) MyDelegate = delegate();

Answer: c) delegate void MyDelegate();

Q3. What is the purpose of a delegate in C#?
a) Delegates are used to store multiple values in a single variable.
b) Delegates are used to define access modifiers for methods.
c) Delegates are used to represent a reference to a method, allowing for method invocation through the delegate.
d) Delegates are used to define custom exception classes.

Answer: c) Delegates are used to represent a reference to a method, allowing for method invocation through the delegate.

Q4. Which of the following statements is true regarding delegates in C#?
a) Delegates can only be used with instance methods.
b) Delegates cannot be used to pass methods as arguments to other methods.
c) Delegates cannot be used to define events.
d) Delegates can be used to invoke multiple methods sequentially.

Answer: d) Delegates can be used to invoke multiple methods sequentially.

Q5. What is the main benefit of using delegates in C#?
a) Delegates allow for better memory management in the application.
b) Delegates eliminate the need for method overloading.
c) Delegates provide a way to achieve better performance in method invocation.
d) Delegates provide more flexibility in method calling, allowing methods to be swapped at runtime.

Answer: d) Delegates provide more flexibility in method calling, allowing methods to be swapped at runtime.

8.1.1. Calling methods using delegates

Q6. How do you call a method using a delegate in C#?
a) By using the "execute" keyword followed by the delegate name.
b) By using the "invoke" keyword followed by the delegate name.
c) By using the "call" keyword followed by the delegate name.
d) By simply using the delegate name as if it were a method.

Answer: b) By using the "invoke" keyword followed by the delegate name.

Q7. What is the purpose of calling methods using delegates in C#?
a) It allows for creating new methods dynamically at runtime.

b) It allows for calling methods from another class without creating an instance of that class.
c) It allows for calling private methods from outside the class.
d) It allows for passing methods as arguments to other methods.

Answer: b) It allows for calling methods from another class without creating an instance of that class.

Q8. Which of the following is a valid example of calling a method using a delegate in C#?
a) delegate void MyDelegate();
   MyDelegate myDelegate = MyMethod;
   myDelegate.invoke();
b) delegate void MyDelegate();
   MyDelegate myDelegate = MyMethod();
   myDelegate.invoke();
c) delegate void MyDelegate();
   MyDelegate myDelegate = MyMethod;
   myDelegate();
d) delegate void MyDelegate();
   MyDelegate myDelegate = MyMethod;
   myDelegate.call();

Answer: c) delegate void MyDelegate();
   MyDelegate myDelegate = MyMethod;
   myDelegate();

8.1.2. Uses of delegates

Q9. What are the main uses of delegates in C#?
a) Delegates are used for defining classes.
b) Delegates are used for defining events.
c) Delegates are used for exception handling.
d) Delegates are used for callback mechanisms and implementing event handling.

Answer: d) Delegates are used for callback mechanisms and implementing event handling.

Q10. How do delegates facilitate callback mechanisms in C#?
a) Delegates allow for creating new methods dynamically at runtime.
b) Delegates allow for invoking methods from another class without creating an instance of that class.
c) Delegates allow for passing methods as arguments to other methods.
d) Delegates allow for calling a method back through a reference to that method.

Answer: d) Delegates allow for calling a method back through a reference to that method.

Q11. Which of the following is a valid example of using delegates for implementing event handling in C#?
a) delegate void MyEventHandler(object sender, EventArgs e);
   event MyEventHandler myEvent;
b) delegate void MyEventHandler();
   event MyEventHandler myEvent;
c) delegate void MyEventHandler(object sender, EventArgs e);
   event MyEventHandler myEvent()
d) delegate void MyEventHandler();
   event MyEventHandler myEvent();

Answer: a) delegate void MyEventHandler(object sender, EventArgs e);
   event MyEventHandler myEvent;

8.1.3. Multicast delegates

Q12. What are multicast delegates in C#?
a) Multicast delegates are delegates that can be used with multiple types of methods.

b) Multicast delegates are delegates that can be used with static methods only.
c) Multicast delegates are delegates that can hold references to multiple methods and invoke them in sequential order.
d) Multicast delegates are delegates that can hold references to multiple methods and invoke them in parallel.

Answer: c) Multicast delegates are delegates that can hold references to multiple methods and invoke them in sequential order.

Q13. How do you create a multicast delegate in C#?
a) By using the "multicast" keyword when declaring the delegate.
b) By using the "+" operator to combine multiple delegates.
c) By using the "combine" method of the delegate class.
d) By using the "new" keyword followed by the delegate names separated by commas.

Answer: b) By using the "+" operator to combine multiple delegates.

Q14. What is the result of invoking a multicast delegate in C# when it has multiple methods attached to it?
a) All methods attached to the delegate are invoked, and their return values are combined into a single value.
b) Only the first method attached to the delegate is invoked.
c) All methods attached to the delegate are invoked in parallel.
d) All methods attached to the delegate are invoked in sequential order.

Answer: d) All methods attached to the delegate are invoked in sequential order.

Q15. Which of the following is a valid example of creating a multicast delegate in C#?
a) delegate void MyDelegate();
   MyDelegate myDelegate = Method1;
   myDelegate = Method2;
   myDelegate = Method3;
b) delegate void MyDelegate();
   MyDelegate myDelegate = Method1;
   myDelegate += Method2;
   myDelegate += Method3;
c) delegate void MyDelegate();
   MyDelegate myDelegate = Method1;
   myDelegate += MyDelegate(Method2);
   myDelegate += new MyDelegate(Method3);
d) delegate void MyDelegate();
   MyDelegate myDelegate = Method1;
   myDelegate.add(Method2);
   myDelegate.add(Method3);

Answer: b) delegate void MyDelegate

();
   MyDelegate myDelegate = Method1;
   myDelegate += Method2;
   myDelegate += Method3;

8.1.4. Action, Func, Predicate delegates

Q16. What are Action, Func, and Predicate delegates in C#?
a) Action, Func, and Predicate are classes used for exception handling.
b) Action, Func, and Predicate are types of multicast delegates.
c) Action, Func, and Predicate are pre-defined delegate types provided by the .NET framework.
d) Action, Func, and Predicate are reserved keywords used for defining events.

Answer: c) Action, Func, and Predicate are pre-defined delegate types provided by the .NET framework.

Q17. What is the purpose of the Action delegate in C#?
a) The Action delegate is used for methods that return a value.
b) The Action delegate is used for methods that do not return a value.
c) The Action delegate is used for methods that take input parameters.
d) The Action delegate is used for methods that have access modifiers.

Answer: b) The Action delegate is used for methods that do not return a value.

Q18. How do you declare an Action delegate in C#?
a) Action MyAction;
b) Action MyAction();
c) Action<int> MyAction;
d) Action MyAction(int);

Answer: c) Action<int> MyAction;

Q19. What is the purpose of the Func delegate in C#?
a) The Func delegate is used for methods that return a value.
b) The Func delegate is used for methods that do not return a value.
c) The Func delegate is used for methods that take input parameters.
d) The Func delegate is used for methods that have access modifiers.

Answer: a) The Func delegate is used for methods that return a value.

Q20. How do you declare a Func delegate in C#?
a) Func MyFunc;
b) Func MyFunc();
c) Func<int> MyFunc;
d) Func MyFunc(int);

Answer: c) Func<int> MyFunc;

8.1.4.Action, Func, Predicate delegates
Q21. What is the purpose of the Predicate delegate in C#?
a) The Predicate delegate is used for methods that return a value.
b) The Predicate delegate is used for methods that do not return a value.
c) The Predicate delegate is used for methods that take input parameters.
d) The Predicate delegate is used for methods that return a boolean value.

Answer: d) The Predicate delegate is used for methods that return a boolean value.

Q22. How do you declare a Predicate delegate in C#?
a) Predicate MyPredicate;
b) Predicate MyPredicate();
c) Predicate<int> MyPredicate;
d) Predicate MyPredicate(int);

Answer: c) Predicate<int> MyPredicate;

Q23. Which of the following is a valid example of using the Action delegate in C#?
a) Action myAction = () => { Console.WriteLine("Hello"); };
b) Action myAction = (string name) => { Console.WriteLine("Hello, " + name); };
c) Action myAction = () => Console.WriteLine("Hello");
d) Action myAction = () => { return "Hello"; };

Answer: c) Action myAction = () => Console.WriteLine("Hello");

Q24. Which of the following is a valid example of using the Func delegate in C#?
a) Func myFunc = () => { Console.WriteLine("Hello"); };
b) Func<int> myFunc = () => 42;
c) Func<string, int> myFunc = (string str) => { return str.Length; };
d) Func myFunc = (string name) => { return "Hello, " + name; };

Answer: c) Func<string, int> myFunc = (string str) => { return str.Length; };

Q25. Which of the following is a valid example of using the Predicate delegate in C#?
a) Predicate myPredicate = (int num) => { return num % 2 == 0; };
b) Predicate<int> myPredicate = (int num) => { return num % 2 == 0; };
c) Predicate myPredicate = (int num) => num % 2 == 0;
d) Predicate<int> myPredicate = (int num) => num % 2 == 0;

Answer: d) Predicate<int> myPredicate = (int num) => num % 2 == 0;

8.2. Anonymous methods

Q26. What is an anonymous method in C#?
a) An anonymous method is a method with no name, defined using the "anonymous" keyword.
b) An anonymous method is a method with a unique identifier, defined using the "anonymous" keyword.
c) An anonymous method is a method with no name, defined using the "delegate" keyword.
d) An anonymous method is a method with a unique identifier, defined using the "delegate" keyword.

Answer: c) An anonymous method is a method with no name, defined using the "delegate" keyword.

Q27. What is the purpose of using anonymous methods in C#?
a) Anonymous methods allow for defining methods with no parameters.
b) Anonymous methods allow for defining methods that return a value.
c) Anonymous methods allow for defining inline code blocks without the need to explicitly define a separate method.
d) Anonymous methods allow for defining methods with access modifiers.

Answer: c) Anonymous methods allow for defining inline code blocks without the need to explicitly define a separate method.

Q28. How do you declare an anonymous method in C#?
a) delegate void MyDelegate();
  MyDelegate myDelegate = delegate() { Console.WriteLine("Hello"); };
b) delegate void MyDelegate();
  MyDelegate myDelegate = () => { Console.WriteLine("Hello"); };
c) delegate void MyDelegate();
  MyDelegate myDelegate = (string name) => { Console.WriteLine("Hello, " + name); };
d) delegate void MyDelegate();
  MyDelegate myDelegate = delegate(string name) { Console.WriteLine("Hello, " + name); };

Answer: b) delegate void MyDelegate();
  MyDelegate myDelegate = () => { Console.WriteLine("Hello"); };

Q29. Which of the following is a valid example of using an anonymous method with parameters in C#?
a) delegate void MyDelegate();
  MyDelegate myDelegate = () => { Console.WriteLine("Hello"); };
b) delegate void MyDelegate();
  MyDelegate myDelegate = (string name) => { Console.WriteLine("Hello, " + name); };
c) delegate void MyDelegate();
  MyDelegate myDelegate = (string name) => Console.WriteLine("Hello, " + name);
d) delegate void MyDelegate();
  MyDelegate myDelegate = delegate(string name) { Console.WriteLine("Hello, " + name); };

Answer: b) delegate void MyDelegate();
MyDelegate myDelegate = (string name) => { Console.WriteLine("Hello, " + name); };

Q30. Can an anonymous method access variables from the enclosing scope in C#?
a) Yes, an anonymous method can access variables from the enclosing scope.
b) No, anonymous methods can only access variables declared inside the method body.
c) Anonymous methods can only access global variables.
d) Anonymous methods cannot access any variables.

Answer: a) Yes, an anonymous method can access variables from the enclosing scope.

8.3. Lambdas

Q31. What are lambdas in C#?
a) Lambdas are a type of collection in C#.
b) Lambdas are a data type used to represent a reference to a method with a specific signature.
c) Lambdas are used to define access modifiers for methods.
d) Lambdas are a concise and expressive way to define inline code blocks.

Answer: d) Lambdas are a concise and expressive way to define inline code blocks.

Q32. What is the syntax for a lambda expression in C#?
a) (parameters) => { code }
b) [parameters] => { code }
c) {parameters} => { code }
d) (parameters) -> { code }

Answer: a) (parameters) => { code }

Q33. How do you declare a lambda expression in C#?
a) lambda MyLambda = (int x) => { return x * 2; };
b) Func MyLambda = (int x) => { return x * 2; };
c) MyLambda lambda = (int x) => { return x * 2; };
d) Func<int, int> MyLambda = (int x) => { return x * 2; };

Answer: d) Func<int, int> MyLambda = (int x) => { return x * 2; };

Q34. What is the main benefit of using lambdas in C#?
a) Lambdas allow for better memory management in the application.
b) Lambdas eliminate the need for method overloading.
c) Lambdas provide a way to achieve better performance in method invocation.
d) Lambdas provide a concise and expressive way to define inline code blocks.

Answer: d) Lambdas provide a concise and expressive way to define inline

code blocks.

Q35. Which of the following is a valid example of using a lambda expression in C#?
a) int result = MyLambda(5);
b) int result = MyLambda.Invoke(5);
c) int result = MyLambda.Execute(5);
d) int result = MyLambda((int x) => { return x * 2; });

Answer: a) int result = MyLambda(5);

8.4. Error Handling (Exceptions Handling)(Checked & Unchecked Statements & The try, catch, finally)
Q36. What is error handling in C#?
a) Error handling is a technique used to prevent errors from occurring in the application.
b) Error handling is a technique used to hide errors from users.

c) Error handling is a mechanism used to detect, respond to, and recover from runtime errors in the application.
d) Error handling is a mechanism used to suppress compiler errors.

Answer: c) Error handling is a mechanism used to detect, respond to, and recover from runtime errors in the application.

Q37. What is an exception in C#?
a) An exception is a type of class used for error handling.
b) An exception is a data type used to represent a reference to a method with a specific signature.
c) An exception is an unexpected or exceptional event that occurs during the execution of a program and disrupts its normal flow.
d) An exception is a reserved keyword used for defining events.

Answer: c) An exception is an unexpected or exceptional event that occurs during the execution of a program and disrupts its normal flow.

Q38. How do you handle exceptions in C#?
a) By using the "error" keyword followed by the exception type.
b) By using the "catch" keyword followed by the exception type.
c) By using the "try" keyword followed by the exception type.
d) By using the "handle" keyword followed by the exception type.

Answer: b) By using the "catch" keyword followed by the exception type.

Q39. What is the purpose of the "try" block in C#?
a) The "try" block is used to handle exceptions and execute the corresponding catch block if an exception occurs.
b) The "try" block is used to define the code that will be executed if an exception occurs.
c) The "try" block is used to suppress compiler errors.
d) The "try" block is used to define access modifiers for methods.

Answer: a) The "try" block is used to handle exceptions and execute the corresponding catch block if an exception occurs.

Q40. Which of the following is a valid example of using the "try-catch" block for exception handling in C#?
a)
```
try {
    // code block
  }
  finally {
    // code block
  }
```
b)
```
try {
    // code block
  }
  catch (Exception ex) {
    // code block
  }
```
c)
```
try {
    // code block
  }
  catch (Exception) {
    // code block
  }
```
d)
```
try {
    // code block
  }
  catch {
    // code block
```

```
}
```

Answer: b)
```
try {
    // code block
}
catch (Exception ex) {
    // code block
}
```

Q41. What is the purpose of the "finally" block in C#?
a) The "finally" block is used to define the code that will always be executed, regardless of whether an exception occurs or not.
b) The "finally" block is used to handle exceptions and execute the corresponding catch block if an exception occurs.
c) The "finally" block is used to define the code that will be executed if an exception occurs.
d) The "finally" block is used to suppress compiler errors.

Answer: a) The "finally" block is used to define the code that will always be executed, regardless of whether an exception occurs or not.

Q42. What is the purpose of the "throw" keyword in C#?
a) The "throw" keyword is used to throw an exception explicitly in a catch block.
b) The "throw" keyword is used to handle exceptions.
c) The "throw" keyword is used to define access modifiers for methods.
d) The "throw" keyword is used to suppress compiler errors.

Answer: a) The "throw" keyword is used to throw an exception explicitly in a catch block.

Q43. Which of the following is a valid example of using the "finally" block for exception handling in C#?
a)
```
try {
    // code block
}
catch (Exception ex) {
    // code block
}
finally {
    // code block
}
```
b)
```
try {
    // code block
}
catch (Exception ex) {
    // code block
}
finally (Exception ex) {
    // code block
}
```
c)
```
try {
    // code block
}
finally {
    // code block
}
catch (Exception ex) {
    // code block
}
```
d)
```
try {
    // code block
}
```

```
  finally {
    // code block
  }
```

Answer: a)
```
try {
    // code block
  }
  catch (Exception ex) {
    // code block
  }
  finally {
    // code block
  }
```

Q44. What are the "checked" and "unchecked" statements used for in C#?
a) The "checked" statement is used to handle checked exceptions, while the "unchecked" statement is used to handle unchecked exceptions.
b) The "checked" statement is used to ensure that arithmetic operations that result in an overflow throw an exception, while the "unchecked" statement suppresses overflow checks.
c) The "checked" and "unchecked" statements are used to suppress compiler errors.
d) The "checked" and "unchecked" statements are used to define access modifiers for methods.

Answer: b) The "checked" statement is used to ensure that arithmetic operations that result in an overflow throw an exception, while the "unchecked" statement suppresses overflow checks.

Q45. What are some best practices for exception handling in C#?
a) Always catch all exceptions using a generic catch block.
b) Rethrow the same exception caught using the "throw;" statement inside a catch block.
c) Use exception handling for normal flow control in the application.
d) Catch specific exceptions and handle them appropriately, and avoid catching generic "System.Exception."

Answer: d) Catch specific exceptions and handle them appropriately, and avoid catching generic "System.Exception."

8.4.1. Dos & Don’ts of Exception Handling

Q46. What are some dos of exception handling in C#?
a) Do use exception handling for normal flow control in the application.
b) Do catch specific exceptions and handle them appropriately.
c) Do catch generic "System.Exception" to handle all exceptions in one place.
d) Do rethrow the same exception caught using the "throw;" statement inside a catch block.

Answer: b) Do catch specific exceptions and handle them appropriately.

Q47. What are some don'ts of exception handling in C#?
a) Don't use exception handling for normal flow control in the application.
b) Don't catch specific exceptions; always catch generic "System.Exception."
c) Don't rethrow the same exception caught using the "throw;" statement inside a catch block.
d) Don't use the "finally" block; it can lead to performance issues.

Answer: a) Don't use exception handling for normal flow control in the application.

Q48. Is it advisable to catch generic "System.Exception" in exception handling?
a) Yes, it is recommended to catch generic "System.Exception" to handle all exceptions in one place.
b) No, catching generic "System.Exception" is considered bad practice as it can hide important error details.
c) Catching generic "System.Exception" is not allowed in C#.
d) Catching generic "System.Exception" will result in a compiler error.

Answer: b) No, catching generic "System.Exception" is considered bad practice as it can hide important error details.

Q49. When should you use the "finally" block in exception handling?
a) The "finally" block is always required after every try-catch block.
b) The "finally" block is optional, but it should be used to release resources, such as closing files or connections.
c) The "finally" block should be used to define access modifiers for methods.
d) The "finally" block is used to suppress compiler errors.

Answer: b) The "finally" block is optional, but it should be used to release resources, such as closing files or connections.

Q50. Which of the following is a recommended approach for handling exceptions in C#?
a) Catch generic "System.Exception" in every catch block.
b) Catch all exceptions using a single generic catch block.
c) Catch specific exceptions and handle them appropriately.
d) Rethrow the same exception caught using the "throw;" statement inside a catch block.

Answer: c) Catch specific exceptions and handle them appropriately.

8.5. User Defined Exception classes

Q51. What are user-defined exception classes in C#?
a) User-defined exception classes are exceptions that are defined by the user using the "try-catch" block.
b) User-defined exception classes are custom exception classes that the user creates by inheriting from the "Exception" base class or other exception classes.
c) User-defined exception classes are exceptions that occur due to errors made by the user while writing code.
d) User-defined exception classes are exceptions that are defined by the user using the "throw" keyword.

Answer: b) User-defined exception classes are custom exception classes that the user creates by inheriting from the "Exception" base class or other exception classes.

Q52. Why would you create a user-defined exception class in C#?
a) To hide exceptions from users.
b) To override built-in exception classes in the .NET framework.
c) To handle generic exceptions more efficiently.
d) To represent specific types of errors that are unique to the application.

Answer: d) To represent specific types of errors that are unique to the application.

Q53. How do you create a user-defined exception class in C#?
a) By using the "exception" keyword followed by the class name.
b) By using the "throw" keyword followed by the class name.
c) By creating a new class that inherits from the "Exception" base class or other exception classes.
d) By using the "catch" keyword followed by the class name.

Answer: c) By creating a new class that inherits from the "Exception" base class or other exception classes.

Q54. When should you throw a user-defined exception in C#?
a) User-defined exceptions should be thrown only in the "finally" block.
b) User-defined exceptions should be thrown in the "try" block to handle specific error scenarios.
c) User-defined exceptions should be thrown in the "catch" block to rethrow caught exceptions.
d) User-defined exceptions should be thrown in the "using" block.

Answer: b) User-defined exceptions should be thrown in the "try" block to handle specific error scenarios.

Q55. How do you throw a user-defined exception in C#?
a) throw MyException;
b) throw new MyException();
c) throw new Exception("MyException");
d) throw MyException();

Answer: b) throw new MyException();

8.6. Declaring and raising events

Q56. What is an event in C#?
a) An event is a data type used to represent a reference to a method with a specific signature.
b) An event is a reserved keyword used for defining exception handling.
c) An event is a mechanism used to notify subscribers when an action occurs.
d) An event is a type of collection in C#.

Answer: c) An event is a mechanism used to notify subscribers when an action occurs.

Q57. What is the purpose of declaring an event in C#?
a) To handle exceptions in the application.
b) To define access modifiers for methods.
c) To define a delegate type that will represent the event.
d) To notify subscribers when a specific action occurs.

Answer: c) To define a delegate type that will represent the event.

Q58. How do you declare an event in C#?
a) event MyEvent;
b) event MyEvent();
c) event EventHandler MyEvent;
d) event MyEvent EventHandler;

Answer: c) event EventHandler MyEvent;

Q59. How do you raise an event in C#?
a) raise MyEvent;
b) raise MyEvent();
c) MyEvent.Raise();
d) MyEvent?.Invoke();

Answer: d) MyEvent?.Invoke();

Q60. What is the purpose of the "event" keyword in C#?
a) The "event" keyword is used to define access modifiers for events.
b) The "event" keyword is used to define the event handler delegate type.
c) The "event" keyword is used to handle exceptions.
d) The "event" keyword is used to suppress compiler errors.

Answer: b) The "event" keyword is used to define the event handler delegate type.

8.7. Handling events

Q61. How do you subscribe to an event in C#?
a) By using the "subscribe" keyword followed by the event name.
b) By using the "+=" operator followed by the event name and the event handler method.
c) By using the "add" keyword followed by the event name and the event handler method.
d) By using the "+=" operator followed by the event name.

Answer: d) By using the "+=" operator followed by the event name.

Q62. What is the purpose of the "+=" operator when subscribing to an event in C#?
a) The "+=" operator is used to handle exceptions in event handling.
b) The "+=" operator is used to suppress compiler errors.
c) The "+=" operator is used to subscribe to an event and attach an event handler method.
d) The "+=" operator is used to define access modifiers for event handlers.

Answer: c) The "+=" operator is used to subscribe to an event and attach an event handler method.

Q63. How do you unsubscribe from an event in C#?
a) By using the "-=" operator followed by the event name and the event handler method.
b) By using the "unsubscribe" keyword followed by the event name.
c) By using the "remove" keyword followed by the event name and the event handler method.
d) By using the "-=" operator followed by the event name.

Answer: a) By using the "-=" operator followed by the event name and the event handler method.

Q64. What is the purpose of the "-=" operator when unsubscribing from an event in C#?
a) The "-=" operator is used to handle exceptions in event handling.
b) The "-=" operator is used to suppress compiler errors.
c) The "-=" operator is used to unsubscribe from an event and detach an event handler method.
d) The "-=" operator is used to define access modifiers for event handlers.

Answer: c) The "-=" operator is used to unsubscribe from an event and detach an event handler method.

Q65. Can multiple event handlers be attached to a single event in C#?
a) No, only one event handler can be attached to a single event.
b) Yes, multiple event handlers can be attached to a single event, and they will be called in the order they were subscribed.
c) Yes, multiple event handlers can be attached to a single event, and they will be called concurrently.
d) No, attaching multiple event handlers to a single event will result in a compiler error.

Answer: b) Yes, multiple event handlers can be attached to a single event, and they will be called in the order they were subscribed.

8.8. Async calls using delegates

Q66. What are async calls using delegates in C#?
a) Async calls are used to handle exceptions in C#.
b) Async calls are used to define access modifiers for methods.
c) Async calls are used to make synchronous method calls.
d) Async calls are used to execute methods asynchronously using delegates.

Answer: d) Async calls are used to execute methods asynchronously using delegates.

Q67. How do you make an async call using delegates in C#?
a) By using the "async" keyword followed by the delegate name.
b) By using the "await" keyword followed by the delegate name.
c) By using the "async" keyword followed by the "await" keyword and the delegate name.
d) By using the "delegate" keyword followed by the delegate name.

Answer: c) By using the "async" keyword followed by the "await" keyword and the delegate name.

Q68. What is the purpose of the "async" keyword in C#?
a) The "async" keyword is used to define access modifiers for methods.
b) The "async" keyword is used to suppress compiler errors.
c) The "async" keyword is used to handle exceptions.
d) The "async" keyword is used to indicate that a method contains asynchronous code.

Answer: d) The "async" keyword is used to indicate that a method contains asynchronous code.

Q69. What is the purpose of the "await" keyword in C#?
a) The "await" keyword is used to define access modifiers for methods.
b) The "await" keyword is used to suppress compiler errors.
c) The "await" keyword is used to handle exceptions.
d) The "await" keyword is used to asynchronously wait for the result of an async method call.

Answer: d) The "await" keyword is used to asynchronously wait for the result of an async method call.

Q70. Can any method be marked as "async" in C#?
a) Yes, any method can be marked as "async" in C#.
b) No, only methods that return Task or Task<T> can be marked as "async" in C#.
c) No, only methods that return void can be marked as "async" in C#.
d) Yes, any method that contains a delegate can be marked as "async" in C#.

Answer: b) No, only methods that return Task or Task<T> can be marked as "async" in C#.

9. Lecture: LINQ and Threading

9.1. Anonymous types

Q1. What is an anonymous type in C#?
a) A type that is not named and can be used for temporary storage of data.
b) A type that is used for creating objects with anonymous properties.
c) A type that is used for declaring variables without specifying their data types.
d) A type that is used for creating classes without specifying their names.

Answer: b) A type that is used for creating objects with anonymous properties.

Q2. How do you create an anonymous type in C#?
a) var anonymousType = new { Property1 = value1, Property2 = value2 };
b) var anonymousType = new AnonymousType { Property1 = value1, Property2 = value2 };
c) var anonymousType = { Property1 = value1, Property2 = value2 };
d) var anonymousType = { value1, value2 };

Answer: a) var anonymousType = new { Property1 = value1, Property2 = value2 };

Q3. Can anonymous types have methods in C#?
a) Yes, anonymous types can have methods.
b) No, anonymous types cannot have methods.
c) Anonymous types can have methods, but they must be declared as static.
d) Anonymous types can have methods, but they must be declared as private.

Answer: b) No, anonymous types cannot have methods.

Q4. What is the scope of an anonymous type in C#?
a) The scope of an anonymous type is limited to the method or block where it is defined.
b) The scope of an anonymous type is global, and it can be accessed from anywhere in the application.
c) The scope of an anonymous type is limited to the class where it is defined.
d) The scope of an anonymous type is limited to the namespace where it is defined.

Answer: a) The scope of an anonymous type is limited to the method or block where it is defined.

Q5. Can you explicitly define the data type of properties in an anonymous type in C#?
a) Yes, you can explicitly define the data type of properties in an anonymous type.
b) No, the data types of properties in an anonymous type are inferred automatically by the compiler.
c) Yes, but you can only define value types as property data types in an anonymous type.
d) No, properties in an anonymous type must always have the "var" data type.

Answer: b) No, the data types of properties in an anonymous type are inferred automatically by the compiler.

9.2. Extension methods

Q6. What are extension methods in C#?
a) Extension methods are methods that extend the functionality of value types.
b) Extension methods are methods that can be used to extend the functionality of interfaces.
c) Extension methods are methods that allow you to add new methods to existing types without modifying their original code.
d) Extension methods are methods that are marked as "static" and can only be called from within the class where they are defined.

Answer: c) Extension methods are methods that allow you to add new methods to existing types without modifying their original code.

Q7. How do you define an extension method in C#?

a) Extension methods are defined within a static class, and the first parameter of the method is marked with the "this" keyword followed by the type to be extended.
b) Extension methods are defined within an abstract class, and the first parameter of the method is marked with the "this" keyword followed by the type to be extended.
c) Extension methods are defined within a sealed class, and the first parameter of the method is marked with the "this" keyword followed by the type to be extended.
d) Extension methods are defined within an interface, and the first parameter of the method is marked with the "this" keyword followed by the type to be extended.

Answer: a) Extension methods are defined within a static class, and the first parameter of the method is marked with the "this" keyword followed by the type to be extended.

Q8. Can extension methods access private members of the type they are extending in C#?
a) Yes, extension methods can access private members of the type they are extending.
b) No, extension methods can only access public members of the type they are extending.
c) Extension methods cannot access any members of the type they are extending.
d) Extension methods can access private members, but only if the extension method is declared within the same class as the type being extended.

Answer: b) No, extension methods can only access public members of the type they are extending.

Q9. How do you call an extension method in C#?
a) By using the "this" keyword followed by the name of the extension method.
b) By using the "this" keyword followed by the name of the type being extended, a dot, and then the name of the extension method.
c) By using the "extension" keyword followed by the name of the extension method.
d) By using the "extends" keyword followed by the name of the extension method.

Answer: b) By using the "this" keyword followed by the name of the type being extended, a dot, and then the name of the extension method.

Q10. What is the purpose of extension methods in C#?
a) To encapsulate private methods within a class.
b) To override existing methods in a class.
c) To add new methods to existing types without modifying their original code.
d) To provide a way to call methods from other classes without importing namespaces.

Answer: c) To add new methods to existing types without modifying their original code.

9.3. Partial classes

Q11. What are partial classes in C#?
a) Partial classes are classes that can only be used within a single assembly.
b) Partial classes are classes that are marked as "partial" and can be divided into multiple parts, each defined in a separate file.
c) Partial classes are classes that have only partial implementations and cannot be instantiated directly.
d) Partial classes are classes that are defined with multiple constructors.

Answer: b) Partial classes are classes that are marked as "partial" and can be divided into multiple parts, each defined in a separate file.

Q12. How do you define a partial class in C#?
a) By using the "partial" keyword followed by the class name and its definition.
b) By using the "class" keyword followed by the class name and its definition.
c) By using the "partial" keyword before the "class" keyword and then defining the class.
d) By using the "partial" keyword before each method of the class.

Answer: c) By using the "partial" keyword before the "class" keyword and then defining the class.

Q13. Can a partial class have multiple definitions within the same file in C#?

a) Yes, a partial class can have multiple definitions within the same file.
b) No, a partial class can have only one definition in a single file.
c) Partial classes cannot be defined within the same file.
d) A partial class can have multiple definitions, but they must be in different namespaces.

Answer: a) Yes, a partial class can have multiple definitions within the same file.

Q14. How does C# handle the merging of multiple partial class definitions?
a) C# automatically merges the definitions based on the order of

appearance in the files.
b) C# automatically merges the definitions based on the alphabetical order of the files.
c) C# requires the developer to manually merge the definitions.
d) C# does not allow multiple partial class definitions in the same file.

Answer: a) C# automatically merges the definitions based on the order of appearance in the files.

Q15. What is the main benefit of using partial classes in C#?
a) Partial classes allow you to define a class with multiple constructors.
b) Partial classes make the code cleaner and more organized by dividing the class implementation into multiple files.
c) Partial classes allow you to define the same class in multiple namespaces.
d) Partial classes allow you to create classes that cannot be instantiated.

Answer: b) Partial classes make the code cleaner and more organized by dividing the class implementation into multiple files.

9.4. Partial methods

Q16. What are partial methods in C#?
a) Partial methods are methods that can only be used within a single assembly.
b) Partial methods are methods that are marked as "partial" and can be divided into multiple parts, each defined in a separate file.
c) Partial methods are methods that have only partial implementations and must be completed in a separate part of the partial class.
d) Partial methods are methods that are defined with multiple parameters.

Answer: c) Partial methods are methods that have only partial implementations and must be completed in a separate part of the partial class.

Q17. How do you define a partial method in C#?
a) By using the "partial" keyword before the "method" keyword and then defining the method.
b) By using the "partial" keyword before the method name and then defining the method.
c) By using the "partial" keyword followed by the method name and its definition.
d) By using the "partial" keyword followed by the class name and the method name.

Answer: b) By using the "partial" keyword before the method name and then defining the method.

Q18. Can a partial method have multiple definitions within the same file in C#?
a) Yes, a partial method can have multiple definitions within the same file.
b) No, a partial method can have only one definition in a single file.
c) Partial methods cannot be defined within the same file.
d) A partial method can have multiple definitions, but they must be in different namespaces.

Answer: a) Yes, a partial method can have multiple definitions within the same file.

Q19. How does C# handle the merging of multiple partial method definitions?
a) C# automatically merges the definitions based on the order of appearance in the files.
b) C# automatically merges the definitions based on the alphabetical order of the files.
c) C# requires the developer to manually merge the definitions.

d) C# does not allow multiple partial method definitions in the same file.

Answer: a) C# automatically merges the definitions based on the order of appearance in the files.

Q20. What is the main purpose of using partial methods in C#?
a) Partial methods allow you to define a method with multiple parameters.
b) Partial methods make the code cleaner and more organized by dividing the method implementation into multiple files.
c) Partial methods allow you to define a method with multiple return values.
d) Partial methods allow you to define a method that may or may not be implemented in a separate part of the partial class.

Answer: d) Partial methods allow you to define a method that may or may not be implemented in a separate part of the partial class.

9.5. LINQ to objects

Q21. What is LINQ in C#?
a) LINQ is a language-independent query standard for accessing databases.
b) LINQ is a set of extension methods for querying data in C# collections and other data sources.
c) LINQ is a data access technology that allows direct access to databases without using SQL.
d) LINQ is a built-in data type in C#.

Answer: b) LINQ is a set of extension methods for querying data in C# collections and other data sources.

Q22. What does "LINQ to objects" refer to in C#?
a) LINQ to objects refers to using LINQ to query data from C# objects and collections.
b) LINQ to objects refers to using LINQ to query data from databases.
c) LINQ to objects refers to using LINQ to query data from web services.
d) LINQ to objects refers to using LINQ to query data from XML files.

Answer: a) LINQ to objects refers to using LINQ to query data from C# objects and collections.

Q23. Which namespace contains the LINQ extension methods in C#?
a) System.Data
b) System.Linq
c) System.Collections
d) System.IO

Answer: b) System.Linq

Q24. How do you use LINQ to objects in C#?
a) By importing the System.Linq namespace and then calling LINQ extension methods on C# objects or collections.
b) By importing the System.Data namespace and then calling LINQ extension methods on C# objects or collections.
c) By importing the System.Collections namespace and then calling LINQ extension methods on C# objects or collections.
d) By importing the System.XML namespace and then calling LINQ extension methods on C# objects or collections.

Answer: a) By importing the System.Linq namespace and then calling LINQ extension methods on C# objects or collections.

Q25. Which LINQ extension method is used to filter data based on a specified condition in C#?
a) Select
b) GroupBy
c) Where
d) OrderBy

Answer: c) Where

9.6. Writing LINQ queries

Q26. How do you write a LINQ query in C#?
a) By using SQL-like syntax within a string.
b) By using the "query" keyword followed by the LINQ query expression.
c) By using the "from" keyword followed by the data source, "select" keyword, and the query expression.
d) By using the "Linq" class and calling the "Query" method with the query expression as an argument.

Answer: c) By using the "from" keyword followed by the data source, "select" keyword, and the query expression.

Q27. Which of the following statements about LINQ queries in C# is true?
a) LINQ queries are executed immediately when they are written.
b) LINQ queries are executed asynchronously by default.
c) LINQ queries are not executed until the data is needed, which allows for deferred execution.
d) LINQ queries are executed in reverse order.

Answer: c) LINQ queries are not executed until the data is needed, which allows for deferred execution.

Q28. What is the purpose of the "select" keyword in a LINQ query in C#?
a) The "select" keyword is used to filter data based on a specified condition.
b) The "select" keyword is used to define the data source for the query.
c) The "select" keyword is used to specify the result projection of the query.
d) The "select" keyword is used to order the data in ascending or descending order.

Answer: c) The "select" keyword is used to specify the result projection of the query.

Q29. Which LINQ extension method is used to sort data in ascending order in C#?
a) Select
b) GroupBy

c) Where
d) OrderBy

Answer: d) OrderBy

Q30. Which of the following statements about LINQ queries in C# is true?
a) LINQ queries support intellisense and compile-time checking.
b) LINQ queries can only be used with databases and not with in-memory collections.
c) LINQ queries cannot be used with custom objects and types.
d) LINQ queries can only be written using SQL-like syntax.

Answer: a) LINQ queries support intellisense and compile-time checking.

9.7. Deferred execution

Q31. What is deferred execution in LINQ?
a) Deferred execution is the process of executing a LINQ query immediately when it is defined.
b) Deferred execution is the process of delaying the execution of a LINQ query until the data is actually needed.
c) Deferred execution is the process of executing a LINQ query asynchronously.
d) Deferred execution is the process of executing a LINQ query in reverse order.

Answer: b) Deferred execution is the process of delaying the execution of a LINQ query until the data is actually needed.

Q32. How does deferred execution work in C#?
a) When a LINQ query is defined, the query is immediately executed and the results are stored in memory.
b) When a LINQ query is defined, the query is converted into an expression tree, and the actual execution is deferred until the data is requested or enumerated.
c) When a LINQ query is defined, the query is executed asynchronously and the results are stored in memory.
d) When a LINQ query is defined, the query is executed in reverse order.

Answer: b) When a LINQ query is defined, the query is converted into an expression tree, and the actual execution is deferred until the data is requested or enumerated.

Q33. What is the benefit of deferred execution in LINQ?
a) Deferred execution allows for immediate execution of LINQ queries, reducing the need for data caching.
b) Deferred execution allows for better performance and optimization by executing the query only when the data is actually needed.
c) Deferred execution allows for faster execution of LINQ queries by parallelizing the operations.
d) Deferred execution allows for querying data from remote sources without having to download the entire dataset.

Answer: b) Deferred execution allows for better performance and optimization by executing the query only when the data is actually needed.

Q34. How can you force the execution of a deferred LINQ query in C#?
a) By using the "execute" keyword before the LINQ query expression.
b) By calling the "ToList" or "ToArray" method on the LINQ query.
c) By calling the "execute" method on the LINQ query object.
d) By using the "deferred" keyword before the LINQ query expression.

Answer: b) By calling the "ToList" or "ToArray" method on the LINQ query.

Q35. Which of the following LINQ methods triggers immediate execution of a deferred LINQ query in C#?
a) Select
b) Where
c) OrderBy
d) ToList

Answer: d) ToList

9.8. LINQ methods

Q36. Which of the following LINQ extension methods is used to filter data based on a specified condition in C#?
a) Select
b) GroupBy
c) Where
d) OrderBy

Answer: c) Where

Q37. Which of the following LINQ extension methods is used to transform data in C#?
a) Select
b) GroupBy
c) Where
d) OrderBy

Answer: a) Select

Q38. Which of the following LINQ extension methods is used to sort data in ascending order in C#?
a) Select
b) GroupBy
c) Where
d) OrderBy

Answer: d) OrderBy

Q39. Which of the following LINQ extension methods is used to group data based on a specified key in C#?
a) Select
b) GroupBy
c) Where
d) OrderBy

Answer: b) GroupBy

Q40. Which of the following LINQ extension methods is used to join two data sources based on a common property in C#?
a) Select
b) GroupBy
c) Where
d) Join

Answer: d) Join

9.9. PLINQ

Q41. What is PLINQ in C#?
a) PLINQ is a language-independent query standard for accessing databases.
b) PLINQ is a set of extension methods for querying data in C# collections and other data sources.
c) PLINQ is a data access technology that allows direct access to databases without using SQL.
d) PLINQ is a parallel version of LINQ that allows for query operations to be performed concurrently on multiple processors.

Answer: d) PLINQ is a parallel version of LINQ that allows for query operations to be performed concurrently on multiple processors.

Q42. How do you use PLINQ in C#?
a) By importing the System.Linq namespace and then calling PLINQ extension methods on C# objects or collections.
b) By importing the System.Threading.Tasks namespace and then calling PLINQ extension methods on C# objects or collections.
c) By importing the System.Parallel namespace and then calling PLINQ extension methods on C# objects or collections.
d) By importing the System.PLINQ namespace and then calling PLINQ extension methods on C# objects or collections.

Answer: a) By importing the System.Linq namespace and then calling PLINQ extension methods on C# objects or collections.

Q43. Which of the following statements about PLINQ in C# is true?
a) PLINQ can only be used with in-memory collections and not with databases.
b) PLINQ queries always execute sequentially on a single processor.
c) PLINQ queries are executed concurrently on multiple processors, allowing for parallel processing of data.
d) PLINQ queries are not supported in C#.

Answer: c) PLINQ queries are executed concurrently on multiple processors, allowing for parallel processing of data.

Q44. What is the purpose of using PLINQ in C#?
a) To enable direct access to databases without using SQL.
b) To perform query operations on data in parallel to achieve better performance on multi-core processors.
c) To perform query operations on data in reverse order.
d) To enable querying data from web services using LINQ.

Answer: b) To perform query operations on data in parallel to achieve better performance on multi-core processors.

Q45. Which of the following LINQ methods is not supported in PLINQ?
a) Where
b) Select
c) OrderBy
d) GroupBy

Answer: d) GroupBy

9.10. Threading

Q46. What is threading in C#?
a) Threading is a technique used to optimize LINQ queries for better performance.
b) Threading is a technique used to execute multiple tasks concurrently, allowing for better utilization of multi-core processors.
c) Threading is a technique used to query data from databases using LINQ.
d) Threading is a technique used to execute LINQ queries asynchronously.

Answer: b) Threading is a technique used to execute multiple tasks concurrently, allowing for better utilization of multi-core processors.

Q47. What is a thread in C#?
a) A thread is a part of a

CPU.
b) A thread is a process that runs independently.
c) A thread is a lightweight unit of a process that can execute independently.
d) A thread is a data structure used to store temporary data.

Answer: c) A thread is a lightweight unit of a process that can execute independently.

Q48. How do you create a new thread in C#?
a) By using the "new" keyword followed by the thread class name.
b) By using the "ThreadStart" delegate to define the method to be executed by the thread.
c) By using the "new" keyword followed by the thread method name.
d) By using the "Thread" keyword followed by the method to be executed by the thread.

Answer: b) By using the "ThreadStart" delegate to define the method to be executed by the thread.

Q49. What is the purpose of using threads in C#?
a) To perform query operations on data in parallel using PLINQ.
b) To execute multiple tasks concurrently to achieve better performance on multi-core processors.
c) To create new LINQ queries.
d) To execute LINQ queries asynchronously.

Answer: b) To execute multiple tasks concurrently to achieve better performance on multi-core processors.

Q50. How do you start a thread in C#?
a) By calling the "Start" method on the thread object.
b) By calling the "Run" method on the thread object.
c) By calling the "Execute" method on the thread object.
d) By calling the "Begin" method on the thread object.

Answer: a) By calling the "Start" method on the thread object.

9.10.1. ThreadStart, Parameterized ThreadStart

Q51. What is the purpose of the ThreadStart delegate in C#?
a) The ThreadStart delegate is used to define methods that return a value.
b) The ThreadStart delegate is used to define methods that accept parameters.
c) The ThreadStart delegate is used to define methods that can be executed by a new thread.
d) The ThreadStart delegate is used to define methods that perform query operations.

Answer: c) The ThreadStart delegate is used to define methods that can be executed by a new thread.

Q52. How do you use the ThreadStart delegate to define a method that can be executed by a new thread in C#?
a) By using the "ThreadStart" keyword followed by the method name.
b) By using the "ThreadStart" keyword followed by the "new" keyword and the method name.
c) By using the "ThreadStart" keyword followed by the "delegate" keyword and the method name.
d) By using the "ThreadStart" keyword followed by the "void" keyword and the method name.

Answer: c) By using the "ThreadStart" keyword followed by the "delegate" keyword and the method name.

Q53. What is the purpose of the Parameterized ThreadStart delegate in C#?
a) The Parameterized ThreadStart delegate is used to define methods that return a value.
b) The Parameterized ThreadStart delegate is used to define methods that do not accept parameters.
c) The Parameterized ThreadStart delegate is used to define methods that can be executed by a new thread and accept parameters.
d) The Parameterized ThreadStart delegate is used to define methods that perform query operations.

Answer: c) The Parameterized ThreadStart delegate is used to define methods that can be executed by a new thread and accept parameters.

Q54. How do you use the Parameterized ThreadStart delegate to define a method that can be executed by a new thread and accepts parameters in C#?
a) By using the "Parameterized ThreadStart" keyword followed by the method name.
b) By using the "Parameterized ThreadStart" keyword followed by the "new" keyword and the method name.
c) By using the "Parameterized ThreadStart" keyword followed by the "delegate" keyword and the method name.
d) By using the "Parameterized ThreadStart" keyword followed by the "void" keyword and the method name.

Answer: c) By using the "Parameterized ThreadStart" keyword followed by the "delegate" keyword and the method name.

Q55. How do you start a thread with the ThreadStart delegate in C#?
a) By passing the ThreadStart delegate to the Thread constructor.
b) By calling the "Start" method on the ThreadStart delegate.
c) By calling the "Run" method on the ThreadStart delegate.
d) By passing the ThreadStart delegate to the "Execute" method.

Answer: a) By passing the ThreadStart delegate to the Thread constructor.

9.10.2. ThreadPool

Q56. What is the ThreadPool in C#?
a) The ThreadPool is a collection of threads that are used to execute methods asynchronously.
b) The ThreadPool is a class that allows you to define and manage new threads.
c) The ThreadPool is a data structure used to store temporary data.
d) The ThreadPool is a process that runs independently.

Answer: a) The ThreadPool is a collection of threads that are used to execute methods asynchronously.

Q57. How do you use the ThreadPool in C#?
a) By importing the System.Threading namespace and then calling ThreadPool methods.
b) By importing the System.Threading.Tasks namespace and then calling ThreadPool methods.
c) By importing the System.ThreadPool namespace and then calling ThreadPool methods.
d) By importing the System.Pool namespace and then calling ThreadPool methods.

Answer: a) By importing the System.Threading namespace and then calling ThreadPool methods.

Q58. What is the advantage of using the ThreadPool in C#?
a) The ThreadPool allows you to create new threads without any overhead.
b) The ThreadPool provides a more efficient way to manage threads by reusing existing threads from a pool.
c) The ThreadPool allows you to execute LINQ queries in parallel.
d) The ThreadPool automatically handles exceptions that occur during thread execution.

Answer: b) The ThreadPool provides a more efficient way to manage threads by reusing existing threads from a pool.

Q59. How do you queue a method to be executed by the ThreadPool in C#?
a) By calling the "Start" method on the ThreadPool class with the method to be executed as an argument.
b) By calling the "Run" method on the ThreadPool class with the method to be executed as an argument.
c) By using the "QueueUserWorkItem" method of the ThreadPool class with the method to be executed as an argument.
d) By using the "Enqueue" method of the ThreadPool class with the method to be executed as an argument.

Answer: c) By using the "QueueUserWorkItem" method of the ThreadPool class with the method to be executed as an argument.

Q60. How do you set the maximum number of threads in the ThreadPool in C#?
a) By using the "SetMaxThreads" method of the ThreadPool class with the maximum number of threads as an argument.
b) By using the "SetThreadPoolSize" method of the ThreadPool class with the maximum number of threads as an argument.
c) By using the "MaxThreads" property of the ThreadPool class and assigning it the maximum number of threads.
d) By using the "ThreadPoolSize" property of the ThreadPool class and assigning it the maximum number of threads.

Answer: a) By using the "SetMaxThreads" method of the ThreadPool class with the maximum number of threads as an argument.

9.10.3. Synchronizing critical data using lock, Monitor, and Interlocked

Q61. What is synchronization in threading?
a) Synchronization is the process of executing multiple threads concurrently on multiple processors.

b) Synchronization is the process of preventing multiple threads from accessing shared resources simultaneously.
c) Synchronization is the process of executing multiple tasks concurrently to achieve better performance on multi-core processors.
d) Synchronization is the process of ensuring that threads execute in reverse order.

Answer: b) Synchronization is the process of preventing multiple threads from accessing shared resources simultaneously.

Q62. What is a critical section in threading?
a) A critical section is a section of code that can be executed by multiple threads simultaneously without causing any issues.
b) A critical section is a section of code that must be executed by a single thread at a time to avoid data corruption and other problems.
c) A critical section is a section of code that is executed using the ThreadPool.
d) A critical section is a section of code that performs query operations on data.

Answer: b) A critical section is a section of code that must be executed by a single thread at a time to avoid data corruption and other problems.

Q63. How do you synchronize a critical section in C#?
a) By using the "synchronized" keyword before the critical section.
b) By using the "lock" keyword followed by an object that serves as a lock for the critical section.
c) By using the "synchronized" method of the Thread class.
d) By using the "lock" method of the ThreadPool class.

Answer: b) By using the "lock" keyword followed by an object that serves as a lock for the critical section.

Q64. What is the purpose of the "lock" keyword in C#?
a) The "lock" keyword is used to execute LINQ queries in parallel using PLINQ.
b) The "lock" keyword is used to define a method that can be executed by a new thread.
c) The "lock" keyword is used to synchronize a critical section to prevent multiple threads from accessing shared resources simultaneously.
d) The "lock" keyword is used to execute methods asynchronously using the ThreadPool.

Answer: c) The "lock" keyword is used to synchronize a critical section to prevent multiple threads from accessing shared resources simultaneously.

Q65. What is the Interlocked class used for in C#?
a) The Interlocked class is used for parallelizing LINQ queries using PLINQ.
b) The Interlocked class is used for defining and managing new threads.
c) The Interlocked class is used for synchronizing access to shared resources without using locks.
d) The Interlocked class is used for defining and raising events.

Answer: c) The Interlocked class is used for synchronizing access to shared resources without using locks.

10. Lecture: Introduction to Asp.Net MVC

10.1. Architecture of an ASP .Net MVC application

Q1. What is the primary architectural pattern used in an ASP .Net MVC application?
a) Model-View-Controller (MVC)
b) Model-View-ViewModel (MVVM)
c) Model-View-Presenter (MVP)
d) Model-View-Adapter (MVA)

Answer: a) Model-View-Controller (MVC)

Q2. In the MVC pattern, what does the "Model" represent?
a) The user interface components
b) The data and business logic of the application
c) The controller that handles user input
d) The view that displays the data

Answer: b) The data and business logic of the application

Q3. In the MVC pattern, what does the "View" represent?
a) The user interface components
b) The data and business logic of the application
c) The controller that handles user input
d) The model that contains the data

Answer: a) The user interface components

Q4. In the MVC pattern, what does the "Controller" represent?
a) The user interface components
b) The data and business logic of the application
c) The controller that handles user input
d) The model that contains the data

Answer: c) The controller that handles user input

Q5. What is the primary advantage of using the MVC pattern in ASP .Net applications?
a) It simplifies the development process by eliminating the need for a database.
b) It allows for the separation of concerns, making the application more maintainable and testable.
c) It provides a user-friendly interface for designing web pages.
d) It enables real-time communication between the client and the server.

Answer: b) It allows for the separation of concerns, making the application more maintainable and testable.

10.2. Understanding Folder structures and configuration files

Q6. In an ASP .Net MVC application, where are the views typically stored?
a) In the "Controllers" folder
b) In the "Models" folder
c) In the "Views" folder
d) In the "App_Data" folder

Answer: c) In the "Views" folder

Q7. What is the purpose of the "web.config" file in an ASP .Net MVC application?
a) It stores the HTML templates for the views.
b) It contains the configuration settings for the application, such as database connection strings and application-wide settings.

c) It defines the routes for the application.
d) It handles user authentication and authorization.

Answer: b) It contains the configuration settings for the application, such as database connection strings and application-wide settings.

Q8. Where is the "Global.asax" file located in an ASP .Net MVC application?
a) In the "Views" folder
b) In the "Controllers" folder
c) In the "Models" folder
d) In the root directory of the application

Answer: d) In the root directory of the application

Q9. What is the purpose of the "Global.asax" file in an ASP .Net MVC application?
a) It contains the HTML templates for the views.
b) It defines the routes for the application.
c) It handles user authentication and authorization.
d) It contains application-level events and settings.

Answer: d) It contains application-level events and settings.

Q10. In an ASP .Net MVC application, where are the controller classes typically stored?
a) In the "Controllers" folder
b) In the "Views" folder
c) In the "Models" folder
d) In the "App_Code" folder

Answer: a) In the "Controllers" folder

10.3. Understanding Controllers and Actions

Q11. What is the role of a controller in an ASP .Net MVC application?
a) To define the layout and design of the web pages.
b) To handle user input and interact with the model and view.
c) To store and manage data for the application.
d) To handle user authentication and authorization.

Answer: b) To handle user input and interact with the model and view.

Q12. In the context of ASP .Net MVC, what is an "action"?
a) An action is a user interface component that the user interacts with.
b) An action is a method in a controller that handles a user request and returns a response.
c) An action is a data structure used to store temporary data.
d) An action is a view template used to display data to the user.

Answer: b) An action is a method in a controller that handles a user request and returns a response.

Q13. How are actions in a controller associated with specific URLs in ASP .Net MVC?
a) Actions are automatically associated with URLs based on their method names.
b) Actions are explicitly mapped to URLs in the "web.config" file.
c) Actions are automatically associated with URLs based on their parameters.
d) Actions are explicitly mapped to URLs in the "Global.asax" file.

Answer: d) Actions are explicitly mapped to URLs in the "Global.asax" file.

Q14. In an ASP .Net MVC application, how do you pass data from a controller action to a view?
a) By using the "return" statement to return the data from the action.
b) By using the "ViewData" dictionary or the "ViewBag" property.
c) By using the "TempData" dictionary.

d) By using the "Session" object.

Answer: b) By using the "ViewData" dictionary or the "ViewBag" property.

Q15. In an ASP .Net MVC application, how do you pass data from a view to a controller action?
a) By using the "return" statement to return the data from the view.
b) By using the "ViewData" dictionary or the "ViewBag" property.
c) By using the "TempData" dictionary.
d) By using HTML forms or query parameters.

Answer: d) By using HTML forms or query parameters.

10.4. Understanding Views & Models

Q16. What is the role of a view in an ASP .Net MVC application?
a) To handle user input and interact with the model and controller.
b) To define the layout and design of the web pages.
c) To store and manage data for the application.
d) To handle user authentication and authorization.

Answer: b) To define the layout and design of the web pages.

Q17. In the context of ASP .Net MVC, what is a "model"?
a) A model is a user interface component that the user interacts with.
b) A model is a method in a controller that handles a user request and returns a response.
c) A model is a data structure used to store temporary data.
d) A model is a data structure that represents the data and business logic of the application.

Answer: d) A model is a data structure that represents the data and business logic of the application.

Q18. How is a model passed from a controller action to a view in ASP .Net MVC?
a) By using the "return" statement to return the model from the action.
b) By using the "ViewData" dictionary or the "ViewBag" property.
c) By using the "TempData" dictionary.
d) By

passing the model as a parameter to the "View" method.

Answer: d) By passing the model as a parameter to the "View" method.

Q19. What is the purpose of the "ViewBag" property in ASP .Net MVC?
a) It is used to store and manage data in a view.
b) It is used to store and manage data in a controller.
c) It is used to store and manage data in a model.
d) It is used to store and manage data in the "web.config" file.

Answer: a) It is used to store and manage data in a view.

Q20. In an ASP .Net MVC application, how do you display data from the model in a view?
a) By using the "ViewData" dictionary or the "ViewBag" property.
b) By using the "TempData" dictionary.
c) By using HTML forms or query parameters.
d) By using server-side code or Razor syntax.

Answer: d) By using server-side code or Razor syntax.

10.5. MVC State Management

Q21. What is state management in an ASP .Net MVC application?
a) State management is the process of managing the layout and design of the web pages.

b) State management is the process of managing the data and business logic of the application.
c) State management is the process of managing the user's interactions with the application over time.
d) State management is the process of managing the user's authentication and authorization.

Answer: c) State management is the process of managing the user's interactions with the application over time.

Q22. Which of the following is NOT a method of state management in ASP .Net MVC?
a) View state
b) Session state
c) Application state
d) Cookie state

Answer: a) View state

Q23. What is session state in an ASP .Net MVC application?
a) Session state is a method of storing data in the client's browser.
b) Session state is a method of storing data on the server that is accessible across multiple requests from the same client.
c) Session state is a method of storing data in a centralized database.
d) Session state is a method of storing data in the "web.config" file.

Answer: b) Session state is a method of storing data on the server that is accessible across multiple requests from the same client.

Q24. What is application state in an ASP .Net MVC application?
a) Application state is a method of storing data in the client's browser.
b) Application state is a method of storing data on the server that is accessible across multiple requests from different clients.
c) Application state is a method of storing data in a centralized database.
d) Application state is a method of storing data in the "web.config" file.

Answer: b) Application state is a method of storing data on the server that is accessible across multiple requests from different clients.

Q25. What is the purpose of cookies in an ASP .Net MVC application?
a) Cookies are used to store data on the server.
b) Cookies are used to store data on the client's browser.
c) Cookies are used to define the routes for the application.
d) Cookies are used to handle user authentication and authorization.

Answer: b) Cookies are used to store data on the client's browser.

10.6. MVC Module

Q26. What is an MVC module in an ASP .Net MVC application?
a) An MVC module is a user interface component that the user interacts with.
b) An MVC module is a method in a controller that handles a user request and returns a response.
c) An MVC module is a data structure used to store temporary data.
d) An MVC module is a self-contained unit of functionality in an MVC application.

Answer: d) An MVC module is a self-contained unit of functionality in an MVC application.

Q27. What is the purpose of an MVC module in an ASP .Net MVC application?
a) To define the layout and design of the web pages.
b) To handle user input and interact with the model and controller.
c) To store and manage data for the application.
d) To provide reusable and extensible functionality in the application.

Answer: d) To provide reusable and extensible functionality in the application.

Q28. How are MVC modules organized in an ASP .Net MVC application?
a) MVC modules are organized based on the type of user interface components they contain.
b) MVC modules are organized into folders based on their functionality.
c) MVC modules are automatically generated based on the data and business logic of the application.
d) MVC modules are organized alphabetically in the "web.config" file.

Answer: b) MVC modules are organized into folders based on their functionality.

Q29. What is the purpose of the "Area" concept in ASP .Net MVC?
a) The "Area" concept is used to define the layout and design of the web pages.
b) The "Area" concept is used to handle user input and interact with the model and controller.
c) The "Area" concept is used to store and manage data for the application.
d) The "Area" concept is used to organize related functionality into separate sections of the application.

Answer: d) The "Area" concept is used to organize related functionality into separate sections of the application.

Q30. How do you create a new "Area" in an ASP .Net MVC application?
a) By using the "CreateArea" method of the Area class.
b) By using the "New Area" option in the "File" menu.
c) By using the "Add" option in the "Area" context menu.
d) By creating a new folder with the name of the area and adding the necessary files.

Answer: d) By creating a new folder with the name of the area and adding the necessary files.

10.7. Data Management with ADO.NET

Q31. What is ADO.NET in an ASP .Net MVC application?
a) ADO.NET is a data access technology used to manage data in the client's browser.
b) ADO.NET is a data access technology used to manage data on the server.
c) ADO.NET is a data access technology used to manage data in a centralized database.
d) ADO.NET is a data access technology used to manage data in the "web.config" file.

Answer: b) ADO.NET is a data access technology used to manage data on the server.

Q32. What are the primary components of ADO.NET?
a) Models, Views, and Controllers
b) DataTables, DataViews, and DataSets
c) Entity Framework, LINQ, and DbContext
d) SqlConnection, SqlCommand, and SqlDataReader

Answer: d) SqlConnection, SqlCommand, and SqlDataReader

Q33. What is the purpose of a SqlConnection object in ADO.NET?
a) The SqlConnection object is used to define the layout and design of the web pages.
b) The SqlConnection object is used to handle user input and interact with the model and controller.
c) The SqlConnection object is used to store and manage data for the application.
d) The SqlConnection object is used to establish a connection to a database.

Answer: d) The SqlConnection object is used to establish a connection to a database.

Q34. What is the purpose of a SqlCommand object in ADO.NET?
a) The SqlCommand object is used to define the layout and design of the web pages.
b) The SqlCommand object

is used to handle user input and interact with the model and controller.
c) The SqlCommand object is used to store and manage data for the application.
d) The SqlCommand object is used to execute SQL queries and commands on the database.

Answer: d) The SqlCommand object is used to execute SQL queries and commands on the database.

Q35. What is the purpose of a SqlDataReader object in ADO.NET?
a) The SqlDataReader object is used to define the layout and design of the web pages.
b) The SqlDataReader object is used to handle user input and interact with the model and controller.
c) The SqlDataReader object is used to store and manage data for the application.
d) The SqlDataReader object is used to retrieve data from the database as a read-only forward-only stream of records.

Answer: d) The SqlDataReader object is used to retrieve data from the database as a read-only forward-only stream of records.

11. Lecture: Advanced Asp.Net MVC Concepts

11.1. Understanding Routing & Request Life Cycle

Q1. What is routing in the context of ASP .Net MVC?
a) The process of defining the layout of a web page.
b) The process of handling incoming HTTP requests and mapping them to specific controller actions.
c) The process of compressing CSS and JavaScript files for improved performance.
d) The process of handling errors and exceptions in an MVC application.

Answer: b) The process of handling incoming HTTP requests and mapping them to specific controller actions.

Q2. In ASP .Net MVC, where is the routing configuration typically defined?
a) In the global.asax file.
b) In the web.config file.
c) In the appsettings.json file.
d) In the Startup.cs file.

Answer: a) In the global.asax file.

Q3. What is the purpose of the "RouteConfig.cs" file in an ASP .Net MVC application?
a) To configure routing settings for the application.
b) To define the layout and design of the web pages.
c) To handle errors and exceptions in the application.
d) To define filters and custom action filters.

Answer: a) To configure routing settings for the application.

Q4. In the ASP .Net MVC Request Life Cycle, what is the first step?
a) Authentication
b) Routing
c) Model Binding
d) Action Execution

Answer: b) Routing

Q5. What is the purpose of the Model Binding step in the ASP .Net MVC Request Life Cycle?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To extract data from the HTTP request and populate the parameters of the action method.
d) To handle errors and exceptions in the application.

Answer: c) To extract data from the HTTP request and populate the parameters of the action method.

11.2. Layouts, Bundle, Minification

Q6. What is a layout in the context of ASP .Net MVC?
a) A predefined design template for a web page.
b) A mechanism to handle errors and exceptions in an MVC application.
c) A method to compress CSS and JavaScript files for improved performance.
d) A mechanism to handle incoming HTTP requests and map them to specific controller actions.

Answer: a) A predefined design template for a web page.

Q7. How do you define a layout in an ASP .Net MVC application?
a) By using the "RenderSection" method in the view.
b) By using the "Layout" directive in the view.

c) By using the "UseLayout" method in the controller.
d) By using the "SetLayout" directive in the view.

Answer: b) By using the "Layout" directive in the view.

Q8. What is bundling in ASP .Net MVC?
a) The process of defining the layout and design of the web pages.
b) The process of handling incoming HTTP requests and mapping them to specific controller actions.
c) The process of compressing CSS and JavaScript files for improved performance.
d) The process of handling errors and exceptions in an MVC application.

Answer: c) The process of compressing CSS and JavaScript files for improved performance.

Q9. What is minification in the context of ASP .Net MVC?
a) The process of defining the layout and design of the web pages.
b) The process of handling incoming HTTP requests and mapping them to specific controller actions.
c) The process of compressing CSS and JavaScript files by removing unnecessary characters and whitespace.
d) The process of handling errors and exceptions in an MVC application.

Answer: c) The process of compressing CSS and JavaScript files by removing unnecessary characters and whitespace.

Q10. How do you enable bundling and minification in an ASP .Net MVC application?
a) By using the "UseBundles" method in the Startup.cs file.
b) By using the "EnableBundles" method in the global.asax file.
c) By using the "BundleConfig.cs" file to register bundles in the application.
d) By using the "UseMinification" method in the Startup.cs file.

Answer: c) By using the "BundleConfig.cs" file to register bundles in the application.

11.3. Asynchronous Actions

Q11. What is the purpose of using asynchronous actions in ASP .Net MVC?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To improve the performance of the application by executing long-running tasks asynchronously.
c) To compress CSS and JavaScript files for improved performance.
d) To handle errors and exceptions in an MVC application.

Answer: b) To improve the performance of the application by executing long-running tasks asynchronously.

Q12. How do you define an asynchronous action method in an ASP .Net MVC controller?
a) By using the "Async" keyword before the action method.
b) By using the "ActionAsync" suffix for the action method.
c) By using the "Async" suffix for the action method.
d) By using the "Asynchronous" keyword before the action method.

Answer: a) By using the "Async" keyword before the action method.

Q13. What is the return type of an asynchronous action method in ASP .Net MVC?
a) void
b) Task
c) ActionResult
d) Task<ActionResult>

Answer: d) Task<ActionResult>

Q14. What is the purpose of the "await" keyword in an asynchronous action method?
a) To handle incoming HTTP requests and map them to specific controller actions.

b) To compress CSS and JavaScript files for improved performance.
c) To indicate that the method is asynchronous and can be awaited.
d) To handle errors and exceptions in an MVC application.

Answer: c) To indicate that the method is asynchronous and can be awaited.

Q15. How do you call an asynchronous action method from a view in an ASP .Net MVC application?
a) By using the "Html.Action" method in the view.
b) By using the "Html.BeginForm" method in the view.
c) By using the "Html.RenderAction" method in the view.
d) By using the "Html.RenderPartial" method in the view.

Answer: c) By using the "Html.RenderAction" method in the view.

11.4. Error Handling in MVC with Log Entry

Q16. What is error handling in the context of ASP .Net MVC?
a) The process of handling incoming HTTP requests and mapping them to specific controller actions.
b) The process of defining the layout and design of the web pages.
c) The process of handling errors and exceptions that occur in an MVC application.
d) The process of compressing CSS and JavaScript files for improved performance.

Answer: c) The process of handling errors and exceptions that occur in an MVC application.

Q17. What is the purpose of logging errors in an ASP .Net MVC application?
a) To display detailed error messages to the users.
b) To compress CSS and JavaScript files for improved performance.
c) To record information about errors that occur in the application for debugging and analysis.
d) To handle incoming HTTP requests and map them to specific controller actions.

Answer: c) To record information about errors that occur in the application for debugging and analysis.

Q18. What is the recommended approach for error handling in ASP

.Net MVC?
a) Using try-catch blocks in every action method.
b) Using the "OnException" method in the base controller class.
c) Using the "HandleError" attribute at the controller level.
d) Using the "HandleError" attribute at the action method level.

Answer: c) Using the "HandleError" attribute at the controller level.

Q19. How do you log errors in an ASP .Net MVC application?
a) By using the "Log" method of the "System.Exception" class.
b) By using the "Logger" class from the "System.Diagnostics" namespace.
c) By using the "EventLog" class from the "System.Diagnostics" namespace.
d) By using the "Log" method of the "System.Web.Mvc" namespace.

Answer: b) By using the "Logger" class from the "System.Diagnostics" namespace.

Q20. What is the purpose of the "EventLog" class in an ASP .Net MVC application?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To display detailed error messages to the users.
d) To log errors and events that occur in the application.

Answer: d) To log errors and events that occur in the application.

11.5. Filters and Custom Action Filter

Q21. What are filters in the context of ASP .Net MVC?
a) A mechanism to handle errors and exceptions in an MVC application.
b) A method to compress CSS and JavaScript files for improved performance.
c) A mechanism to handle incoming HTTP requests and map them to specific controller actions.
d) A way to inject custom logic into the request processing pipeline.

Answer: d) A way to inject custom logic into the request processing pipeline.

Q22. How do you implement a filter in an ASP .Net MVC application?
a) By using the "FilterAttribute" class.
b) By using the "ActionFilter" class.
c) By using the "HandleErrorAttribute" class.
d) By using the "Filter" keyword in the action method.

Answer: b) By using the "ActionFilter" class.

Q23. What is the purpose of an action filter in ASP .Net MVC?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To add custom logic before or after the execution of an action method.
d) To display detailed error messages to the users.

Answer: c) To add custom logic before or after the execution of an action method.

Q24. How do you apply a filter to an action method in an ASP .Net MVC application?
a) By using the "Filter" keyword in the action method.
b) By using the "ApplyFilter" method in the action method.
c) By using the "Filter" attribute on the action method.
d) By using the "Filter" attribute on the controller class.

Answer: c) By using the "Filter" attribute on the action method.

11.6. MVC Security

Q25. What is security in the context of ASP .Net MVC?
a) The process of handling incoming HTTP requests and mapping them to specific controller actions.
b) The process of defining the layout and design of the web pages.
c) The process of protecting an MVC application from unauthorized access and attacks.
d) The process of compressing CSS and JavaScript files for improved performance.

Answer: c) The process of protecting an MVC application from unauthorized access and attacks.

Q26. What are some common security threats in an ASP .Net MVC application?
a) SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF).
b) Cookies, Sessions, and Application state.
c) TempData, ViewData, and Application state.
d) TempData, TempDataDictionary, and TempDataMessage.

Answer: a) SQL injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF).

Q27. What is the purpose of authentication in ASP .Net MVC?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To verify the identity of a user and grant access to protected resources.
d) To display detailed error messages to the users.

Answer: c) To verify the identity of a user and grant access to protected resources.

Q28. How do you enable authentication in an ASP .Net MVC application?
a) By using the "Authentication" method in the Startup.cs file.

b) By using the "UseAuthentication" method in the global.asax file.
c) By using the "Authorize" attribute on the action method or controller.
d) By using the "Authentication" attribute on the action method or controller.

Answer: c) By using the "Authorize" attribute on the action method or controller.

Q29. What is the purpose of the "AllowAnonymous" attribute in ASP .Net MVC?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To allow anonymous access to action methods that are otherwise protected by authentication.
c) To compress CSS and JavaScript files for improved performance.
d) To display detailed error messages to the users.

Answer: b) To allow anonymous access to action methods that are otherwise protected by authentication.

Q30. What is Cross-Site Scripting (XSS) in the context of ASP .Net MVC?
a) A security threat that allows attackers to inject malicious scripts into

web pages viewed by other users.
b) A mechanism to handle errors and exceptions in an MVC application.
c) A process of compressing CSS and JavaScript files for improved performance.
d) A process of handling incoming HTTP requests and mapping them to specific controller actions.

Answer: a) A security threat that allows attackers to inject malicious scripts into web pages viewed by other users.

12. Lecture: Entity Framework and ASP.Net MVC Core

12.1. Entity Framework Introduction

Q1. What is Entity Framework (EF)?
a) A front-end framework for building user interfaces in web applications.
b) A data access technology that allows developers to work with databases using .NET objects.
c) An open-source JavaScript library for building interactive web applications.
d) A cloud computing service provided by Microsoft.

Answer: b) A data access technology that allows developers to work with databases using .NET objects.

Q2. Which of the following is NOT a benefit of using Entity Framework?
a) Reduced development time by eliminating the need to write SQL queries manually.
b) Improved performance compared to ADO.NET.
c) Improved security by preventing SQL injection attacks.
d) Increased productivity due to automatic code generation.

Answer: b) Improved performance compared to ADO.NET.

Q3. Entity Framework supports which of the following database providers?
a) SQL Server only.
b) MySQL only.
c) SQLite only.
d) Multiple database providers, including SQL Server, MySQL, SQLite, and more.

Answer: d) Multiple database providers, including SQL Server, MySQL, SQLite, and more.

Q4. In Entity Framework, what is the role of a DbContext?
a) It is responsible for creating the database schema.
b) It is responsible for defining the business logic of the application.
c) It is responsible for representing a session with the database and can be used to query and save data.
d) It is responsible for defining the model classes that represent database tables.

Answer: c) It is responsible for representing a session with the database and can be used to query and save data.

Q5. Which of the following approaches is used by Entity Framework to represent data models?
a) Code First
b) Model First
c) Database First
d) All of the above

Answer: d) All of the above

12.2. Different Approaches

Q6. What is the Code First approach in Entity Framework?
a) It is an approach where the database schema is created first, and then the model classes are generated from it.
b) It is an approach where the model classes are defined first, and then the database schema is generated from them.
c) It is an approach where the database schema and model classes are defined separately and then mapped together.
d) It is an approach where the database schema and model classes are generated automatically by Entity Framework.

Answer: b) It is an approach where the model classes are defined first, and then the database schema is generated from them.

Q7. In the Model First approach, how is the database schema created?
a) The database schema is created using Entity Framework's migration feature.
b) The database schema is created by writing SQL queries manually.
c) The database schema is created by defining the model classes and then generating the schema from them.
d) The database schema is created automatically by Entity Framework.

Answer: c) The database schema is created by defining the model classes and then generating the schema from them.

Q8. Which approach is best suited for greenfield projects where the database does not exist?
a) Code First
b) Model First
c) Database First
d) All approaches are equally suitable for greenfield projects.

Answer: a) Code First

Q9. In the Database First approach, how are model classes created?
a) Model classes are created by writing SQL queries manually.
b) Model classes are created automatically by Entity Framework based on the existing database schema.
c) Model classes are created by defining the model structure and then generating the database schema from them.
d) Model classes are created by using a third-party library.

Answer: b) Model classes are created automatically by Entity Framework based on the existing database schema.

Q10. Which approach is best suited when working with legacy databases or databases created using other tools?
a) Code First
b) Model First
c) Database First
d) All approaches are equally suited for working with legacy databases.

Answer: c) Database First

12.3. Using Code First Approach

Q11. How are model classes defined in the Code First approach?
a) Model classes are defined using XML files.
b) Model classes are defined using JSON files.
c) Model classes are defined using C# or VB.NET classes with properties representing database tables.
d) Model classes are defined using SQL queries.

Answer: c) Model classes are defined using C# or VB.NET classes with properties representing database tables.

Q12. In the Code First approach, how can you specify the primary key of a model class?
a) By using the [Key] attribute on the property representing the primary key.
b) By using the [Primary] attribute on the property representing the primary key.
c) By using the [PrimaryKey] attribute on the property representing the primary key.
d) By using the [Id] attribute on the property representing the primary key.

Answer: a) By using the [Key] attribute on the property representing the primary key.

Q13. In the Code First approach, how can you specify a property as a foreign key?

a) By using the [Foreign] attribute on the property.
b) By using the [ForeignKey] attribute on the property.
c) By using the [Relation] attribute on the property.
d) By using

the [Association] attribute on the property.

Answer: b) By using the [ForeignKey] attribute on the property.

Q14. How can you configure the relationship between two model classes in the Code First approach?
a) By using the [Association] attribute on both model classes.
b) By using the [Relation] attribute on both model classes.
c) By using the [HasOne] and [HasMany] attributes on the appropriate properties.
d) By writing SQL queries manually to define the relationship.

Answer: c) By using the [HasOne] and [HasMany] attributes on the appropriate properties.

Q15. What is Entity Framework Migrations in the Code First approach?
a) It is a feature that allows you to update the database schema automatically when the model classes change.
b) It is a feature that allows you to generate model classes from an existing database schema.
c) It is a feature that allows you to create a new database from the model classes.
d) It is a feature that allows you to create a model class from an existing database table.

Answer: a) It is a feature that allows you to update the database schema automatically when the model classes change.

12.4. Database Migrations

Q16. What are database migrations in Entity Framework?
a) Database migrations are a way to migrate the database to a different server.
b) Database migrations are a way to create a new database from scratch.
c) Database migrations are a way to update the database schema to match changes in the model classes.
d) Database migrations are a way to delete and recreate the database.

Answer: c) Database migrations are a way to update the database schema to match changes in the model classes.

Q17. How do you enable database migrations in an Entity Framework Code First project?
a) By installing the "EntityFramework.Migrations" NuGet package.
b) By running a command in the Package Manager Console: "Enable-Migrations".
c) By adding a new migration file to the project manually.
d) By configuring the "MigrationEnabled" property in the "web.config" file.

Answer: b) By running a command in the Package Manager Console: "Enable-Migrations".

Q18. What does the "Add-Migration" command do in Entity Framework Code First?
a) It creates a new migration file with the changes made to the model classes.
b) It applies the pending migrations to the database.
c) It deletes the last migration applied to the database.
d) It checks for any changes in the model classes that have not been added as migrations.

Answer: a) It creates a new migration file with the changes made to the model classes.

Q19. How do you apply the pending migrations to the database in Entity Framework Code First?
a) By running a command in the Package Manager Console: "Apply-Migration".
b) By running a command in the Package Manager Console: "Update-Database".
c) By running a command in the Package Manager Console: "ApplyPendingMigrations".
d) By running a command in the Package Manager Console: "Execute-Migrations".

Answer: b) By running a command in the Package Manager Console: "Update-Database".

Q20. What is the purpose of the "Down" method in a migration file?
a) The "Down" method is used to apply the migration to the database.
b) The "Down" method is used to revert the changes made by the migration from the database.
c) The "Down" method is used to delete the migration file.
d) The "Down" method is used to check if the migration can be applied.

Answer: b) The "Down" method is used to revert the changes made by the migration from the database.

12.5. CRUD operation using EF

Q21. What is CRUD in the context of data management?
a) CRUD stands for "Create, Retrieve, Update, Delete," which are the four basic operations performed on data.
b) CRUD stands for "Centralized, Read, Update, Delete," which are the four main data access patterns.
c) CRUD stands for "Control, Retrieve, Update, Delete," which are the four steps in a data transaction.
d) CRUD stands for "Create, Retrieve, Upload, Download," which are the four data synchronization tasks.

Answer: a) CRUD stands for "Create, Retrieve, Update, Delete," which are the four basic operations performed on data.

Q22. How do you perform the "Create" operation using Entity Framework?
a) By calling the "SaveChanges" method on the DbContext after adding a new entity to the DbSet.
b) By calling the "Create" method on the DbContext and passing the new entity as a parameter.
c) By calling the "Insert" method on the DbContext and passing the new entity as a parameter.
d) By calling the "Create" method on the DbSet and passing the new entity as a parameter.

Answer: a) By calling the "SaveChanges" method on the DbContext after adding a new entity to the DbSet.

Q23. How do you perform the "Retrieve" operation using Entity Framework?
a) By calling the "Get" method on the DbContext and passing the primary key as a parameter.
b) By calling the "Find" method on the DbSet and passing the primary key as a parameter.
c) By calling the "Query" method on the DbContext and passing the primary key as a parameter.
d) By calling the "Retrieve" method on the DbSet and passing the primary key as a parameter.

Answer: b) By calling the "Find" method on the DbSet and passing the primary key as a parameter.

Q24. How do you perform the "Update" operation using Entity Framework?
a) By calling the "Update" method on the DbContext and passing the updated entity as a parameter.
b) By calling the "Update" method on the DbSet and passing the updated entity as a parameter.
c) By calling the "SaveChanges" method on the DbContext after making changes to the entity.
d) By calling the "SaveChanges" method on the DbSet after making changes to the entity.

Answer: c) By calling the "SaveChanges" method on the DbContext after making changes to the entity.

Q25. How do you perform the "Delete" operation using Entity Framework?
a) By calling the "Delete" method on the DbContext and passing the entity to be deleted as a parameter.
b) By calling the "Remove" method on the DbSet and passing the entity to be deleted as a parameter.
c) By calling the "Delete" method on the DbSet and passing the entity to be deleted as a parameter.
d) By calling the "Remove" method on the DbContext and passing the entity to be deleted as a parameter.

Answer: b) By calling the "Remove" method on the DbSet and passing the entity to be deleted as a parameter.

12.6. Understanding ASP.Net MVC Core

Q26. What is ASP.Net MVC Core?
a) ASP.Net MVC Core is a cross-platform, open-source framework for building modern web applications using .NET Core.
b) ASP.Net MVC Core is a JavaScript library for building user interfaces in web applications.
c) ASP.Net MVC Core is a database access technology that allows developers to work with databases using .NET objects.
d) ASP.Net MVC Core is a cloud computing service provided by Microsoft.

Answer: a) ASP.Net MVC Core is a cross-platform, open-source framework for building modern web applications using .NET Core.

Q27. What are

the main components of ASP.Net MVC Core?
a) Models, Views, Controllers, and Data Access Layer.
b) Models, Views, Controllers, and Routing.
c) Models, Views, Controllers, and Entity Framework.
d) Models, Views, Controllers, and Web API.

Answer: b) Models, Views, Controllers, and Routing.

Q28. In ASP.Net MVC Core, how does routing work?
a) Routing is configured in the "web.config" file.
b) Routing is configured in the "RouteConfig.cs" file.
c) Routing is configured in the "Startup.cs" file.
d) Routing is configured in the "App_Start" folder.

Answer: c) Routing is configured in the "Startup.cs" file.

Q29. In ASP.Net MVC Core, how are requests processed?
a) Requests are processed by the Model.
b) Requests are processed by the View.
c) Requests are processed by the Controller.
d) Requests are processed by the Data Access Layer.

Answer: c) Requests are processed by the Controller.

Q30. What is the purpose of a Model in ASP.Net MVC Core?
a) The Model is responsible for defining the layout and design of the web pages.
b) The Model is responsible for representing the data and the business logic of the application.
c) The Model is responsible for handling user input and interacting with the database.
d) The Model is responsible for handling HTTP requests and generating HTTP responses.

Answer: b) The Model is responsible for representing the data and the business logic of the application.

12.7. Creating a simple MVC Core Application

Q31. How do you create a new ASP.Net MVC Core application in Visual Studio?
a) By selecting "File" > "New Project" > "ASP.Net Core Web Application" and choosing "MVC" as the project template.
b) By selecting "File" > "New Project" > "ASP.Net Web Application" and choosing "MVC" as the project template.
c) By selecting "File" > "New Project" > "Console Application" and choosing "MVC" as the project template.
d) By selecting "File" > "New Project" > "Class Library" and choosing "MVC" as the project template.

Answer: a) By selecting "File" > "New Project" > "ASP.Net Core Web Application" and choosing "MVC" as the project template.

Q32. In an ASP.Net MVC Core application, where are the views stored?

a) Views are stored in the "Views" folder.
b) Views are stored in the "Models" folder.
c) Views are stored in the "Controllers" folder.
d) Views are stored in the "App_Start" folder.

Answer: a) Views are stored in the "Views" folder.

Q33. How are views rendered in ASP.Net MVC Core?
a) Views are rendered using Razor syntax.
b) Views are rendered using JavaScript.
c) Views are rendered using XML.
d) Views are rendered using CSS.

Answer: a) Views are rendered using Razor syntax.

Q34. What is the purpose of a Controller in ASP.Net MVC Core?
a) The Controller is responsible for defining the layout and design of the web pages.
b) The Controller is responsible for handling user input and interacting with the database.
c) The Controller is responsible for representing the data and the business logic of the application.
d) The Controller is responsible for handling HTTP requests and generating HTTP responses.

Answer: d) The Controller is responsible for handling HTTP requests and generating HTTP responses.

Q35. In an ASP.Net MVC Core application, how are models passed to views?
a) Models are passed to views using the ViewBag object.
b) Models are passed to views using the ViewData dictionary.
c) Models are passed to views using the Model property of the View.
d) Models are passed to views using the Model property of the Controller.

Answer: c) Models are passed to views using the Model property of the View.

13. Lecture: Localization and Deployment in MVC

13.1. Localization in MVC (Demo Only)

Q1. What is localization in the context of ASP .Net MVC?
a) The process of deploying an MVC application to a web server.
b) The process of handling incoming HTTP requests and mapping them to specific controller actions.
c) The process of adapting an application to different cultures and languages.
d) The process of compressing CSS and JavaScript files for improved performance.

Answer: c) The process of adapting an application to different cultures and languages.

Q2. What is the purpose of localization in an MVC application?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To improve the performance of the application by compressing files.
c) To make the application accessible to users from different regions and language backgrounds.
d) To display detailed error messages to the users.

Answer: c) To make the application accessible to users from different regions and language backgrounds.

Q3. How do you enable localization in an ASP .Net MVC application?
a) By using the "Localize" attribute in the view.
b) By using the "Localize" method in the controller.
c) By using the "UseLocalization" method in the global.asax file.
d) By using the "Localization" middleware in the Startup.cs file.

Answer: d) By using the "Localization" middleware in the Startup.cs file.

Q4. What is the purpose of resource files in ASP .Net MVC localization?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To store localized content such as strings, labels, and messages.
d) To display detailed error messages to the users.

Answer: c) To store localized content such as strings, labels, and messages.

Q5. How do you access localized resources in a view in an ASP .Net MVC application?
a) By using the "Localize" method in the view.
b) By using the "ViewData" dictionary in the view.
c) By using the "Localize" attribute on the view model.
d) By using the "ResourceManager" class to retrieve resources.

Answer: d) By using the "ResourceManager" class to retrieve resources.

Q6. In the context of ASP .Net MVC, what does the term "culture" refer to?
a) The process of handling incoming HTTP requests and mapping them to specific controller actions.
b) The specific region and language settings of a user.
c) The process of compressing CSS and JavaScript files for improved performance.
d) The process of deploying an MVC application to a web server.

Answer: b) The specific region and language settings of a user.

Q7. What is the purpose of the "CultureInfo" class in ASP .Net MVC localization?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To provide information about the user's specific culture and language settings.
d) To display detailed error messages to the users.

Answer: c) To provide information about the user's specific culture and language settings.

Q8. How do you set the culture and language in an ASP .Net MVC application?
a) By using the "SetCulture" method in the Startup.cs file.
b) By using the "CultureInfo" class in the global.asax file.
c) By using the "CultureInfo" class in the controller.
d) By using the "SetCulture" method in the view.

Answer: b) By using the "CultureInfo" class in the global.asax file.

Q9. What is the purpose of the "Thread.CurrentThread.CurrentCulture" property in an ASP .Net MVC application?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To provide information about the user's specific culture and language settings.
d) To display detailed error messages to the users.

Answer: c) To provide information about the user's specific culture and language settings.

Q10. How do you use localization in an ASP .Net MVC application for different languages?
a) By creating separate views and controllers for each language.
b) By using the "Localize" attribute on the action methods.
c) By using resource files for each language and accessing them based on the user's culture.
d) By using the "Localization" middleware in the Startup.cs file.

Answer: c) By using resource files for each language and accessing them based on the user's culture.

13.2. Deploying ASP .NET MVC Application (Demo only)

Q11. What is deployment in the context of ASP .Net MVC?
a) The process of handling incoming HTTP requests and mapping them to specific controller actions.
b) The process of adapting an application to different cultures and languages.
c) The process of compressing CSS and JavaScript files for improved performance.
d) The process of deploying an MVC application to a web server for public use.

Answer: d) The process of deploying an MVC application to a web server for public use.

Q12. What are the steps involved in deploying an ASP .Net MVC application?
a) Compression, localization, and authorization.
b) Routing, bundling, and minification.
c) Configuration, compilation, and execution.
d) Build, publish, and deploy.

Answer: d) Build, publish, and deploy.

Q13. How do you build an ASP .Net MVC application for deployment?
a) By using the "Build" option in the Visual Studio IDE.
b) By using the "Deploy" option in the Visual Studio IDE.
c) By using the "Run" option in the Visual Studio IDE.
d) By using the "Build Solution" option in the "Build" menu.

Answer: d) By using the "Build Solution" option in the "Build" menu.

Q14. What is the purpose of publishing an ASP .Net MVC application?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To prepare the application for deployment by creating a deployment package.
d) To display detailed error messages to the users.

Answer: c) To prepare the application for deployment by creating a deployment package.

Q15. How do you publish an ASP .Net MVC application?
a) By using the "Publish" option in the Visual Studio IDE.
b) By using the "Build" option in the Visual Studio IDE.
c) By using the "Deploy" option in the Visual Studio IDE.
d) By using the "Run" option in the Visual Studio IDE.

Answer: a) By using the "Publish" option in the Visual Studio IDE.

Q16. What is the output of publishing an ASP .Net MVC application?
a) A ZIP file containing all the source code and assets.
b) A compiled and optimized version of the application ready for deployment.
c) A report of errors and exceptions that occurred during the publishing process.
d) A summary of the application's performance metrics.

Answer: b) A compiled and optimized version of the application ready for deployment.

Q17. What is the purpose of the "Web.config" file in an ASP .Net MVC application during deployment?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To store

configuration settings for the application, including connection strings and app settings.
c) To compress CSS and JavaScript files for improved performance.
d) To display detailed error messages to the users.

Answer: b) To store configuration settings for the application, including connection strings and app settings.

Q18. What is the typical location for deploying an ASP .Net MVC application?
a) On a local development machine.
b) On a shared network drive.
c) On a production web server accessible over the internet.
d) On a virtual machine in the cloud.

Answer: c) On a production web server accessible over the internet.

Q19. How do you deploy an ASP .Net MVC application to a production web server?
a) By copying the project files directly to the server.
b) By using the "Deploy" option in the Visual Studio IDE.
c) By using the "Publish" option in the Visual Studio IDE.
d) By creating a deployment package and using a deployment tool or script.

Answer: d) By creating a deployment package and using a deployment tool or script.

Q20. What are some common deployment platforms for ASP .Net MVC applications?
a) IIS, Apache, and Nginx.
b) Visual Studio, Eclipse, and IntelliJ IDEA.
c) Windows, Linux, and macOS.
d) Azure, AWS, and Google Cloud Platform.

Answer: a) IIS, Apache, and Nginx.

14. Lecture: Windows Communication Foundation (WCF)

14.1. WCF Overview

Q1. What does WCF stand for in .NET?
a) Windows Communication Foundation
b) Windows Control Foundation
c) Windows Component Framework
d) Windows Core Fundamentals

Answer: a) Windows Communication Foundation

Q2. Which of the following is NOT a key feature of WCF?
a) Interoperability
b) Security
c) Simplicity
d) Performance

Answer: c) Simplicity

Q3. What is the primary purpose of WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To provide a unified programming model for building service-oriented applications.
c) To compress CSS and JavaScript files for improved performance.
d) To display detailed error messages to the users.

Answer: b) To provide a unified programming model for building service-oriented applications.

Q4. Which of the following is true about WCF communication?
a) WCF supports only HTTP communication.
b) WCF supports communication over multiple protocols, such as HTTP, TCP, and MSMQ.
c) WCF supports only UDP communication.
d) WCF supports communication over TCP and UDP only.

Answer: b) WCF supports communication over multiple protocols, such as HTTP, TCP, and MSMQ.

14.2. Service Contracts

Q5. What is a service contract in WCF?
a) A contract between the client and the server that defines the operations and data types used in the service.
b) A contract that specifies the security settings for the service.
c) A contract that defines the layout and design of the service's user interface.
d) A contract that specifies the data contracts used in the service.

Answer: a) A contract between the client and the server that defines the operations and data types used in the service.

Q6. Which attribute is used to define a service contract in WCF?
a) [ServiceBehavior]
b) [DataContract]
c) [OperationContract]
d) [ServiceContract]

Answer: d) [ServiceContract]

Q7. What is the purpose of the [OperationContract] attribute in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To define the layout and design of the service's user interface.

c) To specify the security settings for the service.
d) To define an operation that can be called by clients of the service.

Answer: d) To define an operation that can be called by clients of the service.

Q8. Which of the following is true about service contracts in WCF?
a) A service can have multiple service contracts.
b) A service can have only one service contract.
c) Service contracts are not necessary in WCF.
d) Service contracts can only be defined in the client application.

Answer: a) A service can have multiple service contracts.

14.3. Data Contracts

Q9. What is a data contract in WCF?
a) A contract between the client and the server that defines the operations and data types used in the service.
b) A contract that specifies the data contracts used in the service.
c) A contract that defines the layout and design of the service's user interface.
d) A contract that specifies the security settings for the service.

Answer: b) A contract that specifies the data contracts used in the service.

Q10. Which attribute is used to define a data contract in WCF?
a) [ServiceBehavior]
b) [DataContract]
c) [OperationContract]
d) [ServiceContract]

Answer: b) [DataContract]

Q11. What is the purpose of the [DataMember] attribute in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To define the layout and design of the service's user interface.
c) To specify the operations used in the service.
d) To define a member of a data contract.

Answer: d) To define a member of a data contract.

Q12. Which of the following is true about data contracts in WCF?
a) A data contract can only contain primitive data types.
b) A data contract can only contain complex data types.
c) A data contract can contain a mixture of primitive and complex data types.
d) Data contracts are not necessary in WCF.

Answer: c) A data contract can contain a mixture of primitive and complex data types.

14.4. Message Contracts

Q13. What is a message contract in WCF?
a) A contract between the client and the server that defines the operations and data types used in the service.
b) A contract that specifies the security settings for the service.
c) A contract that defines the layout and design of the service's user interface.
d) A contract that allows precise control over the SOAP message format.

Answer: d) A contract that allows precise control over the SOAP message format.

Q14. Which attribute is used to define a message contract in WCF?

a) [ServiceBehavior]
b) [DataContract]
c) [MessageContract]
d) [ServiceContract]

Answer: c) [MessageContract]

Q15. What is the purpose of the [MessageBodyMember] attribute in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To define the layout and design of the service's user interface.
c) To specify the operations used in the service.
d) To define a member of a message contract.

Answer: d) To define a member of a message contract.

Q16. Which of the following is true about message contracts in WCF?
a) Message contracts are used to define service operations.
b) Message contracts are used to define service data contracts.
c) Message contracts are used to define the layout of the service's user interface.
d) Message contracts allow precise control over the SOAP message format.

Answer: d) Message contracts allow precise control over the SOAP message format.

14.5. Operation Contract and Channel Shapes

Q17. What is an operation contract in WCF?
a) A contract between the client and the server that defines the operations and data types used in the service.
b) A contract that specifies the operations used in the service.
c) A contract that defines the layout and design of the service's user interface.
d) A contract that allows precise control over the SOAP message format.

Answer: b) A contract that specifies the operations used in the service.

Q18. Which attribute is used to define an operation contract in WCF?
a) [ServiceBehavior]
b) [DataContract]
c) [OperationContract]
d) [ServiceContract]

Answer: c) [OperationContract]

Q19. What is the purpose of the [OperationContract] attribute in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To define the layout and design of the service's user interface.
c) To specify the operations used in the service.
d) To define a member of a message contract.

Answer: c) To specify the operations used in the service.

Q20. Which of the following is true about operation contracts in WCF?
a) An operation contract can only be applied to a service contract.
b) An operation contract can only be applied to a data contract.
c) An operation contract can be applied to both service and data contracts.
d) Operation contracts are not necessary in WCF.

Answer: c) An operation contract can be applied to both service and data contracts.

14.6. Channel Listeners

Q21. What is a channel listener in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A component that listens for incoming messages on a specific communication channel.
d) A component that handles exceptions and errors in the service.

Answer: c) A component that listens for incoming messages on a specific communication channel.

Q22. Which of the following is NOT a type of channel listener in WCF?
a) HTTP channel listener
b) TCP channel listener
c) MSMQ channel listener
d) SOAP channel listener

Answer: d) SOAP channel listener

Q23. What is the purpose of a channel listener in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To listen for incoming messages and process them.
d) To handle exceptions and errors in the service.

Answer: c) To listen for incoming messages and process them.

Q24. How do you configure a channel listener in WCF?
a) By using the [ChannelListener] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of channel listener in the configuration file.
d) By using the [ChannelFactory] class in the client application.

Answer: c) By specifying the type of channel listener in the configuration file.

14.7. Channel Factories

Q25. What is a channel factory in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A component that creates channels for communication with a service.
d) A component that handles exceptions and errors in the service.

Answer: c) A component that creates channels for communication with a service.

Q26. Which of the following is NOT a type of channel factory in WCF?
a) HTTP channel factory
b) TCP channel factory
c) MSMQ channel factory
d) SOAP channel factory

Answer: d) SOAP channel factory

Q27. What is the purpose of a channel factory in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To create channels for communication with a service.
d) To handle exceptions and errors in the service.

Answer: c) To create channels for communication with a service.

Q28. How do you configure a channel factory in WCF?
a) By using the [ChannelFactory] attribute in the service contract.

b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of channel factory in the configuration file.
d) By using the [ChannelListener] class in the client application.

Answer: c) By specifying the type of channel factory in the configuration file.

14.8. ICommunicationObject

Q29. What is ICommunicationObject in WCF?
a) An interface that handles incoming HTTP requests and maps them to specific controller actions.
b) An interface that compresses CSS and JavaScript files for improved performance.
c) An interface that represents the core communication functionality of a WCF object.
d) An interface that handles exceptions and errors in the service.

Answer: c) An interface that represents the core communication functionality of a WCF object.

Q30. Which of the following is NOT a state of ICommunicationObject in WCF?
a) Created
b) Opened
c) Closed
d) Configured

Answer: d) Configured

Q31. What is the purpose of ICommunicationObject in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To represent the core communication functionality of a WCF object.
d) To handle exceptions and errors in the service.

Answer: c) To represent the core communication functionality of a WCF object.

Q32. Which methods are present in ICommunicationObject that allow you to transition between different states?
a) Open() and Close()
b) Start() and Stop()
c) Run() and Exit()
d) Connect() and Disconnect()

Answer: a) Open() and Close()

14.9. Binding

Q33. What is a binding in WCF?
a) A configuration setting that handles incoming HTTP requests and maps them to specific controller actions.
b) A configuration setting that compresses CSS and JavaScript files for improved performance.
c) A configuration setting that specifies how the client and service communicate with each other.
d) A configuration setting that handles exceptions and errors in the service.

Answer: c) A configuration setting that specifies how the client and service communicate with each other.

Q34. Which of the following is NOT a type of binding in WCF?
a) BasicHttpBinding
b) WSHttpBinding
c) NetMsmqBinding
d) SoapBinding

Answer: d) SoapBinding

Q35. What is the purpose of a binding in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To specify how the client and service communicate with each other.
d) To handle exceptions and errors in the service.

Answer: c) To specify how the client and service communicate with each other.

Q36. How do you configure a binding in WCF?
a) By using the [Binding] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of binding in the configuration file.
d) By using the [ChannelFactory] class in the client application.

Answer: c) By specifying the type of binding in the configuration file.

14.10. Exposing a Service Contract over Multiple Endpoints

Q37. What is an endpoint in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A component that exposes a service contract to clients.
d) A component that handles exceptions and errors in the service.

Answer: c) A component that exposes a service contract to clients.

Q38. Which of the following is NOT a type of endpoint in WCF?
a) HTTP endpoint
b) TCP endpoint
c) MSMQ endpoint
d) SOAP endpoint

Answer: d) SOAP endpoint

Q39. What is the purpose of an endpoint in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To expose a service contract to clients.
d) To handle exceptions and errors in the service.

Answer: c) To expose a service contract to clients.

Q40. How do you configure an endpoint in WCF?
a) By using the [Endpoint] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of endpoint in the configuration file.
d) By using the [ChannelFactory] class in the client application.

Answer: c) By specifying the type of endpoint in the configuration file.

14.11. Implementing Transactions (Operation Behavior)

Q41. What are transactions in WCF?
a) A way to handle incoming HTTP requests and map them to specific controller actions.
b) A way to compress CSS and JavaScript files for improved performance.
c) A way to manage and control the ACID properties of a set of related operations.
d) A way to handle exceptions and errors in the service.

Answer: c) A way to manage and control the ACID properties of a set of related operations.

Q42. Which attribute is used to define an operation behavior in WCF?
a) [ServiceBehavior]
b) [OperationContract]
c) [TransactionFlow]
d) [Transaction]

Answer: d) [Transaction]

Q43. What is the purpose of the [TransactionFlow] attribute in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To specify the layout and design of the service's user interface.
c) To define the operations used in the service.
d) To specify whether transactions are supported for a particular operation.

Answer: d) To specify whether transactions are supported for a particular operation.

Q44. Which of the following is true about implementing transactions in WCF?
a) All operations in a service must support transactions.
b) Only one transaction can be active at a time for a service.
c) Each operation in a service can have its own transaction settings.
d) Transaction support is not available in WCF.

Answer: c) Each operation in a service can have its own transaction settings.

14.12. Hosting a Service in Windows Process Activation Services

Q45. What is Windows Process Activation Services (WAS) in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A feature in Windows Server that provides process activation and management for WCF services.
d) A component that handles exceptions and errors in the service.

Answer: c) A feature in Windows Server that provides process activation and management for WCF services.

Q46. Which of the following is NOT a type of hosting in WCF?
a) Self-hosting
b) Windows Service hosting
c) IIS hosting
d) Linux hosting

Answer: d) Linux hosting

Q47. What is the purpose of hosting a service in Windows Process Activation Services (WAS)?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To provide process activation and management for WCF services.
d) To handle exceptions and errors in the service.

Answer: c) To provide process activation and management for WCF services.

Q48. How do you host a service in Windows Process Activation Services (WAS)?
a) By using the [Host] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of hosting in the configuration file.
d) By using the [ServiceHost] class in the service application.

Answer: d) By using the [ServiceHost] class in the service application.

14.13. Hosting a Service in IIS 7

Q49. What is IIS 7 in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A web server application in Windows that can host WCF services.
d) A component that handles exceptions and errors in the service.

Answer: c) A web server application in Windows that can host WCF services.

Q50. Which of the following is true about hosting a service in IIS 7?
a) WCF services cannot be hosted in IIS 7.
b) IIS 7 can host WCF services only on Windows Server operating systems.
c) IIS 7 can host WCF services on both Windows Server and client operating systems.
d) IIS 7 can host WCF services only on Linux operating systems.

Answer: c) IIS 7 can host WCF services on both Windows Server and client operating systems.

Q51. What is the purpose of hosting a service in IIS 7?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To host WCF services in a web server application on Windows.
d) To handle exceptions and errors in the service.

Answer: c) To host WCF services in a web server application on Windows.

Q52. How do you host a service in IIS 7?
a) By using the [Host] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of hosting in the configuration file.
d) By deploying the service to the IIS web server.

Answer: d) By deploying the service to the IIS web server.

14.14. Self-Hosting in a Managed Windows Service

Q53. What is self-hosting in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) Hosting a WCF service within the same process as the client application.
d) A component that handles exceptions and errors in the service.

Answer: c) Hosting a WCF service within the same process as the client application.

Q54. Which of the following is NOT a type of self-hosting in WCF?
a) Windows Service self-hosting
b) Console Application self-hosting
c) Desktop Application self-hosting
d) Web Application self-hosting

Answer: d) Web Application self-hosting

Q55. What is the purpose of self-hosting in a managed Windows service?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To host a WCF service within a managed Windows service application.
d) To handle exceptions and errors in the service.

Answer: c) To host a WCF service within a managed Windows service application.

Q56. How do you implement self-hosting in a managed Windows service?
a) By using the [Host] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of hosting in the configuration file.
d) By using the [ServiceHost] class in the service application.

Answer: d) By using the [ServiceHost] class in the service application.

14.15. Creating REST Service with Get & Post

Q57. What is a REST service in WCF?
a) A component that handles incoming HTTP requests and maps them to specific controller actions.
b) A component that compresses CSS and JavaScript files for improved performance.
c) A service that follows the principles of Representational State Transfer (REST) for communication.
d) A component that handles exceptions and errors in the service.

Answer: c) A service that follows the principles of Representational State Transfer (REST) for communication.

Q58. Which of the following HTTP methods are commonly used in REST services?
a) GET and POST
b) PUT and DELETE
c) PATCH and HEAD
d) OPTIONS and TRACE

Answer: a) GET and POST

Q59. What is the purpose of a REST service in WCF?
a) To handle incoming HTTP requests and map them to specific controller actions.
b) To compress CSS and JavaScript files for improved performance.
c) To follow the principles of REST for communication.
d) To handle exceptions and errors in the service.

Answer: c) To follow the principles of REST for communication.

Q60. How do you create a REST service with GET and POST operations in WCF?
a) By using the [RestService] attribute in the service contract.
b) By using the [ServiceBehavior] attribute in the service implementation.
c) By specifying the type of binding and endpoint in the configuration file.
d) By using the [WebGet] and [WebInvoke] attributes on the service operations.

Answer: d) By using the [WebGet] and [WebInvoke] attributes on the service operations.

15. Lecture: Web APIs

15.1. Creating ASP.NET MVC Web API

Q1. What is ASP.NET MVC Web API?
a) It is a front-end framework for building user interfaces in web applications.
b) It is a data access technology that allows developers to work with databases using .NET objects.
c) It is a framework for building HTTP services that can be consumed by various clients, such as web browsers and mobile devices.
d) It is a cloud computing service provided by Microsoft.

Answer: c) It is a framework for building HTTP services that can be consumed by various clients, such as web browsers and mobile devices.

Q2. How do you create a new ASP.NET MVC Web API project in Visual Studio?
a) By selecting "File" > "New Project" > "ASP.NET Core Web Application" and choosing "API" as the project template.
b) By selecting "File" > "New Project" > "ASP.NET Web Application" and choosing "API" as the project template.
c) By selecting "File" > "New Project" > "Console Application" and choosing "API" as the project template.
d) By selecting "File" > "New Project" > "Class Library" and choosing "API" as the project template.

Answer: a) By selecting "File" > "New Project" > "ASP.NET Core Web Application" and choosing "API" as the project template.

Q3. What is the default HTTP method used by ASP.NET MVC Web API for a method that starts with "Get"?
a) GET
b) POST
c) PUT
d) DELETE

Answer: a) GET

Q4. How do you define a Web API method that accepts parameters in the request body?
a) By using the [FromBody] attribute on the parameter.
b) By using the [Route] attribute on the parameter.
c) By using the [HttpGet] attribute on the parameter.
d) By using the [HttpPost] attribute on the parameter.

Answer: a) By using the [FromBody] attribute on the parameter.

Q5. How do you return data from a Web API method?
a) By using the return keyword.
b) By using the ViewResult object.
c) By using the ContentResult object.
d) By using the HttpResponseMessage object.

Answer: d) By using the HttpResponseMessage object.

15.2. Different Verbs

Q6. Which HTTP verb is used to create a new resource in a Web API?
a) GET
b) POST
c) PUT
d) DELETE

Answer: b) POST

Q7. Which HTTP verb is used to update an existing resource in a Web API?
a) GET
b) POST
c) PUT
d) DELETE

Answer: c) PUT

Q8. Which HTTP verb is used to retrieve data from a Web API?
a) GET
b) POST
c) PUT
d) DELETE

Answer: a) GET

Q9. Which HTTP verb is used to delete a resource from a Web API?
a) GET
b) POST
c) PUT
d) DELETE

Answer: d) DELETE

Q10. How do you specify the HTTP verb for a Web API method?
a) By using the [HttpGet] attribute.
b) By using the [HttpPost] attribute.
c) By using the [HttpPut] attribute.
d) By using the [HttpDelete] attribute.

Answer: You can use any of the following attributes based on the HTTP verb: [HttpGet], [HttpPost], [HttpPut], [HttpDelete].

15.3. Consuming using a client

Q11. What is a Web API client?
a) It is a front-end framework for building user interfaces in web applications.
b) It is a data access technology that allows developers to work with databases using .NET objects.
c) It is a component that consumes the Web API and interacts with the data provided by the API.
d) It is a cloud computing service provided by Microsoft.

Answer: c) It is a component that consumes the Web API and interacts with the data provided by the API.

Q12. How can you consume a Web API in a .NET application?
a) By using the HttpClient class from the System.Net.Http namespace.
b) By using the SqlClient class from the System.Data.SqlClient namespace.
c) By using the WebClient class from the System.Net namespace.
d) By using the WebRequest class from the System.Net namespace.

Answer: a) By using the HttpClient class from the System.Net.Http namespace.

Q13. Which HTTP verb does the HttpClient class use by default for the GetAsync method?
a) GET
b) POST
c) PUT
d) DELETE

Answer: a) GET

Q14. How do you deserialize the JSON response from a Web API into a .NET object?
a) By using the JsonSerializer class from the System.Text.Json namespace.
b) By using the JavaScriptSerializer class from the System.Web.Script.Serialization namespace.
c) By using the XmlSerializer class from the System.Xml.Serialization namespace.
d) By using the JsonConverter class from the Newtonsoft.Json namespace.

Answer: a) By using the JsonSerializer class from the System.Text.Json namespace.

Q15. How do you include custom headers in the request when consuming a Web API using HttpClient?
a) By setting the Headers property of the HttpClient instance.
b) By setting the Headers property of the HttpRequestMessage instance.
c) By setting the Headers property of the HttpResponseMessage instance.
d) By setting the Headers property of the HttpClientHandler instance.

Answer: b) By setting the Headers property of the HttpRequestMessage instance.

15.4. Using Newtonsoft APIs

Q16. What is Newtonsoft.Json?
a) It is a front-end framework for building user interfaces in web applications.
b) It is a data access technology that allows developers to work with databases using .NET objects.
c) It is a JSON library for serializing and deserializing JSON data in .NET applications.
d) It is a cloud computing service provided by Microsoft.

Answer: c) It is a JSON library for serializing and deserializing JSON data in .NET applications.

Q17. How do you serialize a .NET object to JSON using Newtonsoft.Json?
a) By using the JsonSerializer class.
b) By using the JavaScriptSerializer class.
c) By using the XmlSerializer class.
d) By using the JsonConverter class.

Answer: a) By using the JsonSerializer class.

Q18. What is JSON serialization?
a) It is the process of converting a .NET object into a JSON string.
b) It is the process of converting a JSON string into a .NET object.
c) It is the process of converting a .NET object into a binary format.
d) It is the process of converting a binary format into a .NET object.

Answer: a) It is the process of converting a .NET object into a JSON string.

Q19. How do you deserialize a JSON string to a .NET object using Newtonsoft.Json?
a) By using the JsonSerializer class.
b) By using the JavaScriptSerializer class.
c) By using the XmlSerializer class.
d) By using the JsonConverter class.

Answer: a) By using the JsonSerializer class.

Q20. What is JSON deserialization?
a) It is the process of converting a .NET object into a JSON string.
b) It is the process of converting a JSON

string into a .NET object.
c) It is the process of converting a .NET object into a binary format.
d) It is the process of converting a binary format into a .NET object.

Answer: b) It is the process of converting a JSON string into a .NET object.

15.3. Consuming using a client

Q21. When consuming a Web API, which of the following HTTP verbs is used to delete a resource?
a) GET
b) POST
c) PUT
d) DELETE

Answer: d) DELETE

Q22. What is the purpose of the HttpResponseMessage class in ASP.Net Web API?
a) It represents an HTTP response message that can be returned from a Web API method.
b) It represents an HTTP request message that is sent to a Web API method.
c) It represents an HTTP status code that indicates the success or failure of a Web API method.
d) It represents an HTTP response body that contains data returned from a Web API method.

Answer: a) It represents an HTTP response message that can be returned from a Web API method.

Q23. How can you handle exceptions when consuming a Web API using HttpClient in ASP.Net Core?
a) By using the try-catch block to catch HttpClientException.
b) By setting the ExceptionHandling property of the HttpClient instance.
c) By using the Global Exception Handler in the ASP.Net Core application.
d) By using the HttpResponseMessage.EnsureSuccessStatusCode() method.

Answer: d) By using the HttpResponseMessage.EnsureSuccessStatusCode() method.

Q24. What is the purpose of the HttpResponseMessage.Content property in ASP.Net Web API?
a) It represents the HTTP status code of the response.
b) It represents the content of the HTTP response message.
c) It represents the HTTP request message.
d) It represents the headers of the HTTP response.

Answer: b) It represents the content of the HTTP response message.

Q25. How can you pass query parameters to a Web API method when consuming it using HttpClient?
a) By adding the parameters directly to the HttpClient URL.
b) By using the [Query] attribute on the parameters in the Web API method.
c) By using the [FromBody] attribute on the parameters in the Web API method.
d) By setting the query parameters in the HttpResponseMessage.

Answer: a) By adding the parameters directly to the HttpClient URL.

Q26. What is the role of HttpClient in consuming a Web API in ASP.Net Core?
a) It represents the HTTP status code of the response.
b) It represents the content of the HTTP response message.
c) It sends HTTP requests to the Web API and receives HTTP responses.
d) It represents the HTTP request message.

Answer: c) It sends HTTP requests to the Web API and receives HTTP responses.

Q27. How can you handle errors or exceptions when consuming a Web API using HttpClient?
a) By using the try-catch block in the consuming code.
b) By setting the ExceptionHandling property of the HttpClient instance.
c) By using the [HandleError] attribute on the Web API controller.
d) By using the HttpResponseMessage.EnsureSuccessStatusCode() method.

Answer: a) By using the try-catch block in the consuming code.

Q28. When making a GET request using HttpClient to consume a Web API, how can you pass query parameters?
a) By adding the parameters directly to the URL.
b) By using the [FromQuery] attribute on the parameters in the Web API method.
c) By setting the query parameters in the HttpResponseMessage.
d) By using the [HttpGet] attribute on the Web API method.

Answer: a) By adding the parameters directly to the URL.

Q29. In an ASP.Net Core application, how can you set a custom timeout for HttpClient requests?
a) By setting the Timeout property of the HttpClient instance.
b) By using the [Timeout] attribute on the Web API method.
c) By using the [HttpGet] attribute on the Web API method.
d) By setting the Timeout property of the HttpRequestMessage.

Answer: a) By setting the Timeout property of the HttpClient instance.

Q30. What is the purpose of the HttpResponseMessage.Content property in ASP.Net Core Web API?
a) It represents the HTTP status code of the response.
b) It represents the content of the HTTP response message.
c) It represents the HTTP request message.
d) It represents the headers of the HTTP response.

Answer: b) It represents the content of the HTTP response message.

15.4. Using Newtonsoft APIs

Q31. What is the purpose of the JsonConvert.SerializeObject method in Newtonsoft.Json?
a) It is used to serialize a .NET object to a JSON string.
b) It is used to deserialize a JSON string to a .NET object.
c) It is used to serialize a .NET object to a binary format.
d) It is used to deserialize a binary format to a .NET object.

Answer: a) It is used to serialize a .NET object to a JSON string.

Q32. In Newtonsoft.Json, how can you ignore specific properties during JSON serialization?
a) By using the [JsonIgnore] attribute on the properties.
b) By using the [JsonProperty] attribute on the properties.
c) By using the [JsonConverter] attribute on the properties.
d) By using the [JsonRequired] attribute on the properties.

Answer: a) By using the [JsonIgnore] attribute on the properties.

Q33. How can you customize the JSON serialization and deserialization behavior in Newtonsoft.Json?
a) By creating a custom JsonConverter class.
b) By modifying the JsonSerializer settings directly.
c) By using the [JsonConverter] attribute on the properties.
d) By using the [JsonRequired] attribute on the properties.

Answer: a) By creating a custom JsonConverter class.

Q34. What is JSON deserialization in the context of Newtonsoft.Json?
a) It is the process of converting a .NET object into a JSON string.
b) It is the process of converting a JSON string into a .NET object.
c) It is the process of converting a .NET object into a binary format.
d) It is the process of converting a binary format into a .NET object.

Answer: b) It is the process of converting a JSON string into a .NET object.

Q35. Which method is used to deserialize a JSON string to a .NET object using Newtonsoft.Json?

a) SerializeObject
b) DeserializeObject
c) Parse
d) Stringify

Answer: b) DeserializeObject

Q36. What is Newtonsoft.Json?
a) It is a front-end framework for building user interfaces in web applications.
b) It is a data access technology that allows developers to work with databases using .NET objects.
c) It is a serialization and deserialization library for working with JSON data.
d) It is a cloud computing service provided by Microsoft.

Answer: c) It is a serialization and deserialization library for working with JSON data.

Q37. How can you specify the date format during JSON serialization using Newtonsoft.Json?
a) By using the DateTimeFormatInfo setting in the JsonSerializer.
b) By using the [JsonFormat] attribute on the property.
c) By using the [DateFormatString] attribute on the property.
d) By using the [JsonProperty] attribute on the property.

Answer: c) By using the [DateFormatString] attribute on the property.

Q38. In Newtonsoft.Json, how can you ignore specific properties during JSON serialization?
a) By using the [JsonIgnore] attribute on the property.
b) By using the [JsonProperty] attribute on the property.
c) By using the NullValueHandling setting in the JsonSerializer.
d) By using the [JsonRequired] attribute on the property.

Answer: a) By using the [JsonIgnore] attribute on the property.