Module: Basics of C#

1. Question: What are Windows Forms Applications in C#?

Answer: Windows Forms Applications are a type of application in C# that utilize the Windows Forms library to create graphical user interfaces (GUI) for desktop-based applications.

2. Question: Explain the concept of Classes and Objects in C#.

Answer: Classes are blueprints or templates that define the structure and behavior of objects. Objects, on the other hand, are instances of classes, representing real-world entities.

3. Question: Name some commonly used UI Controls in C#.

Answer: Some commonly used UI controls in C# include TextBox, Button, Label, ComboBox, CheckBox, RadioButton, etc.

4. Question: What is Inheritance, and how is it used in C#?

Answer: Inheritance is a mechanism in C# where one class (derived class) inherits properties and behaviors from another class (base class). It allows code reuse and promotes the concept of a hierarchical relationship between classes.

5. Question: Define Interfaces and their significance in C#.

Answer: Interfaces are a set of abstract methods and properties that define a contract that implementing classes must follow. They allow classes to provide common behavior without the need for inheritance, promoting loose coupling and flexibility in the code.

6. Question: How do Abstract Classes differ from Interfaces in C#?

Answer: Abstract classes can have both abstract and concrete methods, whereas interfaces can only have abstract methods. Also, a class can inherit from only one abstract class, but it can implement multiple interfaces.

7. Question: What are Indexers in C#? How do you use them?

Answer: Indexers are special properties in C# that allow objects to be indexed like arrays. They enable objects to be accessed using index notation, similar to arrays.

8. Question: Explain the concept of Strings Manipulation in C#.

Answer: String manipulation involves performing various operations on strings, such as concatenation, substring extraction, case conversions, searching, and replacing, among others.

Module: Introduction to ASP.NET

9. Question: How do you design simple web pages using ASP.NET?

Answer: To design simple web pages using ASP.NET, you can use HTML along with server-side controls to create dynamic content. The content and functionality can be defined in the code-behind files.

10. Question: What are the basic server-side controls in ASP.NET?

Answer: Basic server-side controls in ASP.NET include TextBox, Button, Label, DropDownList, RadioButton, CheckBox, etc.

11. Question: What is CrossPage, Postback, and Autopostback in ASP.NET?

Answer: CrossPage allows you to post data from one web page to another. Postback is the process of submitting an HTML form to the server. Autopostback automatically initiates a postback when the value of a control changes.

12. Question: Name some advanced web server controls and their uses.

Answer: Advanced web server controls in ASP.NET include Validation controls (e.g., RequiredFieldValidator, RegularExpressionValidator), Calendar, AdRotator (for displaying advertisements), Navigation controls (e.g., Menu, TreeView), and FileUpload (for uploading files).

13. Question: How can you build an application using AngularJS, jQuery, and NodeJS in ASP.NET?

Answer: You can build a web application using AngularJS for front-end development, jQuery for DOM manipulation and Ajax, and NodeJS for server-side JavaScript programming.

14. Question: Explain the concept of Master Pages in ASP.NET and how to create them.

Answer: Master Pages provide a consistent layout and design for multiple pages in an ASP.NET application. To create a Master Page, you create a separate ASP.NET file with the desired layout and content placeholders and use it as the master template for other pages.

15. Question: What are Themes and Skins in ASP.NET? How do you apply them?

Answer: Themes and Skins are used to apply a consistent look and feel to ASP.NET applications. Themes define a collection of styles, images, and skins for controls. Skins are specific to individual controls and override the styles defined in themes.

Module: Database Programming in ASP.NET

16. Question: What is the difference between Connected and Disconnected Architecture of ADO.NET?

Answer: Connected architecture involves keeping the connection to the database open while executing commands, while disconnected architecture involves opening a connection, retrieving data into a dataset, and then closing the connection.

17. Question: How do you work with Commands, Datasets, and Data Readers in ADO.NET?

Answer: Commands (SqlCommand, OleDbCommand, etc.) are used to execute SQL queries and commands on a database. Datasets (DataSet) are used to hold data retrieved from the database. Data Readers (DataReader) provide a fast, forward-only, read-only access to data.

18. Question: Explain the role of Data Adapters in ADO.NET.

Answer: Data Adapters (SqlDataAdapter, OleDbDataAdapter, etc.) act as bridges between a database and a dataset, facilitating the exchange of data between the two.

19. Question: How do you work with Stored Procedures in ADO.NET?

Answer: To work with stored procedures in ADO.NET, you create a Command object with the CommandType set to StoredProcedure and then execute the

stored procedure using the ExecuteNonQuery, ExecuteScalar, or ExecuteReader method of the Command object.

20. Question: What are Data-bound controls, and give examples of some in ASP.NET?

Answer: Data-bound controls are controls that can display data retrieved from a data source. Examples include DataList, DetailsView, FormView, GridView, ListView, and Repeater.

21. Question: What is LINQ, and how is it used with ASP.NET?

Answer: LINQ (Language-Integrated Query) is a set of features in C# that allows developers to query various data sources using a consistent syntax. LINQ can be used with ASP.NET to perform data querying and manipulation operations.

22. Question: How do you map data model to an object model using LINQ in ASP.NET?

Answer: LINQ provides a feature called Object Relational Mapping (ORM) that automatically maps database tables to objects in C#, allowing developers to work with data in an object-oriented manner.

Module: Session Management and AJAX

23. Question: Describe Client-Side State Management in ASP.NET.

Answer: Client-side state management involves storing data on the client's browser. Common techniques include ViewState, Query String, Cookies, and Hidden Fields.

24. Question: What are the various Server-Side State Management Techniques in ASP.NET?

Answer: Server-side state management techniques in ASP.NET include Profiles (Profile properties), Session State (Session variables), Application State (Application variables), and Cache (caching data in memory).

25. Question: How do you use AJAX in an ASP.NET application?

Answer: AJAX (Asynchronous JavaScript and XML) enables you to update parts of a web page without refreshing the entire page. In ASP.NET, AJAX can be implemented using UpdatePanels or client-side frameworks like jQuery and AngularJS.

26. Question: Name some AJAX controls in ASP.NET.

Answer: Some AJAX controls in ASP.NET include UpdatePanel, ScriptManager, UpdateProgress, and Timer.

27. Question: What is the purpose of the Global.asax and Web.config files in ASP.NET?

Answer: The Global.asax file contains events that are raised on application-level events, such as Application_Start, Session_Start, and Application_Error. The Web.config file contains configuration settings for the ASP.NET application.

28. Question: Explain Caching in ASP.NET and its benefits.

Answer: Caching involves storing frequently accessed data or pages in memory to reduce the load on the server and improve the application's performance. Caching can be done at the page level, data level, or output level.

29. Question: What are Web Parts, and how are they used?

Answer: Web Parts are reusable components that can be added to web pages dynamically. They allow users to customize the layout and content of a page according to their preferences.

Module: Web Services and WCF

30. Question: How do you create and consume an XML Web Service in ASP.NET?

Answer: To create an XML Web Service, you define methods in a Web Service class and decorate them with WebMethod attribute. To consume the web service, you add a reference to the service and call its methods.

31. Question: What are the steps to create and consume a WCF service in ASP.NET?

Answer: To create a WCF service, you define a contract using interfaces, implement the service, and host it in a suitable environment. To consume the service, you generate a proxy using the Service Reference tool and use it to call the service methods.

32. Question: Explain the differences between XML Web Services and WCF services.

Answer: XML Web Services are based on the SOAP protocol and use the ASMX technology, while WCF (Windows Communication Foundation) services are more flexible, support multiple protocols (e.g., HTTP, TCP, MSMQ), and provide better interoperability.

33. Question: What is Caching of Web Service responses? How can it be implemented?

Answer: Caching of Web Service responses involves storing the results of service calls to avoid redundant requests. It can be implemented using caching techniques available in ASP.NET or custom caching mechanisms in the service code.

Module: ASP.NET MVC

34. Question: What is ASP.NET MVC, and how is it different from traditional ASP.NET?

Answer: ASP.NET MVC is a design pattern that separates an application into three components: Model, View, and Controller. It differs from traditional ASP.NET (Web Forms) in terms of its architecture, control flow, and testability.

35. Question: Describe the process of designing an MVC application in ASP.NET.

Answer: Designing an MVC application involves creating models to represent data, views to display the user interface, and controllers to handle user input and business logic. The views are connected to the controllers, which interact with the models.

36. Question: How do you create a Simple Data-Entry Application with validations in ASP.NET MVC?

Answer: To create a Simple Data-Entry Application in ASP.NET MVC, you would create a model class to represent the data, a view with HTML form elements for data entry, and a controller to handle form submissions and data validation.

37. Question: Explain the concept of Automatically Implemented Properties in C#.

Answer: Automatically Implemented Properties, introduced in C# 3.0, allow you to declare properties without explicitly writing the backing field. The compiler generates the private backing field automatically.

38. Question: What are Object and Collection Initializers, and how do you use them?

Answer: Object and Collection Initializers are C# language features that allow you to initialize objects and collections in a concise and readable manner using a set of initial values.

39. Question: Define Extension Methods and provide an example of their usage.

Answer: Extension Methods allow you to add new methods to existing classes without modifying their source code. They must be defined in a static class and should be accessible through the instance of the extended class.

40. Question: How are Lambda Expressions used in C# and ASP.NET?

Answer: Lambda Expressions are used to write concise and inline delegate or function expressions. In ASP.NET, they are often used with LINQ to perform filtering, projection, and other data manipulation operations.

41. Question: Explain the concept of Forms and HTML Helpers in ASP.NET MVC.

Answer: HTML Helpers in ASP.NET MVC are methods that generate HTML code for common user interface elements. Forms are HTML elements that allow users to input data and submit it to the server.

42. Question: How do you define and access the model type in ASP.NET MVC?

Answer: The model type is specified when rendering a view using the `@model` directive. It allows you to strongly type the view, making it easier to work with the model data in the view.

43. Question: Describe how you can reduce duplication in views in ASP.NET MVC.

Answer: To reduce duplication in views, you can use partial views and layout pages. Partial views allow you to extract reusable pieces of the view, while layout pages define the common layout for multiple views.

44. Question: What is a default layout, and how do you specify it in ASP.NET MVC?

Answer: A default layout defines the common layout for views that do not specify a layout explicitly. In ASP.NET MVC, you can specify the default layout using the `_ViewStart.cshtml` file.

45. Question: How can you pass data values to the view from the controller in ASP.NET MVC?

Answer: Data values can be passed to the view from the controller using the `ViewData`, `ViewBag`, or `Model` objects. The `ViewData` and `ViewBag` are dynamic containers, while the `Model` object allows you to strongly type the data.

46. Question: How do you generate different content based on data values in ASP.NET MVC?

Answer: In ASP.NET MVC, you can use conditional statements (e.g., `if`, `switch`) and iteration (e.g., `foreach`, `for`) in the view to generate different content based on data values.

47. Question: Explain how to add a namespace to a view in ASP.NET MVC.

Answer: To add a namespace to a view in ASP.NET MVC, you can use the `@using` directive at the top of the view file to import the desired namespace.

Self-Learning Topics: Indexers and Strings Manipulations

48. Question: Explain the use of Indexers in C# with an example.

Answer: Indexers allow objects to be indexed like arrays, allowing you to access elements using index notation. For example, in a class representing a collection, an indexer can be used to access elements by their unique identifier.

49. Question: Discuss some common String Manipulation methods in C#.

Answer: Some common String Manipulation methods in C# include `Concat`, `Substring`, `ToUpper`, `ToLower`, `Replace`, `Trim`, `Split`, `Contains`, `IndexOf`, and `Format`.

Self-Learning Topics: Themes and Skins

50. Question: How do Themes and Skins impact the appearance of an ASP.NET application?

Answer: Themes and Skins allow you to apply a consistent look and feel to an ASP.NET application by defining styles, images, and other resources. Themes provide a collection of styles and images for the entire application, while skins are specific to individual controls.