Q. **What is Dot Net Framework?**

A. Dot.net Framework is the collection of tools and classes that are required to create a web development application.

The .NET Framework allows you to:

- Apply common skills across a variety of devices, application types, and programming tasks.
- Integrate with other tools and technologies to build the right solution with less work.
- Build compelling applications faster
- The Components of .NET Framework are:
  - **Common Language Runtime** – Handles the execution of code at Run Time.
  - **Base Class Libraries** – Pre-built code for common low-level programming tasks.
  - **Development frameworks and technologies** – reusable, customizable solutions for larger programming tasks.

Q. What is CLR.What are various responsibilities of CLR.

A. The .NET Framework provides a run-time environment called the common language runtime, which runs the code and provides services that make the development process easier. It handles the execution of Code at Run Time.

Compilers and tools expose the common language runtime's functionality and enable you to write code that benefits from this managed execution environment.

Garbage collection eliminates memory leaks as well as some other common programming errors. Code that runs under CLR environment is called Managed Code. If your code is managed, you can use managed data, unmanaged data, or both managed and unmanaged data in your .NET Framework application. Because language compilers supply their own types, such as primitive types, you might not always know (or need to know) whether your data is being managed.

The common language runtime makes it easy to design components and applications whose objects interact across languages. Objects written in different languages can communicate with each other, and their behaviors can be tightly integrated.

The CLR provides the following benefits:

- Performance improvements.
- The ability to easily use components developed in other languages.
- Extensible types provided by a class library.
- Language features such as inheritance, interfaces, and overloading for object-oriented programming.

WinSoft Question Booklet

Microsoft .Net

Introduction

- What is .Net?
- Why .Net
- Advantage of .Net
- D.N.A Architecture
  - One Tier
  - Two Tier
  - Three Tier
  - N-Tier

Understanding the .net Framework

- Need of Framework
- Type of Framework
  - .Net Framework
  - Mono Framework
- Version of Framework
- Introduction of CLR
- CLR Architecture
- Just In Time(JIT)Compiler
- Microsoft Intermediate Language(MSIL)
- Understanding MSIL with ILDASM
- Common Type System(CTS)
- Common Language Specification(CLS)
- Application Domain
- Framework Class Libraries(FCL)
- What is Assembly

C# Language Fundamentals

- Data Type in C#
  - Value Type
  - Reference Type
- CTS Type
- Nullable Types in C#
- Operation in C#
- Conditional, Looping , Jumping statement in C#
- Conversion Types:
  - Implicit Conversion

- Widening Conversion
- Narrowing Conversion
- Explicit Conversion
  - Casting
  - Convert Class

    Using Parse and ToString Method
- Exploring Visual 2010 IDE Environment
- Creating a Project and Solution
- Building Project
- Debugging Project

Object Oriented Programming System(OOPS)

- Concept of Classes and Object
- Understanding Oops Pillars
  - Inheritance
  - Polymorphism
  - Abstraction
  - Encapsulation
- Creating and using Class Libraries
- Creating and using Namespaces
- Access Modifier
- Understanding Instance and Static member in Class
- Type of method parameter in C#
- Optional parameters in C#
- Defining Properties and their type
- Events and delegates
- Understanding Delegates
  - Type of delegates
  - Simple delegates
  - Delegate implementation anonymous method
  - Multicast delegate
  - Generic delegate
  - Lambda operator and expression
- Understand Events
  - System define events
  - Events process cycle
  - Event handlers
  - Creating and binding users events
- Overview of Constructors
  - Default Constructor
  - Parameterized Constructor

    - Copy Constructor
    - Static Constructor
    - Private Constructor
    - Working with this keyword in Constructor
    - Implementing Singleton Design Pattern in C#
  - Concept of Polymorphism
    - Type of polymorphism
    - Late Binding and Early Binding
    - Method Overloading
    - Operator Overloading
- **Inheritance in C#**
  - What is Inheritance
  - Need of Inheritance
  - Type of Inheritance
  - Exposing Interface
    - Interface polymorphism
    - Play with Namespaces
    - Class Access Modifier
    - Partial Class
- **Generics Collection**
  - List
  - Dictionary
  - Stack
  - Queue
  - Linq
    - Introduction to var data type
    - Understanding the role of Linq
    - Linq query expression
    - Linq and non generic collection
    - Linq and generic collection
    - Plinq
- **File Handling**
  - System .IO Namespace
  - Stream reader and stream writer
  - File class
  - Directory class
  - Path class
  - Reading and writing a file
- **Forms and Control**
  - Building windows application

- Programming forms and control
- Constructing MDI form
- Implementation of MDI child and parent

- **Exception Handling**
  - Means of error, bug and exceptions
  - Role of .Net Exception Handling
  - System Level Exception
  - Application level exception
  - Try and catch block
  - Handling multiple exception
  - Finally block
  - Throw Exception on request
  - Custom exception
- **Reflection**
  - System.Reflection namespace
  - Core reflection classes
  - Accessing assembly through reflection
  - Methodinfo
  - Memberinfo
  - Propertyinfo
  - Browsing and querying member
  - Invoking methods and property and runtime

ADO .Net

- **Introduction of ADO.Net**
  - Evaluation of ADO
  - Difference between ADO
  - Providers
  - Connected and disconnected architecture
  - ADO.Net Architecture
    - Connection object
    - Command object
    - Dataset object
    - Data Reader object
    - Data adapter object
- **Working with Dataset**
  - Creating Tables in Dataset
  - Data column
  - Data Row

  - Data Table
  - Updating, Inserting and Modification in data table
  - Saving changes into Database using command builder
  - Working with command and connection
- **Understand command and connection classes**
  - Execute command with command object
  - Implementation and difference execute nonquery and execute scalar
  - Executing stored procedure
  - Creating parameters and SQL injection
- **Data Reader**
  - Reading data in connected environment
  - Moving and accessing value of record
  - Execute reader
- **Data View**
  - Row Filter
  - Sorting
  - Update, Insert, Delete via Dataview
- **Data Relation**
  - Creating relations
  - Apply relation on dataset
- **Adding Constraint**
  - Foreign key constraints
  - Unique constrains
  - Define Primary Key
- **Schema**
  - Schema type
  - Missing schema action
- **Transaction**
  - Introduction of Transaction(ACID)
  - Transaction class
  - Commit and rollback action
  - Implementation of save points
- **LINQ**
  - Linq to Dataset
    - Quering Typed/Untyped Datasets
    - Datarows and set operations
    - Creating new Data Tables SYLLABUS
    - Dataset-specific APIs
  - Linq to Sql
    - Creating and Querying Databases and Tables
    - Entity classes, Data context

- Creating/managing 1-many and many-1 relations
- Association & Foreign Keys
- Calls to Sprocs and Functions

- **Entity Framework**
  - Overview of the Entity Framework
  - Difference between Linq to Sql and Entity Framework
  - Learn how to create an Entity Data Model
  - Conceptual Schema Definition Language(CSDL)
  - Store Schema Definition Language(SSDL)
  - Mapping Specification Language (MSL)
  - Techniques for using LINQ to Entities to query data in a SQL Server Database
  - Creating/Managing 1-many and many-1 relation
  - Associations and Foreign Keys

- **WPF Internal**
  - WPF Overview
    - Understanding the motivation behind WPF
    - Examine the various ‘flavors’ of WPF application
    - Introduction of XAML
    - Work with the Window and Application class types
    - WPF Controls
    - Review the WPF control programming model
    - WPF Layouts
      - Stack Panel
      - Wrap Panel
      - Dock Panel
      - Grid
      - Canvas
      - Uniform Grid
    - Use of WPF Brushes
      - Solid Color Brush
      - Linear Gradient Brush
      - Radial Gradient Brush
      - Visual Brush
      - Image Brush
    - Transformation in WPF
      - Layout Transform
      - Render Transform
      - Rotate Transform
      - Scale Transform

      - Skew Transform
      - Translate Transform
      - Transformation Group
    - Style in WPF
    - Define style as Window Resource
    - Define style as Application Resource
    - Creating Resources Dictionary and Merging Dictionaries
    - WPF Data binding

ASP .Net

- **Introduction of asp.net**
  - Need of web application
  - Http protocol
  - Static and Dynamic page
  - Concept working of IIS and Browser
  - Differences between ASP and ASP.Net
- **Introduction of html**
  - Creating simple html page
  - Html tags
  - Hosting a html page
  - IIS virtual directory
  - Request transfer throws Get and Post methods

- **Server pages using ASP.Net**
  - Creating server pages
  - Architecture of handling request in webserver
  - Understanding page life cycle
  - Http request object
  - Http response object
  - Understanding post back
  - Auto event wire up
  - Tracing and Debugging ASP.Net page
- **Sate management**
  - Introduction of stateless process
  - Need of state persistence
  - Type of state Management
    - Http Cookies
      - Cookies Creation

- Cookies Types
  - Non Persistence
  - Persistence
  - Third Party
  - Single Value
  - Multi Value
- Scope of cookies and their merits

- **Http Session**
  - Understand the Session
  - Understand Session cleanup and creation
  - Implementation Session setting in Web.Config File
  - In process and out process Session Management
  - Session state using Sql Server
  - Session state mode
  - Session state using Sate Service
- **Http Application**
  - Understand Application Life Cycle
  - Understand Application cleanup and creation
  - Implementation and use application object
  - Working with Global.asax
- **Query String method**
  - Implementation of query string
  - Transfer values through Query String
  - Getting value by Request object
  - Scope
- **State management using Post Back Url**
  - Context handler class
  - Cross Page Postback
  - Understand Postback URL
  - Previous Page Process
- **Viewstate**
  - Understanding View State
  - Store value in View State
  - Retrieve Values from View State
  - Enable View Sate Property
- **State management using Hidden Field**
- **State management using Profile**
  - Creating Profile
  - Reading and Writing values in Profile

- Server Controls

- Standard Controls
  - Textbox
  - Combobox
  - Listbox
  - Checkbox
  - Addrotator control
  - File upload control
  - Multiview control
  - And many more
- Validation Controls
  - Types of Validation Controls
  - Working with Range, Ragular Expression, Compare
  - And Requred Field Validator Controls
  - Use Controlto validate Property
  - Custom Validator Control
  - Implementation of Client and Server Side Validation
  - Working with Validation Summary Control

Data Bound Controls

- Use of Data Binding Control
- Type of Data Binding Control (Formatted and Unformatted
- Working with repeater Control
  - Working with template
  - Working with repeater events
  - Understand Event Bubbling
- Working with Datalist Control
  - Working with template
  - Working with Datalist events
- Working with GridView Control
  - Data Binding with GridView
  - Working with Column Collection
  - Using paging
  - Sort
- Working with FormView Control
  - Working with template
  - Working with repeater events

- Working with DetailsView Control
  - Working with template
  - Working with repeater events
  - Understand Event Bubbling

Data Bound Controls

- Site Map

  - Define a site map file
  - Use of site map
  - Binding sitemappath control
- Tree View
  - Property of tree view
  - Define a node relation
  - Binding with site map
- Menu Control
  - Property of menu view
  - Define an items relation
  - Binding with sitemap

Designing Work with Look & Feel

- Skin – Theme and CSS
- Css
  - Creating style sheet
  - Apply Style Sheet
  - Creating Rules
  - Creating Elements setting
- Skin-Theme
  - Theme folder and Skin
  - Creating theme
  - Apply theme
  - Handling theme conflict
  - Using CSS in a Theme
  - Apply theme dynamically
- Master Pages
  - Introduction and need of master page
  - Creating a master page
  - Default contents
  - Master page with CSS and table layout
  - Appling master page through configuration file
  - Dynamic setting a master page
  - Nested master page
  - Master page and relative path
  - Designing Work with Look & Feel

Understand the Security Concepts

- Type of security

- Forms
  - Define a form base security
  - Applying security
  - Form Authentication class and their methods
  - Declare user into web configuration
  - Apply Password formats
  - Location attribute
  - Authentication
  - Authorization
- Windows
  - Applying window base security
  - IIS setting regarding windows authentication
  - Authentication
  - Authorization
- Passport
  - Concept of passport security
  - Passport Url

- **Login controls**
  - Configure forms authentication
  - Creating the data source
  - Configure Connection string and membership Provider
  - Creating a authentication user
  - Using security controls
    - Login control
    - Login status control
    - Login View Control
    - Password Recovery Control
    - Change Password Control
    - Create User Control
  - Creating and manage roles
  - Apply role in security
- **Performance tunning**
  - Caching
  - Introduction of caching
  - Need of caching
  - Type of caching
    - Page output Caching
      - Declarative output caching
      - Time duration and location setting
      - Parameter caching settings

- Page fragment Caching
  - User Control Caching
  - Time duration setting
  - Parameter caching settings
- Caching with HttpCachePolicyClass
- Post Cache Substitution & Fragment Caching
- Data Caching
  - Cache object
  - Managing data into cache
  - Cache Property
  - Cache Removal Callback
  - Cache profiles

Partial Postback mechanism

- Ajax
  - Introduction of Ajax
  - Creating ajax enabled website
  - Understanding and working with update panel
  - Role of script manager
  - Creating a ajax enabled web service

Culture Implementation

- Globalization and Localization
  - Introduction of multilingual application
  - Understanding Culture and UICulture
  - Setting the UICulture and Culture
  - CultureInfo Class
  - Overriding Initialize culture
  - Auto attribute
  - Local and Global resources

Cross Information Exchanging

- Introduction of xml
- Xml Documentation structure
- Well Formed Format of xml
- Parsers DOM and SAX
- Xml classes
- Reading and writing xml file

LINQ to XML (XLinq)

- Xdocument, XElement, XAttribute
- Loading, Parsing, Saving

    - Trees, Navigation, Updating
    - Retrieving and using Values
- **Deal with Dynamic Link Lib Hell**
    - Assembly and deployment
        - Role of .net assembly
        - Building and consume a single file assembly
        - Building and consume a multi file assembly
        - Private assembly
        - Shared assembly
        - Register assembly in GAC
        - Creating multiple version of a single assembly
- **Serialization**
    - Understand object serialization
    - Serialization formatters
    - Serialization
    - Binary formatters
    - Soap formatter
    - Xml serialization
- **Distributed application implementation**
    - D.N.A Implementations
        - Remoting
            - Introduction of remoting
            - Remoting architecture and scenario
            - Creating .net remoting object
            - Hosting a remote object
            - Activation type
                - Singleton
                - Singlecall
        - Creating channels
        - Implementations of a remote based application

- **Working with SOA**
- **Windows Communication Foundation (WCF)**
    - Programmatically Configuring a Managed Application to Host a WCF Service
    - Programmatically Configuring a Managed Application to Call a WCF Service
    - Defining Client and Service Settings by Using File-Based Configuration
    - Selecting a Hosting Option for a WCF Service Deploying a WCF Service
    - Exposing WCF Services over Different Endpoints
    - Adding Behaviors to Services and Endpoints
    - Interoperating with Non-WCF Web services

Working Custom AAOP with Dynamic

- Understanding the Dynamic Language Runtime
  - The dynamic type
  - The DLR Script Runtime
  - Dynamic Object
  - ExpandoObject

MVC 2.0

- ASP.NET MVC Special
- What's New in ASP.NET MVC 2.0?
  - MVC 2.0Model-View-Controller and AS P.NE
  - Serving Methods, Not Files
  - Is This Web Forms 4.0?
  - Why “(ASP.NET > ASP.NET MVC) == True”
  - Convention over Configuration
  - The Third Request Is the Charm
- Understanding Routes and URL s
  - Routing Compared to URL Rewriting
  - Defining Routes
- Controllers
  - Defining the Controller: The IController Interface
  - The Controller Class and Actions
  - Working with Parameters
  - Working with Multiple Parameters
- Views
  - Specifying a View
  - Strongly Typed Views
  - Using HTML Helper Methods
  - HtmlHelper Class and Extension Methods

Introduction to Silverlight

- The Evolution of the User Interface.
- Rich Internet Application(RIA) Solutions
- What Is Silverlight?
- Benefits of Silverlight
- The Silverlight Development Environment
- New Features in Silverlight 4

  - Building Your First Silverlight Application in Visual Studio

- **Layout Management in Silverlight 4**
  - The Canvas Panel
  - The StackPanel Control
  - The Grid Control
  - The WrapPanel Control
  - The DockPanel Control

- **Silverlight 4 Controls**
  - Setting Control Properties
  - Nesting Controls Within Controls
  - Handling Events in Silverlight
  - The Border Control
  - User Input Controls
  - Extended Controls.
  - AutoCompleteBox
  - ViewBox
  - Modal Windows

- **Data Binding**

  - Element to Element Binding
  - Binding with a data source
  - The DataGrid Control
  - Accessing Data Through a WCF Service

- **Styling in Silverlight**
  - Using Styles As Static Resources
  - Defining Styles at the Application Level
  - Merged Resource Dictionaries
  - Silverlight Style Hierarchy