Course Information

.Net is a software framework of Microsoft. It primarily runs on Windows. It is used by developers to develop new applications for Windows platform. ASP.NET is a server side web application framework that is developed from initial .NET release. With ASP.NET developers can now efficiently produce dynamic web pages and applications and help in providing web services. Following professionals can go for it:

- Fresher's
- IT Experts

This course will cover 12 months of training in which 80% of the training will be practical based with regular assignments and after completion of the training, a project will be given to the student and their evaluation will be based on their projects. Also regular tests and mock sessions on technical as well as on HR rounds will be a part of the curriculum. This course also includes soft skill development which will help students to perform better in interview.

Eligibility: Technical graduate having fundamental knowledge of any programming languages will be benefited but not mandatory.

Lecture Duration: 12 months

Placement: 100% Placement Assistance

Job Profile: Dotnet developer, Front end Developer

To know more about us, please visit www.aiiits.com

.NET Professional Track

Part 1: Introducing C# and the .NET Platform

The Philosophy of .NET

Understanding the Previous State of Affairs
The .NET Solution
Introducing the Building Blocks of the .NET Platform (the CLR, CTS, and CLS)
Additional .NET-Aware Programming Languages
An Overview of .NET Assemblies
Understanding the Common Type System
Understanding the Common Language Specification
Understanding the Common Language Runtime
The Assembly/Namespace/Type Distinction
Exploring an Assembly Using ildasm.exe
Exploring an Assembly Using Reflector
Deploying the .NET Runtime
The Platform-Independent Nature of .NET

Building C# Applications

The Role of the .NET Framework 4.0 SDK
Building C# Applications Using csc.exe
Building .NET Applications Using Notepad++
Building .NET Applications Using SharpDevelop
Building .NET Applications Using Visual C# 2010 Express
Building .NET Applications Using Visual Studio 2010

Part 2: Core C# Programming Constructs

Core C# Programming Constructs, Part I

The Anatomy of a Simple C# Program
An Interesting Aside: Some Additional Members of the System. Environment Class
The System. Console Class
System Data Types and C# Shorthand Notation
Working with String Data
Narrowing and Widening Data Type Conversions
Understanding Implicitly Typed Local Variables
C# Iteration Constructs
Decision Constructs and the Relational/Equality Operators

Core C# Programming Constructs, Part II

Methods and Parameter Modifiers
Understanding C# Arrays
Understanding the Enum Type
Understanding the Structure Type
Understanding Value Types and Reference Types
Understanding C# Nullable Types
Defining Encapsulated Class Types
Introducing the C# Class Type
Understanding Constructors
The Role of the this Keyword
Understanding the static Keyword

Defining the Pillars of OOP

C# Access Modifiers
The First Pillar: C#'s Encapsulation Services
Understanding Automatic Properties
Understanding Object Initializer Syntax
Working with Constant Field Data
Understanding Partial Types
Understanding Inheritance and Polymorphism The Basic Mechanics of Inheritance
Revising Visual Studio Class Diagrams
The Second Pillar of OOP: The Details of Inheritance
Programming for Containment/Delegation
The Third Pillar of OOP: C#'s Polymorphic Support
Understanding Base Class/Derived Class Casting Rules
The Master Parent Class: System. Object
Understanding Structured Exception Handling
Ode to Errors, Bugs, and Exceptions

The Role of .NET Exception Handling
The Simplest Possible Example
Configuring the State of an Exception
System-Level Exceptions (System. System Exception)
Application-Level Exceptions (System. Application Exception)
Processing Multiple Exceptions

The Result of Unhandled Exceptions
Debugging Unhandled Exceptions Using Visual Studio

Understanding Object Lifetime
Classes, Objects, and References
The Basics of Object Lifetime
The Role of Application Roots
Understanding Object Generations
Building Finalizable Objects
Building Disposable Objects
Building Finalizable and Disposable Types
Understanding Lazy Object Instantiation

Part 3: Advanced C# Programming Constructs
Working with Interfaces
Understanding Interface Types
Defining Custom Interfaces
Implementing an Interface
Invoking Interface Members at the Object Level
Interfaces As Parameters
Interfaces As Return Values
Arrays of Interface Types

Resolving Name Clashes via Explicit Interface Implementation
Designing Interface Hierarchies

Understanding Generics
The Issues with Non-Generic Collections
The Role of Generic Type Parameters
The System.Collections.Generic Namespace
Creating Custom Generic Methods
Creating Custom Generic Structures and Classes
Constraining Type Parameters
Delegates, Events, and Lambdas
Understanding the .NET Delegate Type
Defining a Delegate Type in C#
Understanding Delegate Covariance
Understanding Generic Delegates
Understanding C# Events
Understanding C# Anonymous Methods
Understanding Lambda Expressions
Advanced C# Language Features
Understanding Indexer Methods
Understanding Operator Overloading
Understanding Custom Type Conversions
Understanding Extension Methods
Understanding Partial Methods
Understanding Anonymous Types
Working with Pointer Types

LINQ to Objects
LINQ Specific Programming Constructs
Understanding the Role of LINQ
Applying LINQ Queries to Primitive Arrays
Returning the Result of a LINQ Query
Applying LINQ Queries to Collection Objects
Investigating the C# LINQ Query Operators
The Internal Representation of LINQ Query Statements

Programming with .NET Assemblies
Configuring .NET Assemblies
Defining Custom Namespaces
The Role of .NET Assemblies
Understanding the Format of a .NET Assembly
Building and Consuming a Single-File Assembly
Building and Consuming a Multifile Assembly
Understanding Private Assemblies
Understanding Shared Assemblies
Consuming a Shared Assembly
Configuring Shared Assemblies
Understanding Publisher Policy Assemblies
Understanding the <codeBase> Element
The System.Configuration Namespace
Dynamically Loading Assemblies
Reflecting on Shared Assemblies
Understanding Late Binding

Understanding the Role of .NET Attributes
Building Custom Attributes
Assembly-Level (and Module-Level) Attributes

The Role of a Windows Process
Interacting with Processes Under the .NET Platform
Understanding .NET Application Domains
Interacting with the Default Application Domain
Creating New Application Domains
Understanding Object Context Boundaries
Summarizing Processes, AppDomains, and Context
Understanding CIL and the Role of Dynamic Assemblies
Reasons for Learning the Grammar of CIL
Examining CIL Directives, Attributes, and Opcodes
Pushing and Popping: The Stack-Based Nature of CIL
Understanding Round-Trip Engineering
Understanding CIL Directives and Attributes

.NET Base Class Library, C#, and CIL Data Type Mappings
Defining Type Members in CIL
Examining CIL Opcodes
Building a .NET Assembly with CIL
Understanding Dynamic Assemblies
Dynamic Types and the Dynamic Language Runtime
The Role of the C# dynamic Keyword
The Role of the Dynamic Language Runtime (DLR)
Simplifying Late Bound Calls Using Dynamic Types
Simplifying COM Interoperability using Dynamic Data
COM Interop using C# 4.0 Language Features

Introducing the .NET Base Class Libraries
Multithreaded and Parallel Programming
The Process/AppDomain/Context/Thread Relationship
A Brief Review of the .NET Delegate
The Asynchronous Nature of Delegates
Invoking a Method Asynchronously
The System.Threading Namespace
The System.Threading.Thread Class
Programmatically Creating Secondary Threads
The Issue of Concurrency
Programming with Timer Callbacks
Understanding the CLR ThreadPool
Parallel Programming under the .NET Platform
Parallel LINQ Queries (PLINQ)

Summary

File I/O and Object Serialization

Exploring the System.IO Namespace

The Directory(Info) and File(Info) Types
Working with the DirectoryInfo Type
Working with the Directory Type
Working with the DriveInfo Class Type
Working with the FileInfo Class
Working with the File Type
The Abstract Stream Class
Working with StreamWriters and StreamReaders
Working with StringWriters and StringReaders
Working with BinaryWriters and BinaryReaders
Watching Files Programmatically

Understanding Object Serialization
Configuring Objects for Serialization
Choosing a Serialization Formatter
Serializing Objects Using the BinaryFormatter
Serializing Objects Using the SoapFormatter
Serializing Objects Using the XmlSerializer
Serializing Collections of Objects
Customizing the Soap/Binary Serialization Process

ADO.NET Part I: The Connected Layer
A High-Level Definition of ADO.NET
Understanding ADO.NET Data Providers
Additional ADO.NET Namespaces
The Types of the System.Data Namespace
Abstracting Data Providers Using Interfaces
Creating the AutoLot Database
The ADO.NET Data Provider Factory Model
Understanding the Connected Layer of ADO.NET

Working with Data Readers
Building a Reusable Data Access Library
Creating a Console UI–Based Front End
Understanding Database Transactions
ADO.NET Part II: The Disconnected Layer
Understanding the Disconnected Layer of ADO.NET
Understanding the Role of the DataSet
Working with DataColumns
Working with DataRows
Working with DataTables
Binding DataTable Objects to Windows Forms GUIs
Working with Data Adapters
Adding Disconnection Functionality to AutoLotDAL.dll
Multitabled DataSet Objects and Data Relationships
The Windows Forms Database Designer Tools
Isolating Strongly Typed Database Code into a Class Library
Programming with LINQ to DataSet

ADO.NET Part III: The Entity Framework
Understanding the Role of Entity Framework
Building and Analyzing your First EDM
Programming Against the Conceptual Model
AutoLotDAL Version 4.0, Now with Entities
Data Binding Entities to Windows Forms GUIs

Introducing LINQ to XML
A Tale of Two XML APIs
Members of the System.Xml.Linq Namespace
Working with XElement and XDocument
Manipulating an in Memory XML Document
Introducing Windows Communication Foundation
A Potpourri of Distributed Computing APIs

The Role of WCF
Investigating the Core WCF Assemblies
The Visual Studio WCF Project Templates
The Basic Composition of a WCF Application
The ABCs of WCF
Building a WCF Service
Hosting the WCF Service
Building the WCF Client Application
Simplifying Configuration Settings with WCF 4.0
Using the WCF Service Library Project Template
Hosting the WCF Service within a Windows Service
Invoking a Service Asynchronously from the Client.
Designing WCF Data Contracts
Introducing Windows Workflow Foundation 4.0

Defining a Business Process
Building a (Painfully) Simple Workflow
The WF 4.0 Runtime
Examining the Workflow 4.0 Activities
Building a Flowchart Workflow
Isolating Workflows into Dedicated Libraries
Consuming the Workflow Library

Building Desktop User Interfaces with WPF
Introducing Windows Presentation Foundation and XAML
The Motivation Behind WPF
The Various Flavors of WPF
Investigating the WPF Assemblies
Building a WPF Application without XAML
Building a WPF Application using Only XAML
Transforming Markup into a .NET Assembly
Understanding The Syntax of WPF XAML
Building a WPF Application using Code-Behind Files
Building WPF Applications Using Visual Studio 2010

Programming with WPF Controls
A Survey of the Core WPF Controls
Controlling Content Layout Using Panels
Building a Window's Frame Using Nested Panels
Understanding WPF Control Commands
Building a WPF User Interface with Expression Blend
Building the Ink API Tab
Introducing the Documents API
Building the Documents Tab
Introducing the WPF Data-Binding Model
WPF Graphics Rendering Services
Understanding WPF's Graphical Rendering Services
Rendering Graphical Data Using Shapes
WPF Brushes and Pens

Applying Graphical Transformations
Working with Shapes using Expression Blend
Rendering Graphical Data Using Drawings and Geometries
Generating Complex Vector Graphics using Expression Design
Rendering Graphical Data Using the Visual Layer
WPF Resources, Animations, and Styles
Understanding the WPF Resource System
Working with Object (Logical) Resources
Understanding WPF's Animation Services
Authoring Animations in XAML

Understanding the Role of WPF Styles
Generating Styles with Expression Blend
WPF Control Templates and User Controls
Understanding the Role of Dependency Properties
Building a Custom Dependency Property
Understanding Routed Events
Logical Trees, Visual Trees and Default Templates
Building a Custom Control Template with Visual Studio 2010
Building Custom user Controls with Blend
Creating the Jackpot Deluxe WPF Application
Part 7: Building Web Applications with ASP.NET
Building ASP.NET Web Pages
The Role of HTTP

Understanding Web Applications and Web Servers

The Role of HTML
The Role of Client-Side Scripting
Posting Back to the Web Server
The Feature Set of the ASP.NET API
Building a Single File ASP.NET Web Page
Building an ASP.NET Web Page using Code Files
ASP.NET Web Sites and ASP.NET Web Applications
The ASP.NET Web Site Directory Structure

The Inheritance Chain of the Page Type
Interacting with the Incoming HTTP Request
Interacting with the Outgoing HTTP Response
The Life Cycle of an ASP.NET Web Page
The Role of the Web.config File

ASP.NET Web Controls, Master Pages and Themes
Understanding the Nature of Web Controls
The Control and WebControl Base Classes
Major Categories of ASP.NET Web Controls
Building the ASP.NET Cars Web Site
The Role of the Validation Controls
Working with Themes

ASP.NET State Management Techniques
The Issue of State
ASP.NET State Management Techniques
Understanding the Role of ASP.NET View State
The Role of the Global.asax File
Understanding the Application/Session Distinction
Working with the Application Cache
Maintaining Session Data
Understanding Cookies
The Role of the <sessionState> Element
Understanding the ASP.NET Profile API

Web Development

HTML & CSS Overview
Introduction
HTML Basics
HTML Elements
HTML Attributes
HTML Styles
HTML Forms
HTML Form Elements
HTML Input Element Types
HTML Input Attributes
HTML File Paths
Script tag and its uses
HTML & XHTML
CSS Introduction
CSS Syntax
CSS Selectors
CSS Styling

Javascript Primer
Introduction to Javascript
Javascript Statements
Javascript Keywords
Javascript Functions
Javascript Programs
Javascript Operators
Function Parameters
Function Return Values
Javascript Data Types
Primitive Types

Working with Objects
Object Overview
Object Oriented Programming
Object creation
Adding Properties to Objects
Adding Methods to Objects
Javascript Conditional Statements
Javascript Loops & Iteration
Enumerating properties
Callbacks
JSON

Environmental setup
MVC Architecture
Model-View-Controller explained
Why MVC matters
MVC - the AngularJS way

First Application

Directives
Introduction to Directives
Directive lifecycle
Using AngularJS built-in directives
Binding controls to data
Matching directives
Creating a custom directive

Expressions

Controllers
Role of a Controller
Attaching properties and functions to scope
Nested Controllers

Using filters in Controllers
Controllers in External Files
Controllers & Modules

Filters
Built-in filters
Using AngularJS filters
Creating custom filters
Tables

HTML DOM
Modules
Introduction to AngularJS Modules
Bootstrapping

Forms
Working with Angular Forms
Model binding
Form controller
Validating Angular Forms
Form events
Updating models with a twist
$error object

Scope
What is scope
Scope lifecycle
Two way data binding
Scope inheritance
Scope & controllers
Scope & directives
$apply and $watch
Rootscope
Scope broadcasting

Dependency Injection & Services
What is Dependency Injection
Using Dependency Injection
What are services
Creating services
Factory, Service & Provider
Using AngularJS built in services

Single Page Application(SPA)
What is SPA
Pros & Cons of SPA

Installing the ngRoute module
Configure routes
Passing parameters
Changing location
Resolving promises
Create a Single Page Application

Angular-X

Getting Started

1. Course Introduction.
2. What is Angular?
3. Angular vs Angular 2 vs Angular 4+
4. Project Setup and First App.
5. Editing the First App.
6. The Course Structure.
7. What is TypeScript.
8. A Basic Project Setup using Bootstrap for Styling

The Basics

How an Angular App gets Loaded and Started
Components
- Using Custom Components
- Creating Components with the CLI & Nesting Components
- Working with Component Templates
- Working with Component Style

Practicing Components
What is Databinding
- String Interpolation
- Property Binding
- Property Binding vs String Interpolation
- Event Binding
- Bindable Properties and Events
- Passing and Using Data with Event Binding
- Two-Way-Databinding

Important: FormsModule is Required for Two-Way-Binding
- Combining all Forms of Databinding

Practicing Databinding
- Understanding Directives
- Using ngIf to Output Data Conditionally
- Enhancing ngIf with an Else Condition.

Components & Databinding Deep Dive
- Splitting Apps into Components
- Property & Event Binding Overview
- Binding to Custom Properties
- Assigning an Alias to Custom Properties

Binding to Custom Events
Assigning an Alias to Custom Events
Custom Property and Event Binding Summary
Understanding View Encapsulation
More on View Encapsulation
Using Local References in Templates.
Getting Access to the Template & DOM with @ViewChild.
Understanding the Component Lifecycle
Lifecycle Hooks.
Assignment 3: Practicing Property & Event Binding and View Encapsulation

Directives Deep Dive
ngFor and ngIf
ngClass and ngStyle
How to create a Basic Directive
Using Services & Dependency Injection
Why would you Need Services?
Creating a Logging Service
Injecting the Logging Service into Components
Creating a Data Service
Understanding the Hierarchical Injector
How many Instances of Service Should It Be?
Injecting Services into Services
Using Services for Cross-Component Communication

Practicing Services

Changing Pages with Routing
Why do we need a Router?
Understanding the Example Project.
Setting up and Loading Routes.
Navigating with Router Links.
Understanding Navigation Paths.
Styling Active Router Links.

Navigating Programmatically.
Using Relative Paths in Programmatic Navigation
Passing Parameters to Routes
Fetching Route Parameters
Fetching Route Parameters Reactively
An Important Note about Route Observables
Passing Query Parameters and Fragments
Retrieving Query Parameters and Fragments
Setting up Child (Nested) Routes
Using Query Parameters - Practice
Configuring the Handling of Query Parameters
Redirecting and Wildcard Routes
Important: Redirection Path Matching

- An Introduction to Guards
- Protecting Routes with canActivate
- Protecting Child (Nested) Routes with canActivateChild
- Using a Fake Auth Service

Controlling Navigation with canDeactivate.

- Handling Forms in Angular Apps
- Template-Driven (TD) vs Reactive Approach
- TD Forms

Practicing Template-Driven Forms.

- Introduction to the Reactive Approach
- Reactive Forms
- Assignment 6: Practicing Reactive Forms.
- Using Pipes to Transform Output
- Introduction & Why Pipes are Useful
- Using Pipes
- Making Http Requests
- Introduction & How Http Requests Work in SPAs
- Sending Requests (Example: POST Request)
- Adjusting Request Headers
- Sending GET Requests
- Sending a PUT Request
- Transform Responses Easily with Observable Operators (map())
- Using the Returned Data
- Catching Http Errors
- Using the "async" Pipe with Http Requests.

Authentication & Route Protection in Angular Apps

- How Authentication Works in Single-Page-Applications.
- Creating a Signup Page and Route
- Setting up the Firebase SDK
- Signing Users Up
- Signin Users In
- Requiring a Token (on the Backend)
- Sending the Token
- Checking and Using Authentication Status
- Adding a Logout Button
- Route Protection and Redirection.
- Wrap Up
- The HttpClient (ANGULAR 5 Addition Bonus SECTION)
- Request Configuration and Response.
- Requesting Events.
- Setting Headers.
- Interceptors.

SQL - Structure Query Language

RDBMS - An Introduction
Database
Relational Database Systems
Working with the Book's Sample Database
SQL: A Relational Database Language
Normal Forms
Entity-Relationship Model
Syntax Conventions

Foundations of T-SQL

A Short History of T-SQL
Imperative vs. Declarative Languages
SQL Basics
Statements
Databases
Transaction Logs
Schemas
Tables
Views
Indexes
Stored Procedures
User-Defined Functions
SQL CLR Assemblies
Elements of Style
Whitespace
Naming Conventions
One Entry, One Exit
Defensive Coding
SQL-92 Syntax Outer Joins
The SELECT * Statement
Variable Initialization

Procedural Code and CASE Expressions

Three-Valued Logic
Control-of-Flow Statements
The BEGIN and END Keywords
The IF...ELSE Statement
The WHILE, BREAK, and CONTINUE Statements
The GOTO Statement
The WAITFOR Statement
The RETURN Statement
The TRY...CATCH Statement
The CASE Expression
The Simple CASE Expression
The Searched CASE Expression

CASE and Pivot Tables
COALESCE and NULLIF
Cursors

User-Defined Functions

Scalar Functions
Recursion in Scalar User-Defined Functions
Procedural Code in User-Defined Functions
Multistatement Table-Valued Functions
Inline Table-Valued Functions
Restrictions on User-Defined Functions
Nondeterministic Functions
State of the Database

Stored Procedures

Introducing Stored Procedures
Calling Stored Procedures
Managing Stored Procedures
Stored Procedures in Action
Recursion in Stored Procedures
Table-Valued Parameters
Temporary Stored Procedures
Recompilation and Caching
Stored Procedure Statistics
Parameter Sniffing
Recompilation

Triggers

DML Triggers
When to Use DML Triggers
Auditing with DML Triggers
Nested and Recursive Triggers
The UPDATE and COLUMNS_UPDATED Functions
Triggers on Views
DDL Triggers
Logon Triggers

Common Table Expressions and Windowing Functions

Common Table Expressions
Multiple Common Table Expressions
Recursive Common Table Expressions
Windowing Functions
The ROW_NUMBER Function
The RANK and DENSE_RANK Functions
The NTILE Function
Aggregate Functions and OVER

XML

- Legacy XML
- OPENXML
- OPENXML Result Formats
- FOR XML Clause
- FOR XML RAW
- FOR XML AUTO
- FOR XML EXPLICIT
- FOR XML PATH
- The xml Data Type
- Untyped xml
- Typed xml
- The xml Data Type Methods
- The query Method
- The value Method
- The exist Method
- The nodes Method
- The modify Method
- XML Indexes
- XSL Transformations

XQuery and XPath

- XPath and FOR XML PATH
- XPath Attributes
- Columns Without Names and Wildcards
- Element Grouping
- The data Function
- XPath and NULL
- The WITH XMLNAMESPACES Clause
- Node Tests
- XQuery and the xml Data Type
- Expressions and Sequences
- The query Method
- Location Paths
- Node Tests
- Namespaces
- Axis Specifiers
- Dynamic XML Construction
- XQuery Comments
- Data Types
- Predicates
- Conditional Expressions (if...then...else)
- Arithmetic Expressions
- XQuery Functions
- Constructors and Casting
- FLWOR Expressions

Catalog Views and Dynamic Management Views

- Catalog Views
- Table and Column Metadata
- Index Metadata
- Querying Permissions
- Dynamic Management Views and Functions
- Session Information
- Connection Information
- Currently Executing SQL
- Tempdb Space
- Server Resources
- Unused Indexes
- INFORMATION_SCHEMA Views

SQL CLR Programming

- The Old Way
- The SQL CLR Way
- SQL CLR Assemblies
- User-Defined Functions
- Stored Procedures
- User-Defined Aggregates
- Creating a Simple UDA
- Creating an Advanced UDA
- SQL CLR User-Defined Types

New T-SQL Features

- Set Operators
- The OUTPUT Clause
- The TOP Keyword
- CROSS APPLY and OUTER APPLY
- The TABLESAMPLE Clause
- The NEWSEQUENTIALID Function
- Date and Time Functions
- The max Data Types
- Synonyms
- FILESTREAM Support
- Enabling FILESTREAM Support
- Creating FILESTREAM Filegroups
- FILESTREAM-Enabling Tables
- Accessing FILESTREAM Data

Error Handling and Dynamic SQL

- Error Handling
- Legacy Error Handling
- Try...Catch Exception Handling
- The RAISERROR Statement
- Debugging Tools
- PRINT Statement Debugging
- Trace Flags
- SSMS Integrated Debugger

Visual Studio T-SQL Debugger
Dynamic SQL
The EXECUTE Statement
SQL Injection and Dynamic SQL
Troubleshooting Dynamic SQL
The sp_executesql Stored Procedure
Dynamic SQL and Scope
Client-Side Parameterization

Performance Tuning

SQL Server Storage
Files and Filegroups
Space Allocation
Data Compression
Indexes
Heaps
Clustered Indexes
Nonclustered Indexes
Filtered Indexes
Optimizing Queries
Reading Query Plans
Methodology

Software Testing

MANUAL TESTING

Software Development Life Cycle:

- What are the different phases of SDLC?
- How does the process of Software Development Start?
- Project Initiation

Requirement Gathering and Analysis

- What is Requirement document and what it contains?
- What is use case document and what it contains?
- What is Basic path and Alternate Path?
- Role of Business Analyst
- Example for explaining each phase
- Role of technical specification team
- What is Technical specification document?

What is System Design?

- Role of Design team
- What is design document?
- Role of architecture team

System development

- Role of development team
- Deliverable of Development phase

System testing

- Role of testers and types of testing
- User acceptance testing
- System deployment

System maintenance

- Events in the maintenance phase like bug fixes

Software Testing Life Cycle

- How are the phases of STLC carried out?
- What is testing?
- Role of testers
- Why do we need to test?
- Activities involved in the testing phase

What is test plan and test case document?

- Steps of test case execution
- What does test case document contain?
- How to write test case document?
- What is required to test any application?

TEST CASES

- What is test case?
- What does test case document contain?
- How to write test case document?
- Different test case techniques

TEST PLAN

- What is Test Plan?
- How to write test plan document?
- What does the test plan document contain?
- Who writes and approves the test plan document?
- How manage the test case documents?
- What is the pass/fail criterion?

TYPES OF TESTING

- Different Phases of testing
- What is unit testing?

- What is Minimum acceptance testing?
- What is integration, system and system integration testing?
- What is User acceptance testing?
- What is Regression Testing?

DEFECT ANALYSIS

- What is a defect?
- Various Defect tracking tools
- How to use the defect tracking tools?
- How to enter the details of defect in the defect tracking tool?
- How to identify a defect?
- What is severity and priority?

TRACEABILITY MATRIX

- What is Traceability Matrix[TM]?
- Who Prepares the TM document?
- What is the reference for writing TM?
- What is the use of TM?
- What is present in the TM document?
- Sample TM
- Tools used for developing TM








WHAT STUDENT FEEL ABOUT US:

Subarna Mukherjee
1 review

★★★★★ a week ago

I am working on database in a IT company, i started python in iiht-kharghar, i liked their environment, sincerity and professional approach then i upgarded myself for R, machine learning and Hadoop. I am happy to be here.

Ashish Ravi
1 review

★★★★★ a week ago -

It was an amazing experience. I got to learn so many things. The trainers are extremely knowledgeable and are very friendly, love their way of teaching, it was very practical, excellent training pattern. Batch timing and course duration is flexible. This is one of the best institute for learning database and programming languages. Also provides placements and helps to get placed in good companies. I would highly recommend this institute to others to help move their career forward.

Pranay Gadhave
1 review

★★★★★ a month ago

I joined IIHT kharghar for the core Java language.
The experience was superb.
The faculties are good and very helpful.

Vedant Pathak
1 review

★★★★★ a month ago -

Surely a good place to learn about programming, staff is great and responsive.
Overall it's a good place

rahul chouhan
2 reviews

★★★★☆ a month ago -

Definitely a good place for courses related Software & Hardware. A good add-on to your skills and CV. The certificates provided a worth it. I have pursued Python from here. Now comes to Facilities, they are so good. The step in your shoes n make you understand it.

shaligram wagh
1 review

★★★★★ a month ago

I join IIHT Kharghar for CCNA Networking classes, here I learn lot of things regarding networking and clear my all confusions, classes teachers are really good , teaches very friendly , and all staff and HR department are very helpful

Didar Hossain
Local Guide · 13 reviews · 3 photos

★★★★★ a month ago

I registered for Angular and refresher for HTML/CSS/JavaScript/PHP. Faculty for web technologies is knowledgeable and helpful. Angular faculty was a professional from the industry whose competence level was extraordinary. *But* students need to be serious and willing to put their efforts to make use of such talented faculties - no place for slackers.

fahad datey
1 review

★★★★★ 3 months ago -

i completed my cloud training, the training was very good and trainer explain each module practically and i got placed very good company

thank u iiht for giving right carrier path

YOU CAN FIND OUR STUDENTS IN: