

# Introduction to Dot Net

## Dot Net Beginners: Level 1

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 4 weeks

### Course Summary

The DTC – Dot Net course is targeted for beginners who want to learn how to think and write meaningful pieces of codes or read Dot Net codes written by someone else. This course teaches how to map literary description of a problem (requirement) to an application/library coded in Dot Net. This is a core basic level course that is essential for anyone who has no prior programming experience but wishes to be a professional Dot Net engineer in future.

### Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

### Required Text Books

1. Pro C #9 with Dot Net 5
2. The Ultimate Beginner's guide to learn C#

### Prerequisites

- Basic knowledge about programming, bits/bytes, procedures, classes, computer architecture, etc. If you just have a theoretical knowledge that is perfectly okay but you should have strong convictions on what programming is, and what you hope to achieve from this class.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Dot Net (self-study and practice).
- There is no prior educational level requirement for this course. Anyone from 10+2 student to someone who is doing their PHD in Genetic Engineering is welcome to take this course.
- If you are only interested in theory and have no interest/patience in spending at least 10 hours every week throughout the duration of the course, then this course might not be for you.
- If you have absolutely no idea about programming or do not see yourself doing programming in the next six -odd months, then this class may not be for you!

## Course Details

### WEEK 1

#### **STARTING WITH VISUAL STUDIO 2010**

- Creating Console Application Project
- Project vs. Solution
- How to compile the project?

#### **THE MAIN() METHOD – WHERE IT ALL BEGINS**

#### **CLASS IN BRIEF**

#### **WHAT IS A METHOD?**

- Argument List
- Return Type
- Breaking down solution (to a problem) to one or more methods

#### **NAMESPACE**

- Alias
- Global scope

#### **VARIABLES**

- BASE CLASSES – BASIC
- How to write to Console?
- How to read from Console?

### WEEK 2

#### **FLOW CONTROL**

- What is it and why do we need it?

#### **FLOW CONTROL - CONDITIONAL STATEMENTS**

- If-Else
- Switch

#### **FLOW CONTROL – ITERATION AND JUMPS**

- For
- While
- Do While
- Break
- Continue
- Go to

- For each

## **OPERATORS – BASIC**

- Arithmetic
- Increment/Decrement
- Comparison
- Logical
- Bitwise
- Bit Shifting
- Assignment

## **SCOPE OF A VARIABLE**

## **CONSTANTS**

## **DATA TYPES**

- Value Types
- Reference Types

## **WEEK 3**

### **VALUE TYPES IN DETAIL**

- Signed vs. Unsigned
- byte, sbyte
- short, ushort
- int, uint
- long, ulong
- float
- double
- decimal
- bool
- char

### **ARRAYS**

- Single Dimensional Arrays, introducing the [] operator
- Multiple Dimensional Arrays
- Jagged Arrays
- Array is a Reference Type

### **STRING**

- String is a Reference Type
- What is a string made up of – understanding the char type
- Ways of constructing strings

- How to copy one string to another
- Using the [] operator
- Converting strings from lower case to upper and vice-versa
- Searching for specific characters in a string
- Searching for specific words in a sentence
- Complex String operations
- String Builder
- Format Strings
- Immutability

## WEEK 4

### **ENUMERATIONS**

#### **PREPROCESSOR DIRECTIVES**

- Define
- Undef
- If, Elif, Else, Endif
- Warning
- Error
- Region, Endregion

#### **COMPILING WITH MULTIPLE MAIN() METHODS**

#### **VISUAL STUDIO 2010 REVISITED – HOW TO DEBUG YOUR CODE**

#### **STRUCTURES**

#### **HOW TO EFFECTIVELY DESIGN AND WRITE YOUR OWN CLASSES?**

#### **WHERE TO GO NEXT?**

### **LABS**

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

## Intermediate Dot Net: Level 2

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 4 weeks

### Course Summary

The DTC – Dot Net – Level 2 course is designed for students who have some prior hands-on programming experience with the Dot Net programming language at a beginning level. This course is ideal for people who have previously programmed in another programming language (e.g., Java, Obj-C, PHP, C, C++, etc.) and wish to learn Dot Net. This course is designed for high school and university students who want to do Dot Net coursework, including those who are already working as a professional VB.NET developer and want to switch to Dot Net, as well as those who have worked in the media industry since graduation or are working as a professional freelance PHP developer.

### Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

### Required Text Books

3. Pro C #9 with Dot Net 5
4. The Ultimate Beginner's guide to learn C#
5. C# 8.0 in a nutshell

### Prerequisites

- Successfully complete the entrance test with score of at least 40% (for trainees directly applying to this level).
- Successfully complete the DWIT Training – Dot Net – Level 1 course (not applicable to trainees directly applying to this level).
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Dot Net (self-study and practice).

### Course Details

#### WEEK 1

#### **CLASS IN DETAIL**

- Data Members

- Function Members
- Access Modifiers
- Data Encapsulation
- Set and Get methods
- Passing parameters by Value
- Passing parameters by Reference
- Using keywords ref, out, and params in methods
- Named Arguments
- Optional Arguments
- Method Overloading
- Properties and Accessory
- Constructors and Destructors
- Partial Classes
- Static Class and Static Methods
- Static Constructor, read only fields

## WEEK 2

### **OBJECT ORIENTED PROGRAMMING**

- Implementation Inheritance
- Interface Inheritance
- Multiple Inheritance
- The Object class
- Polymorphism, Virtual Methods
- Abstract Class and Abstract Methods
- Sealed Class
- Exceptions
- Structures Revisited
- Using Constructors
- Inheritance
- Coding Conventions and Guidelines
- Properly Writing Comments

## WEEK 3

### **ADVANCED**

- Use of operators – checked, unchecked
- Use of operators – as, is, type of, unsafe, size of
- Null able types and operations
- Null Coalescing operator
- Type Inference
- Anonymous Type
- Boxing and Unboxing

- Data Conversions – Implicit and Explicit
- Four ways to compare objects for equality
- Operator Overloading
- User Defined Casts
- Indexers
- Generics
- Type Safety
- Constraints
- Default
- Inheritance
- Interface
- Statics
- Structures
- Delegates
- Covariance
- Contra-variance
- Delegates and Events

## WEEK 4

### **WRITING WINDOWS FORMS APPLICATIONS**

- What makes Windows Forms application different from Console Application
- Human Computer Interaction
- Slight Diversion: The WinMain() loop
- Class Hierarchy
- Controls
  - User Interface
  - User Interaction
  - How it all works
- Forms class
- Standard Controls
  - Button
  - Check Box
  - Checked List Box
  - Combo Box
  - Data Grid View
  - Date Time Picker
  - Error Provider
  - Image List
  - Label
  - List Box
  - List View
  - Masked Text Box

- Menu Strip
- Panel
- Picture Box
- Progress Bar
- Radio Button
- Rich Text Box
- Tab Control
- Tab Pages
- Text Box

## **LABS**

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.



## Advanced Dot Net: Level 3

Nature of the course: Theory + Practical

Total hours per day: 2 hours

Course duration: 4 weeks

### Course Summary

This course expands on the DTC – Dot Net – Level 2 foundation and offers advanced subjects to equip learners for a career as an Android software engineer.

### Completion Criteria

After fulfilling all of the following criteria, the student will be deemed to have finished the Module:

1. Has attended 90% of all classes held
2. Has received an average grade of 80% on all assignments
3. Has received an average of 60% in assessments
4. The tutor believes the student has grasped all of the concepts and is ready to go on to the second module.

### Required Text Books

1. The Busy Coder's Guide to Advanced Android Development
2. Android Design Patterns
3. Android Programming: Pushing the Limits.

### Prerequisites

- Successfully completed the DWIT Training – Dot Net – Level 2 or obtained at least 40% score on the entrance exam.
- The latter case applies for new students that are directly attempting this training.
- Successfully complete the interview.
- Willing and eager to spend at least 10-20 hours (varying from student-to-student) per week outside of the training class to read/write codes in Dot Net (self-study and practice).
- Please note that this is a lab intensive course where the students will be expected to work on lab exercises for approximately half the duration of the session.

### Course Details

#### WEEK 1

#### **DATA STRUCTURES**

- Introduction
- Key Interfaces
- Array Class
- Lists
  - Queue
  - Stack

- Linked List
- Sorted List
- Dictionaries
- Sets
- Bit Arrays
- Trees
- Graphs
- Deciding which data structure to use
- Thinking about performance

## **MANAGING FILES**

- File I/O Revisited
- Serialize / De-serialize
- Moving, Copying and Deleting Files
- Collecting Drive Information
- Memory Mapped File

## WEEK 2

### **WORKING WITH XML**

- Standards in .NET
- XML I/O
- DOM
- XPATH

### **INSTRUMENTATION**

- Event Logging
- Tracing
- Perfmon
- Contracts

### **WRITING MULTITHREADED APPLICATION**

- Asynchronous Delegates
- Thread Class
- Thread Pools
- Tasks
- Race Conditions and Deadlock
- Lock statement
- Wait Handle
- Mute x
- Semaphore
- Timers

## WEEK 3

### WRITING NETWORK APPLICATIONS

#### REFLECTION AND ASSEMBLIES

- DCustom Attributes
- System.Type
- Assembly Class
- Overview of Assemblies
- What constitutes Assemblies?
- Structure
- Manifest
- Attributes
- Private, Shared, Satellite
- Creating and Loading Assemblies
- Application Domains
- Versioning
- GAC
- Shared Assemblies
- Strong Names
- Delayed Signing

## WEEK 4

### WINDOWS FORMS

- Multiple Document Interface
- User Controls

### INTRODUCTION TO ASP.NET.

- What is ASP.NET?
- ASP.NET Life-cycle explained
- Web Sites vs. Web Applications
- Managing States
- Coding Models
- ASP.NET Web Forms
- ASP.NET Server Controls

### LABS

Lab assignments will focus on the practice and mastery of contents covered in the lectures; and introduce critical and fundamental problem-solving techniques to the students.

### Learning Outcomes

- Web Application Configuration and Deployment

- Create a safe online application
- Understand the ASP.NET page structure and the Microsoft.NET Framework
- Create a web application with a wide range of controls
- Learn how to use the features of the Dot Net Framework as well as the features of C#, access the data using inbuilt data access tools
- Perform database operations for Windows Form and web applications