# Document Revise History

|  |  |  |
| --- | --- | --- |
| Owner | Update Date | Description |
| v-weibzh | 1/11/2016 | First draft |

**Summary**

This roadmap illustrates technique skills and knowledge required to support VCS forums. To be a qualified support engineer you need to finish level 100 and level 200 at least. You can then read level 250+ materials in your daily work to improve your technique skills gradually. It’s not mandatory to read the optional materials in the readiness but it’s recommended to learn in your free time to develop yourself.

**Prerequisite Skills and Knowledge**

* C# Syntax

## VC# Level 0-150: (9 days)

**Purpose:**

After this period, following technical expertise should be built up: C# basic programming skill, .Net Framework Concept.

**Material:**

**C# Programming Guide:** (5 days)

[Inside a C# Program](https://msdn.microsoft.com/en-us/library/0cs3416d.aspx)

[Namespaces](https://msdn.microsoft.com/en-us/library/0d941h9d.aspx)

[Classes and Structs](https://msdn.microsoft.com/en-us/library/ms173109.aspx)

[Main() and Command-Line Arguments](https://msdn.microsoft.com/en-us/library/acy3edy3.aspx)

[Strings](https://msdn.microsoft.com/en-us/library/ms228362.aspx)

[Types](https://msdn.microsoft.com/en-us/library/ms173104.aspx)

[Arrays](https://msdn.microsoft.com/en-us/library/9b9dty7d.aspx)

[Enumeration Types](https://msdn.microsoft.com/en-us/library/cc138362.aspx)

[Nullable Types](https://msdn.microsoft.com/en-us/library/1t3y8s4s.aspx)

[Statements, Expressions, and Operators](https://msdn.microsoft.com/en-us/library/ms173142.aspx)

[Properties](https://msdn.microsoft.com/en-us/library/x9fsa0sw.aspx)

[Interfaces](https://msdn.microsoft.com/en-us/library/ms173156.aspx)

[Indexers](https://msdn.microsoft.com/en-us/library/6x16t2tx.aspx)

[Iterators](https://msdn.microsoft.com/en-us/library/dscyy5s0.aspx)

**C# Reference:** (2 days)

[C# Keywords](https://msdn.microsoft.com/en-us/library/x53a06bb.aspx)

[C# Operators](https://msdn.microsoft.com/en-us/library/6a71f45d.aspx)

[C# Preprocessor Directives](https://msdn.microsoft.com/en-us/library/ed8yd1ha.aspx)

[C# Compiler Options](https://msdn.microsoft.com/en-us/library/2fdbz5xd.aspx)

[C# Compiler Errors](https://msdn.microsoft.com/en-us/library/ms228296.aspx)

[C# Language Specification](https://msdn.microsoft.com/en-us/library/ms228593.aspx)

**Others** (2 days)

[Debugging,Tracing and Profiling](http://msdn.microsoft.com/en-us/library/7fe0dd2y.aspx)

[.NET Framework Tool](http://msdn.microsoft.com/en-us/library/d9kh6s92.aspx)

[C# GENERAL FAQ](http://social.msdn.microsoft.com/Forums/en-US/csharpgeneral/thread/2d666562-ed08-4461-bf92-7808913b4e96)

<http://v.itcast.cn/course/28-0-0-1.html>

**Exercise:**

1.Write a program to define three variables of Double type. Respectively, input values to them from the keyboard and then use Console.WriteLine method to output them in a row. Align decimal points and retain 3 decimals.

2.Write a program to input the three numbers from the keyboard. Using the ternary operator (? :) to find out maximum number.

3.Write a program to input a character. If the character is capital letters, please convert it to lowercase letters. If it is not, please do not convert.

4.Write a program to input the 6 numbers from the keyboard. And then find the maximum number.

5.Write a program to input a character. If the input character is uppercase, convert it to lowercase. If the input character is lowercase letters, please convert it to uppercase.

6.Write a program and defined structure (including school, name, gender and programming scores four fields). And then the declare variable of struct-type. After assigning statement to a value, output.

7.Write a program to input a positive number. The number is rounded to the nearest digit calculations. For example, the real number 12.56 obtain results 13 after rounding operation and 12.46 get the results 12 after rounding operation.

8.Write a program using do-while loop to calculate the PI=4\* (1-1/3+1/5+...+1/n). It will stop when the time is 1/n<0.000001.

9.Write a program to input two real numbers from the keyboard. Use the Max () and Min () in Math class to find the maximum and minimum number of them.

10.Make a program that prints out all of the "Narcissus". The "Narcissus" is a three-digit number. The cube of number is equal to the number. For example, 153=1\*1\*1+5\*5\*5+3\*3\*3, so 153 is " “Narcissus”.

11.Write a program to use while statement. Input the user name and password to enable user login function. It allows you to enter up three times at most. It does not allow to log on more than three times.

12.Write a program that requires using a while loop statement to print the positive integers from 1 to 100. Each line has 5 numbers to print and each column align to the right.

13.Write a program using for loops. Calculate the sum of 1!+2!+3!+...+10!.

14.Write a program to input 10 real number from keyboard and put them into an array. And using bubble sort to sort the numbers in ascending order.

15.Write a program to define an array. Input 10 real numbers into this array with the for statement. And then store in reverse chronological order and then output.

16.Enter a string that will change lowercase letters to uppercase letters, and uppercase letters into lowercase letters, characters remaining unchanged, output the string.

17. There are two kinds of members of a supermarket, gold and Platinum members.

Two members have different discount methods. Gold member discount is descease, such as 200 minus 10.

Platinum member discount is the discount method. For example, to have a 90 percent.

Discounts for each Member are different.

Please Simulate to write the classes of members, the gold medal of members, the platinum members of classes. To input the number of members and the sum price. And then output the price with discant.

18.Write the three user classes (including user number, user name), character classes (including part number role name) and function classes (including feature number, function name).

The relationship between the three classes is one user could have more characters and one characters could have more functions.

Output the user number and function number to determine whether this user can implement this function.

19.Get the area code and telephone number from given string "0510-88102566".

20.Write a method to exchange the value of two strings.

21. Let users to input names of 5 students, separated by a space between the name. And then output the names of all student surnamed Wang.

22. Find out all the mail address and replace it to the links.

  <a href="<mailto:406545545@qq.com>">[406545545@qq.com</a](mailto:406545545@qq.com%3c/a)>

23.The information of student has three attributes, such as ID, name, email. To implement the increase, find (keyword search), deleted (number deleted), and modified (modified as number) of the student’s information in the console. Please note the correctness and completeness of the data, such as mailbox format, repeat of number.

24.Write a generic Tree type to express all the tree structure.

**Checkpoint:**

1. Do some demo code with the knowledge learn every day.
2. 10 cases on MSDN about C# language specific issues.
3. Finish above **exercise**

## VC# Level 150-250: (14 days)

**Purpose:**

After this period, following technical expertise should be built up: delegates and events. Generics concept, LINQ Related program skills, GC related concept, XML related concept.

**Material:**

[Delegates](https://msdn.microsoft.com/en-us/library/ms173171.aspx)

[Events](https://msdn.microsoft.com/en-us/library/awbftdfh.aspx)

[Generics](https://msdn.microsoft.com/en-us/library/512aeb7t.aspx)

[Generics in the .NET Framework](https://msdn.microsoft.com/en-us/library/ms172192.aspx)

[LINQ Query Expressions](https://msdn.microsoft.com/en-us/library/bb397676.aspx)

[Lambda Expressions](https://msdn.microsoft.com/en-us/library/bb397687.aspx)

[XML Documentation Comments](https://msdn.microsoft.com/en-us/library/b2s063f7.aspx)

[Garbage Collection](https://msdn.microsoft.com/en-us/library/0xy59wtx.aspx)

[File and Stream I/O](https://msdn.microsoft.com/en-us/library/k3352a4t.aspx)

[File System and the Registry](https://msdn.microsoft.com/en-us/library/2kzb96fk.aspx)

[How to: Serialize and Deserialize JSON Data](https://msdn.microsoft.com/en-us/library/bb412179(v=vs.110).aspx)

[Mapping Between JSON and XML](https://msdn.microsoft.com/en-us/library/bb924435(v=vs.110).aspx)

[What's New in the .NET Framework](https://msdn.microsoft.com/en-us/library/ms171868(v=vs.110).aspx)

[New Features in C# 6.0](https://channel9.msdn.com/Events/Visual-Studio/Connect-event-2014/116)

<http://v.itcast.cn/course/28-0-0-2.html>

**Checkpoint:**

1. Delegate demo
2. Event demo
3. Generics demo
4. LINQ Query demo

## VC# Level 250+: (14 days)

**Purpose:**

After this period, following technical expertise should be built up: reflection, Thread, Asynchronous program skills, interop and unsafe code concept.

**Material:**

[Reflection](http://msdn.microsoft.com/en-us/library/f7ykdhsy.aspx)

[Threading Programming](http://msdn.microsoft.com/en-us/library/ms173178.aspx)

[Asynchronous Programming Overview](http://msdn.microsoft.com/en-us/library/ms228963.aspx)

[Asynchronous Programming with Async and Await](https://msdn.microsoft.com/en-us/library/hh191443.aspx)

[Interoperability](http://msdn.microsoft.com/en-us/library/ms173184.aspx)

[Unsafe Code and Pointers](https://msdn.microsoft.com/en-us/library/t2yzs44b.aspx)

[An Overview of Managed/Unmanaged Code Interoperability](https://msdn.microsoft.com/en-us/library/ms973872.aspx)

<http://v.itcast.cn/course/28-0-0-3.html>

**Checkpoint:**

1. Reflection demo
2. Threading demo
3. Asynchronous demo

# Document Revise History

|  |  |  |
| --- | --- | --- |
| Owner | Update Date | Description |
| v-pixi | 1/12/2016 | First draft |

**Summary**

This roadmap illustrates technique skills and knowledge required to support CLR forums. To be a qualified support engineer you need to finish level 100 and level 200 at least. You can then read level 250+ materials in your daily work to improve your technique skills gradually. It’s not mandatory to read the optional materials in the readiness but it’s recommended to learn in your free time to develop yourself.

**Prerequisite Skills and Knowledge**

* C++ Syntax, Win32 programming basics (message mechanism, Process and Thread management, memory management, DLL and static library, and etc.)
* C# basic
* VB basic

## CLR Level 0-150: (7 days)

**Purpose:**

After this period, following technical expertise should be built up: CLR, .Net Framework concept.

**Material:**

[Debugging Managed Code](http://msdn.microsoft.com/en-us/library/awtaffxb.aspx)

[Debugger Roadmap](http://msdn.microsoft.com/en-us/library/k0k771bt.aspx)

[.NET Framework Conceptual Overview](http://msdn.microsoft.com/en-us/library/zw4w595w.aspx)

[CLR Overview](http://msdn.microsoft.com/en-us/library/ddk909ch.aspx)

[CLR](https://msdn.microsoft.com/en-us/library/8bs2ecf4(v=vs.110).aspx)

[Unicode](http://msdn.microsoft.com/en-us/library/dd374081(VS.85).aspx)

[GAC](http://msdn.microsoft.com/en-us/library/yf1d93sz.aspx)

[Strong Name Assembly](http://msdn.microsoft.com/en-us/library/wd40t7ad.aspx)

[Boxing and Unboxing](http://msdn.microsoft.com/en-us/library/yz2be5wk.aspx)

[New features in .Net Framework 4](-%20http:/msdn.microsoft.com/en-us/library/ms171868.aspx)

[.NET Framework 4.6 and 4.5](https://msdn.microsoft.com/en-us/library/w0x726c2(v=vs.110).aspx)

[.NET Framework Versions and Dependencies](https://msdn.microsoft.com/en-us/library/bb822049(v=vs.110).aspx)

**CLR new feature**

[.NET Core and Open-Source](https://msdn.microsoft.com/en-us/library/dn878908(v=vs.110).aspx)

[Looking back – motivating .NET Core](http://blogs.msdn.com/b/dotnet/archive/2014/12/04/introducing-net-core.aspx)

[Welcome to .NET Core](https://dotnet.github.io/)

**Checkpoint:**

1. Do some demo code with the knowledge learn every day.
2. 10 cases on MSDN about C# language specific issues.

## VC# Level 150-250: (14 days)

**Purpose:**

After this period, following technical expertise should be built up: Exception related, GC, CLR Binder, CLR Thread Pool.

**Material:**

[Reliability](http://msdn.microsoft.com/en-us/library/ms172337.aspx)

[using Reliability feature](http://msdn.microsoft.com/en-us/magazine/cc163716.aspx)

[The Exception Model](http://blogs.msdn.com/b/cbrumme/archive/2003/10/01/51524.aspx)

[Interrupts and Exceptions](http://www.internals.com/articles/protmode/interrupts.htm)

[Using GC Efficiently – Part 1](http://blogs.msdn.com/b/maoni/archive/2004/06/15/156626.aspx)

[Using GC Efficiently – Part 2](http://blogs.msdn.com/b/maoni/archive/2004/09/25/234273.aspx)

[Using GC Efficiently – Part 3](http://blogs.msdn.com/b/maoni/archive/2004/12/19/327149.aspx)

[Using GC Efficiently – Part 4](http://blogs.msdn.com/b/maoni/archive/2005/05/06/415296.aspx)

[In-Process Side-by-Side Execution](http://msdn.microsoft.com/en-us/library/ee518876.aspx)

[CLR Inside Out: In-Process Side-by-Side](http://msdn.microsoft.com/en-us/magazine/ee819091.aspx)

[CppHostCLR](http://1code.codeplex.com/wikipage?title=All-In-One%20Code%20Framework%20Examples)

[Understanding The CLR Binder](http://msdn.microsoft.com/en-us/magazine/dd727509.aspx)

[How the Runtime Locates Assemblies](http://msdn.microsoft.com/en-us/library/yx7xezcf.aspx)

[Debugging Assembly Loading Failures](http://blogs.msdn.com/b/suzcook/archive/2003/05/29/57120.aspx)

[64-bit Applications](http://msdn.microsoft.com/en-us/library/ms241064.aspx)

[The CLR's Thread Pool](http://msdn.microsoft.com/en-us/magazine/cc164139.aspx)

[Understanding Classic COM Interoperability With .NET Applications](http://www.codeproject.com/Articles/990/Understanding-Classic-COM-Interoperability-With-NE)

FAQ:

[Base Class Library FAQs](http://social.msdn.microsoft.com/Forums/en-US/netfxbcl/thread/011d7cc0-52db-46c1-b66b-49afc0149c59)

[Chinese CLR FAQs (The 3rd reply)](http://social.msdn.microsoft.com/Forums/zh-CN/2212/thread/f2a90155-c760-4d5d-ba2e-6b0e427cb6eb)

**Checkpoint:**

1. Tech Talk: GC, Thread pool

## VC# Level 250+:(14 days)

**Purpose:**

After this period, following technical expertise should be built up: CLR Advanced, Debugging, Profiling.

**Material:**

1. CLR Advanced

[Drill Into .NET Framework Internals to See How the CLR Creates Runtime Objects](http://msdn.microsoft.com/en-us/magazine/cc163791.aspx)

[.NET Type Internals - From a Microsoft CLR Perspective](http://www.codeproject.com/Articles/20481/NET-Type-Internals-From-a-Microsoft-CLR-Perspecti)

2. Debugging

[.NET Debugging Demos - Information and setup instructions](http://blogs.msdn.com/b/tess/archive/2008/02/04/net-debugging-demos-information-and-setup-instructions.aspx)

[Getting started with WinDbg - part I](http://blogs.msdn.com/b/johan/archive/2007/11/13/getting-started-with-windbg-part-i.aspx)

[Getting started with WinDbg - part II](http://blogs.msdn.com/b/johan/archive/2007/11/26/getting-started-with-windbg-part-ii.aspx)

[(Tool) Debugging Tools for Windows (WinDbg)](http://msdn.microsoft.com/en-us/windows/hardware/gg463009.aspx)

[(Tool) CLR Profiler for .NET Framework 2.0](http://www.microsoft.com/en-us/download/details.aspx?id=13382)

[(Tool) CLR Profiler for .NET Framework 4.0](http://www.microsoft.com/en-us/download/details.aspx?id=16273)

[(Tool)CLRProfiler 4.5 released](http://clrprofiler.codeplex.com/)

[(Tool)CLRProfiler 4.5 released: includes Windows Store app support](http://blogs.msdn.com/b/davbr/archive/2012/11/19/clrprofiler-4-5-released-includes-windows-store-app-support.aspx)

3. Profiling

[Under the hood - The .NET Profiling API and the DNProfiler Tool](http://msdn.microsoft.com/en-us/magazine/cc301725.aspx)

[.NET Internals: The Profiling API](http://www.blong.com/Conferences/DCon2003/Internals/Profiling.htm)

[Debugging (Unmanaged API Reference)](http://msdn.microsoft.com/en-us/library/ms404520.aspx)

[.net tools](http://msdn.microsoft.com/en-us/library/d9kh6s92.aspx)

**Checkpoint:**

1. Tech Talk: .Net Framework, CLRProfiler, API