

Financial Engineering Mathematics

財務工程數學

NTUST/First Semester, 2019

昀騰金融科技

Wintom Financial Technology

技術長 CTO

董夢雲 博士 Dr. Andy Dong

dongmy@ms5.hinet.net

Contents

1. Introduction to QuantLib Projects
2. Black-Scholes Model and Equity Option Calculation
3. Black 76 Model and IRS、Caps/Floors、Swaptions Calculation

昀騰金融科技股份有限公司

技術長
金融博士、證券分析師

董夢雲 Andy Dong



ID:50917111
Line/WeChat:andydong3137
E:andydong1209@gmail.com
<https://github/andydong1209>
M:(T)0988-065-751 (C)1508-919-2872
10647 台北市大安區辛亥路一段 50 號 4 樓

學經歷

國立台灣大學電機工程學系學士
國立中央大學財務管理學研究所博士
中國信託商業銀行交易室研發科主管
凱基證券風險管理部主管兼亞洲區風險管理主管
中華開發金控、工業銀行風險管理處處長
永豐金控、商業銀行風險管理處處長
永豐商業銀行結構商品開發部副總經理

專業

證券暨投資分析人員合格(1996)
台灣金融研訓院 2019 年菁英講座

專長

風險管理理論與實務，資本配置與額度規劃、資產負債管理實務
外匯與利率結構商品評價實務，股權與債權及衍生商品評價實務
GPU 平行運算與結構商品系統開發，CUDA、OpenCL
CPU 平行運算與 ALM 系統開發，C#/C++/C、.Net Framework、SQL
人工智慧(Deep Learning)交易策略開發，Python、Keras、TensorFlow

1. Introduction to QuantLib Projection

1.1 Pricing tools and pricing software

- ◆ In every financial institution, we buy & sell financial instruments every day.
 - include listed securities(TSE, TFE).
 - ✓ Stock, Warrant, Options, Futures
 - OTC securities(OTC, Banks).
 - ✓ IRS, CCS, FX Forward, FRA, Caps/Floors, Swaptions
 - ✓ FX Structured Products, Rate Structured Products
- ◆ We need a system, a calculator, or a program to do these things, at least.
 - We need to evaluate our PL, Mark-to-Market(MTM) these instruments
 - We also need to estimate their risk, VaR.
 - Sometime we need to hedge our position risk dynamically, Delta, Vega, Gamma, DV01.
 - Also, we want to arbitrage among them.

◆ In every financial institution, some programs are used for these purpose.

- Different choose results in different solution, but all spend a lot of money.
- But no one solves these problems successfully.
 - ✓ Recently, more complex products are developed,
 - ✓ Regulators require more strict risk management standard of FI.
- As a FE, we may face the challenge and equip ourselves some coding skill.
 - ✓ Do the dirty jobs, develop your programs with best toolkits.
 - ✓ Every FI needs such skilled workers.

1.1.1 Alternatives

◆ In-house libraries (Matlab, C++)

- Hard to use.
- Not standard.
- Highly customizable.
- Expensive(Matlab).
- Quick bug resolution.

◆ Commercial libraries (NumeriX, FinCad)

- Well documented.
- Very expensive(NumeriX).
- Source code not available.
- Slow bug resolution

◆ Open-source library (QuantLib)

- Not-so-well documented.
- Highly customizable.
- Source code available.
- Quick bug resolution.

◆ Data-provider terminals (Bloomberg)

- Easy to use.
- Hard to customize.
- Only interactive (no batch)

◆ Commercial pricing services (StatPro, Markit)

- Easy to use.
- Hard to customize.
- Moderately expensive.

1.2 Why use QuantLib?

- ◆ Available since November 2000.
- ◆ Completely free (as in no charge, no constraints)
- ◆ On average about 15,000 downloads each year.
- ◆ Used all-over the world (five continents)
 - Major banks in Taiwan use QuantLib.
- ◆ Implements the standard financial models
- ◆ Advanced and experimental implementations
- ◆ Great Excel interface (QuantLibXL)
- ◆ Object-oriented library

1.3 QuantLib Project

- ◆ The QuantLib project is aimed at providing a comprehensive software framework for quantitative finance.
 - QuantLib is a free/open-source library for modeling, trading, and risk management in real-life.
- ◆ Appreciated by quantitative analysts and developers, it is intended for academics and practitioners alike, eventually promoting a stronger interaction between them.
 - QuantLib offers tools that are useful both for practical implementation and for advanced modeling, with features such as
 - ✓ market conventions, yield curve models,
 - ✓ solvers, PDEs, Monte Carlo (low-discrepancy included),
 - ✓ exotic options, VAR, and so on.

◆ Finance is an area where well-written open-source projects could make a tremendous difference:

- any financial institution needs a solid, time-effective, operative implementation of cutting edge pricing models and hedging tools.
 - ✓ However, to get there, one is currently forced to re-invent the wheel every time.
 - ✓ Even standard decade-old models, such as Black-Scholes, still lack a public robust implementation.
 - ✓ As a consequence many good quants are wasting their time writing C++ classes which have been already written thousands of times.
- By designing and building these tools in the open, QuantLib will both encourage peer review of the tools themselves, and demonstrate how this ought to be done for scientific and commercial software.
 - ✓ Dan Gezelter's [talk](#) at the first Open Source/Open Science conference discussed how the scientific tradition of peer review fits well with the philosophy of the Open Source movement.
 - ✓ Open standards are the only fair way for science and technology to evolve.

- ◆ The library could be exploited across different research and regulatory institutions, banks, software companies, and so on. Being a free/open-source project, quants contributing to the library would not need to start from scratch every time.
 - Students could master a library that is actually used in the real world and contribute to it in a meaningful way. This would potentially place them in a privileged position on the job market.
 - Researchers would have a framework at hand, which vastly reduces the amount of low-level work necessary to build models, so to be able to focus on more complex and interesting problems.
 - Financial firms could exploit QuantLib as base code and/or benchmark, while being able to engage in creating more innovative solutions that would make them more competitive on the market.
 - Regulatory institutions may have a tool for standard pricing and risk management practices.

檔案 (F) 編輯 (E) 檢視 (V) 歷史 (S) 書籤 (B) 工具 (T) 說明 (H)

QuantLib, a free/open-source library for quantitative finance

QuantLib

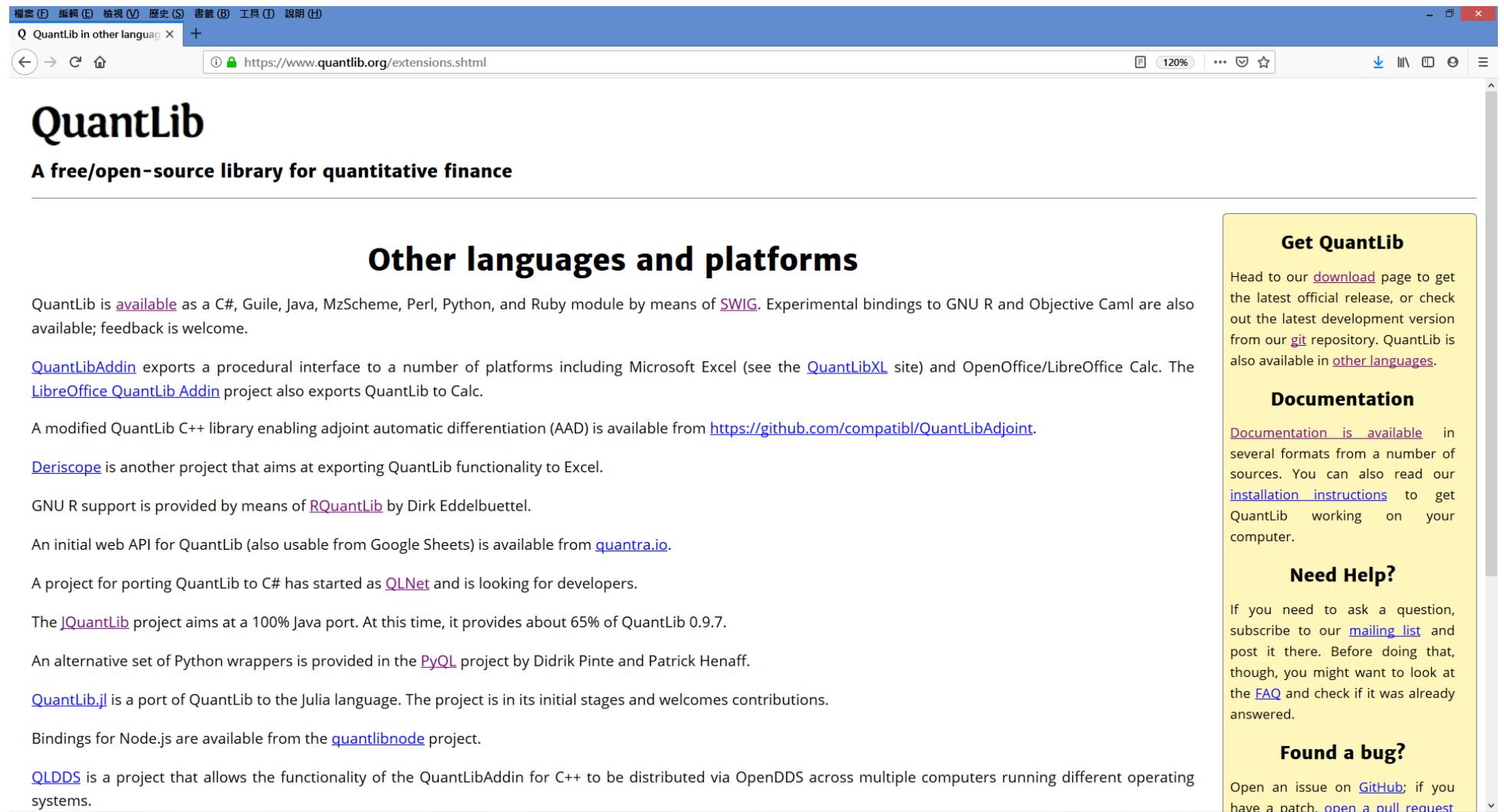
A free/open-source library for quantitative finance

| Get QuantLib | Documentation | Need Help? |
|--|--|--|
| Head to our download page to get the latest official release, or check out the latest development version from our git repository. QuantLib is also available in other languages . | Documentation is available in several formats from a number of sources. You can also read our installation instructions to get QuantLib working on your computer. | If you need to ask a question, subscribe to our mailing list and post it there. Before doing that, though, you might want to look at the FAQ and check if it was already answered. |
| Found a bug? | Want to contribute? | More info |
| Open an issue on GitHub ; if you have a patch, open a pull request instead. | Just fork our repository on GitHub and start coding (instructions are here). Please have a look at our developer intro and guidelines . | Here is the QuantLib license , the list of contributors , and the version history . |

The QuantLib project is aimed at providing a comprehensive software framework for quantitative finance. QuantLib is a [free/open-source](#) library for modeling, trading, and risk management in real-life. QuantLib is written in C++ with a clean object model, and is then exported to different languages such as C#, Objective Caml, Java, Perl, Python, GNU R, Ruby, and Scheme. An [AAD-enabled version](#) is also available. The [repository](#) project facilitates deployment of object libraries to end user platforms and is used to generate [QuantLibXL](#), an Excel addin for QuantLib, and [QuantLibAddin](#), QuantLib addins for other platforms such as LibreOffice Calc. See the [extensions](#) page for details on bindings and ports to other languages.

Appreciated by quantitative analysts and developers, it is intended for academics and practitioners alike, eventually promoting a stronger interaction between them. QuantLib offers tools that are useful both for practical implementation and for advanced modeling, with features such as market conventions, yield curve models, solvers, PDEs, Monte Carlo (low-discrepancy included), exotic options, VAR, and so on.

- ◆ QuantLib is written in C++ with a clean object model, and is then exported to different languages such as
 - C#, Objective Caml, Java, Perl, Python, GNU R, Ruby, and Scheme. An [AAD-enabled version](#) is also available.
 - The [repository](#) project facilitates deployment of object libraries to end user platforms and is used to generate [QuantLibXL](#), an Excel addin for QuantLib, and [QuantLibAddin](#),
 - ✓ QuantLib addins for other platforms such as LibreOffice Calc.
 - See the [extensions](#) page for details on bindings and ports to other languages.



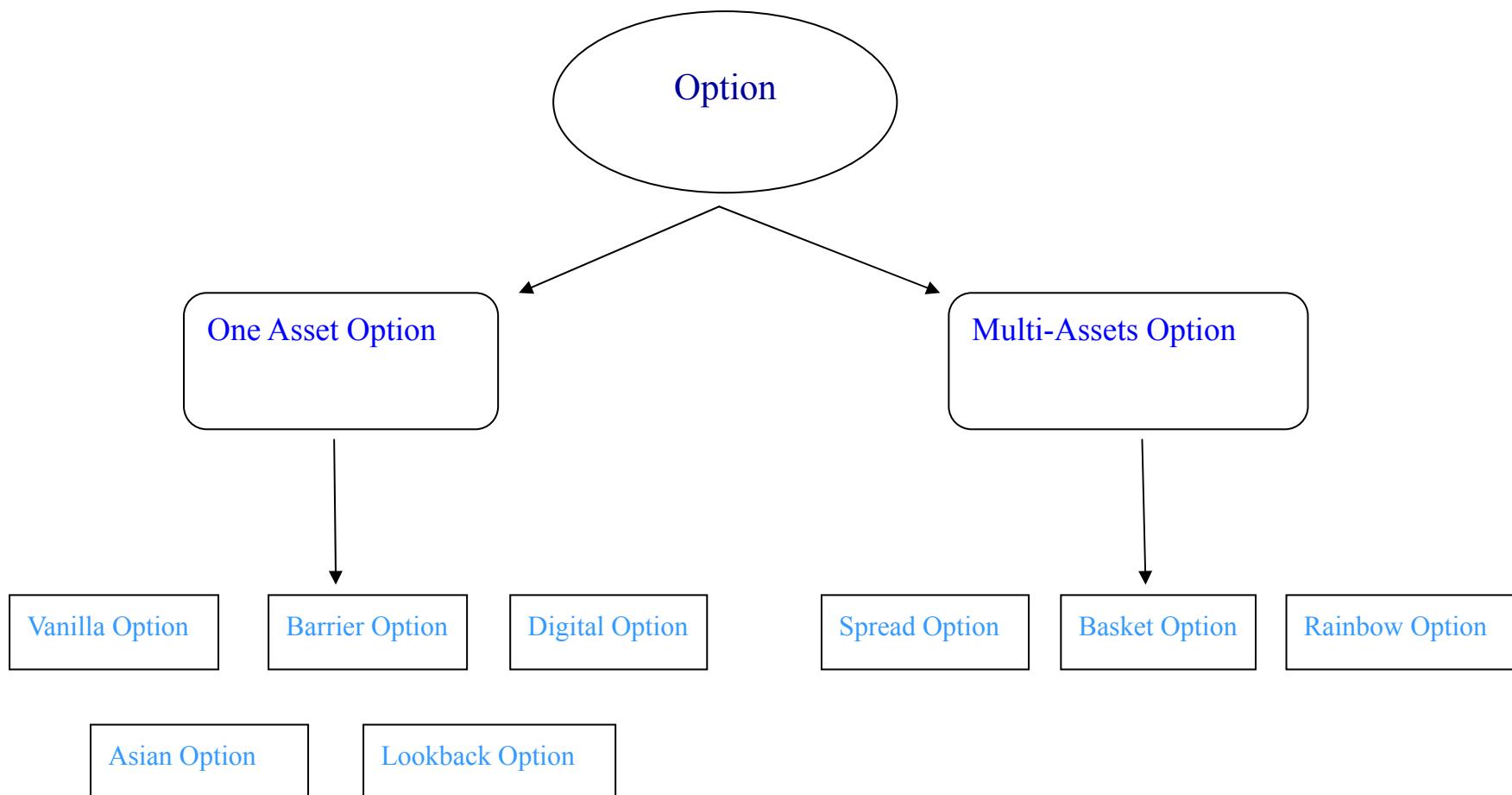
1.4 Object-oriented programming

- ◆ Object-oriented programming is a programming paradigm that uses objects to design applications and computer programs.
 - Objects are data structures consisting of data fields and methods together with their interactions .
 - Programming techniques may include features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance.
- ◆ Most modern programming languages support object-oriented programming.

◆ Example : Option Contract is an Object

- Premium is a function of Asset Price(S), Strike Prices(K), Volatility(sig), Funding Rate(r), Dividend Yield(y), Start Date & Maturity Date(TTM) 。
 - ✓ Underlying Quote Object , including S 。
 - ✓ Payoff Object , including K 。
 - ✓ Exercise Object , including TTM 。
 - ✓ Volatility Term Structure Object , including sig 。
 - ✓ Rate Term Structure Object , including r 。
 - ✓ Dividend Term Structure Object , including y 。
- Build a pricing engine object to calculate its premium & greeks.

◆ The derived classes of Option Contract



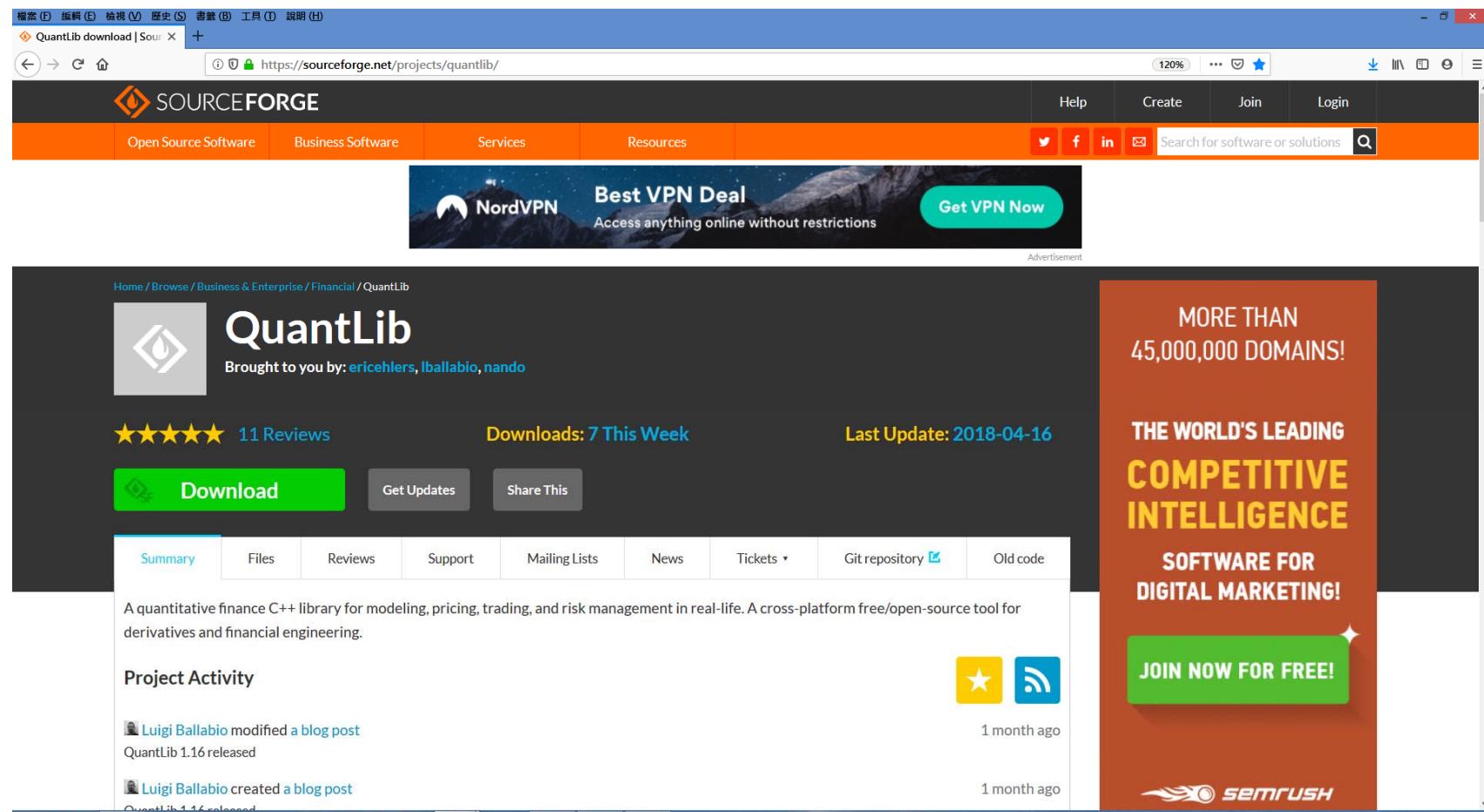
◆ QuantLib version history(Latest version is 1.16)

| Version | Release date | Notes |
|---------|---------------|---|
| 0.1.1 | Nov 21, 2000 | |
| 0.2.0 | Sep 18, 2001 | |
| 0.3.4 | Nov 21, 2003. | First time found it , in KGI 。 |
| 0.3.7 | Jul 23, 2004. | Later version need Boost library 。 |
| 0.4.0 | Feb 20, 2007. | |
| 0.8.0 | May 30, 2007. | Version change quickly to converge to 1.0 version 。 |
| 0.9.0 | Dec 24, 2007. | |
| 0.9.9 | Nov 2009. | |
| 1.0.0 | Feb 24, 2010 | |
| 1.0.1 | Sep 17, 2010 | |
| 1.1.0 | May 23, 2011 | |
| 1.2.0 | March 6, 2012 | |

1.5 Where to download?

◆ Open Source Software website , QuantLib Project ,

- <http://sourceforge.net/projects/quantlib/> , 2018/09/16 。



◆ Under Files , Directory Structure ◇

➤ Click QuantLib ,

The screenshot shows a web browser window with the URL <https://sourceforge.net/projects/quantlib/files/>. The page is titled "QuantLib - Browse Files at" and features the SourceForge logo. The main content area shows a file directory structure under the "Files" tab. The table lists the following files:

| Name | Modified | Size | Downloads / Week |
|---------------|------------|------|------------------|
| QuantLib | 2018-10-01 | | 1 |
| QuantLibAddin | 2018-02-27 | | 0 |
| QuantLibXL | 2018-02-27 | | 3 |
| ObjectHandler | 2018-02-15 | | 0 |
| test | 2017-04-12 | | 0 |
| repository | 2016-05-29 | | 3 |

Below the table are download links: "Download Latest Version" (QuantLibXL-1.8.0-bin.zip) and "Get Updates". A sidebar on the right contains an advertisement for NBA basketball tickets.

◆ In Sourceforge, we can get version 1.12 ◇

The screenshot shows a web browser window with the SourceForge interface. The URL in the address bar is [https://sourceforge.net/projects/quantlib/files/QuantLib/old releases from 1.0 to 1.12/](https://sourceforge.net/projects/quantlib/files/QuantLib/old%20releases%20from%201.0%20to%201.12/). The main content area shows the QuantLib project page with a sidebar for recommended projects.

QuantLib Project Page:

- Header:** SOURCEFORGE, Open Source Software, Business Software, Services, Resources, Help, Create, Join, Login.
- Banner:** NordVPN Best VPN Deal, Access anything online without restrictions, Get VPN Now.
- Project Summary:** QuantLib, Brought to you by: ericehlers, lballabio, nando.
- File Download Options:** Download Latest Version (QuantLibXL-1.8.0-bin.zip, 4.7 MB), Get Updates.
- File List:** Home / QuantLib / old releases from 1.0 to 1.12. The table lists the following releases:

| Name | Modified | Size | Downloads / Week |
|---------------|------------|------|------------------|
| Parent folder | | | |
| 1.12 | 2018-02-01 | | 1 |
| 1.11 | 2017-10-02 | | 0 |
| 1.10.1 | 2017-08-31 | | 0 |
| 1.10 | 2017-05-16 | | 0 |
| 1.9.2 | 2017-02-27 | | 0 |

Recommended Projects:

- QLNet: *** MOVED TO GITHUB : <https://github.com/amaggiulli/qlnet> *** QLNet is a...
- Ethereum Wallet and Mist Browser: Gateway to decentralized applications on the Ethereum blockchain

◆ We can get the latest version 1.16 in JFrog Bintray. Click QuantLib ◇

The screenshot shows a web browser window with the following details:

- Title Bar:** 檔案 (F) | 編輯 (E) | 檢視 (V) | 歷史 (S) | 書籤 (B) | 工具 (I) | 說明 (H)
- Address Bar:** releases - Generic - Bintray X + https://bintray.com/quantlib/releases 120% ⌂ ⌃ ⌄ ⌅ ⌆ ⌈ ⌉ ⌋ ⌊ ⌊ ⌋ ⌈ ⌉ ⌁ ⌂ ⌃ ⌄ ⌅ ⌆ ⌈ ⌉ ⌋ ⌊ ⌊ ⌋ ⌈ ⌉ ⌁
- Header:** JFrog Bintray | Search Bintray | Go to New Look | API | User Guide | Pricing | Sign In
- Section:** quantlib / releases
- Owned by:** QuantLib | Report
- Official QuantLib releases:** SET ME UP!
- Search and Filter:** Package Name | Sorted By Name
- Releases:**
 - QuantLib:** ★★★★★ The QuantLib C++ library | Version 1.16 (Aug 05, 2019)
 - QuantLib-SWIG:** ★★★★★ Wrappers for QuantLib in a number of languages | Version 1.16 (Aug 05, 2019)
- Feedback:** Feedback

◆ See all versions ◦

The screenshot shows a web browser window with the following details:

- Title Bar:** 檔案 (F) | 編輯 (E) | 檢視 (V) | 歷史 (S) | 書籤 (B) | 工具 (I) | 說明 (H)
- Address Bar:** Package QuantLib - quant X + https://bintray.com/quantlib/releases/QuantLib
- Header:** JFrog Bintray | Search Bintray | Go to New Look | API | User Guide | Pricing | Sign In
- Left Sidebar:** Q quantlib / releases / QuantLib
- Content Area:**
 - Owned by:** QuantLib
 - Rating:** ★★★★★
 - Report:** Report
 - Image:** A small icon of a cube.
 - Description:** The QuantLib C++ library
 - Buttons:** SET ME UP! (blue button), Feedback (dark grey button)
- Navigation Tabs:** General (selected), Readme, Release Notes, Reviews (0), Statistics, Files
- About This Package:**

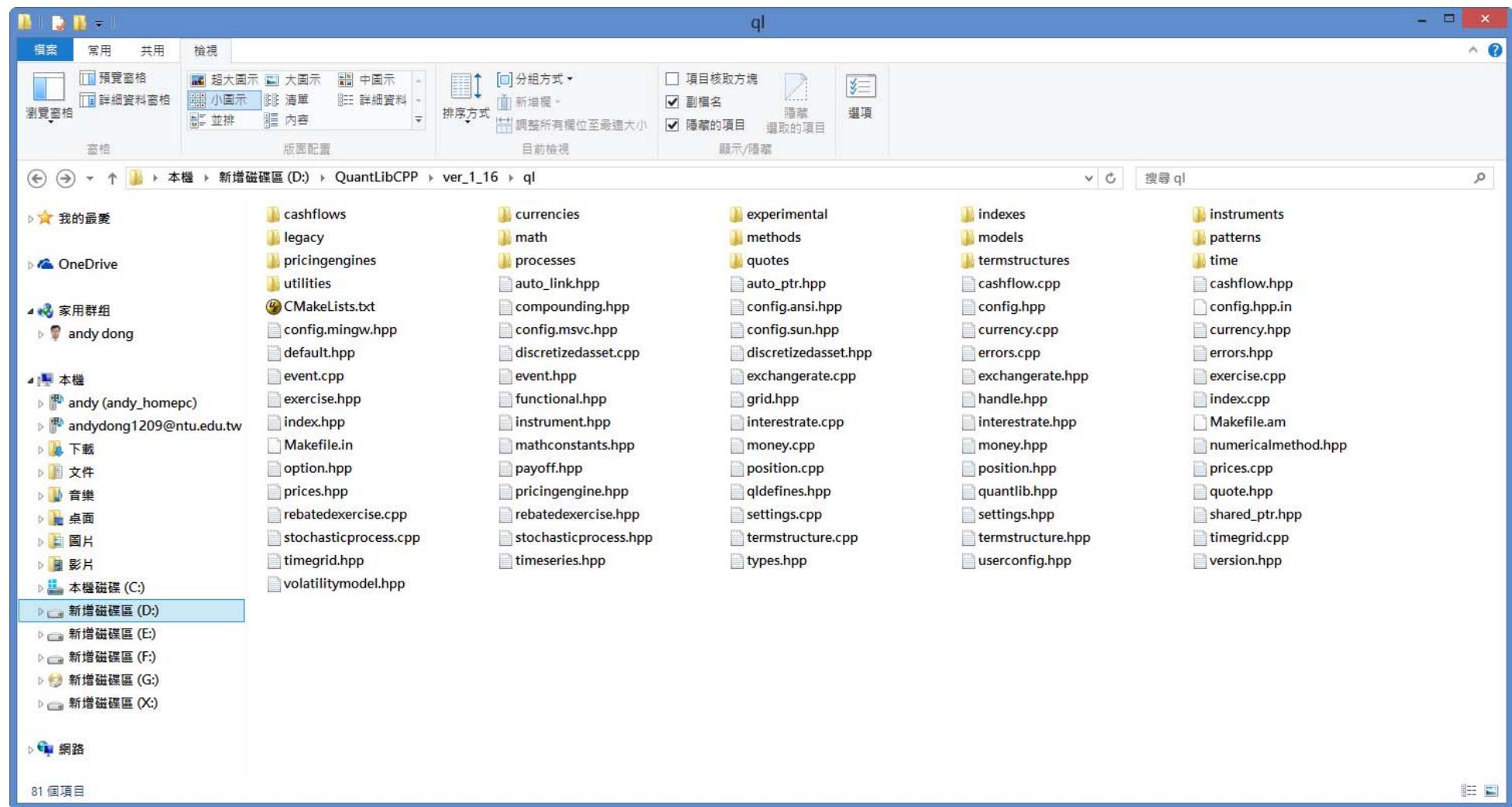
| | |
|---------------|---|
| Website | http://www.quantlib.org |
| Issue Tracker | https://github.com/lballabio/QuantLib/issues |
| VCS | https://github.com/lballabio/QuantLib |
| Product | None |
| Maturity | Official |
- Version Notification Links:** Version Notification Links (eye icon)
- Versions:**
 - 1.16
 - 1.15
 - 1.14
 - 1.13
 - 1.12.1
- Latest Version Badge:** A blue badge with the text "1.16" and a Twitter icon.

◆ Click left-bottom corner, QuantLib-1.16.zip, to download.

The screenshot shows a web browser window with the following details:

- Header:** 檔案 (F) | 編輯 (E) | 檢視 (V) | 歷史 (S) | 書籤 (B) | 工具 (I) | 說明 (H)
- Title Bar:** Version QuantLib/1.16 - q
- Address Bar:** https://bintray.com/quantlib/releases/QuantLib/1.16
- Toolbar:** Back, Forward, Stop, Refresh, Home, Zoom (120%), More, Star, Download, Print, E-mail, etc.
- Content Area:**
 - General Tab:** Selected. Other tabs include Readme, Release Notes, Reviews (0), Statistics, and Files.
 - About This Version:** Includes links to Website (<http://www.quantlib.org>), Issue Tracker (<https://github.com/lballabio/QuantLib/issues>), VCS (<https://github.com/lballabio/QuantLib>), Licenses (BSD 3-Clause), and VCS Tag (None).
 - Version Publication Date:** Released in Aug 05, 2019.
 - Version Badge:** A green badge with a checkmark icon.
 - Watchers:** 0. Watch button with 66 notifications.
 - No watchers yet.**
 - Downloads:** QuantLib-1.16.tar.gz (Size: 8.53 MB, sha256: 204ad5822259f9a9146eaf660f0b756100604e3adb85c501d41d201bf09dec94) and QuantLib-1.16.zip (Size: 11 MB, sha256: 0dd7fecac39f36496809716cac9a30e0b9776a24f79877cb68b49d80dc67af5).
- Sidebar:** Feedback button.

◆ Download unzip ◦



◆ QuantLib's Wiki , <http://en.wikipedia.org/wiki/QuantLib> ◈



The screenshot shows a Microsoft Edge browser window displaying the English Wikipedia page for QuantLib. The title bar reads "QuantLib - Wikipedia". The main content area features the QuantLib logo and the heading "QuantLib". Below the heading, it says "From Wikipedia, the free encyclopedia". A warning box is present, stating: "This article has multiple issues. Please help improve it or discuss these issues on the talk page. (Learn how and when to remove these template messages)" with a large orange exclamation mark icon. A bulleted list follows: "• The topic of this article may not meet Wikipedia's notability guidelines for products and services. (January 2014)" "• This article includes a list of references, related reading or external links, but its sources remain unclear because it lacks inline citations. (March 2010)" "• This article relies too much on references to primary sources. (January 2017)" To the right, there is a summary box titled "QuantLib" containing the following information:

| |
|--|
| QuantLib logo (Fontin Bold font) |
| Developer(s) QuantLib Team |
| Stable release 1.8 / May 18, 2016; 3 years ago |
| Repository github.com/lballabio/quantlib |
| Written in C++ |
| Type Numerical library |

The left sidebar contains the standard Wikipedia navigation menu.

◆ Github , 2019/08/25 ◇

The screenshot shows Luigi Ballabio's GitHub profile page. At the top, there is a navigation bar with links for Pull requests, Issues, Marketplace, and Explore. Below the navigation bar, there is a pinned repository section featuring "QuantLib" and "QuantLib-SWIG". The "QuantLib" repository is described as "The QuantLib C++ library" and has 1.6k stars. The "QuantLib-SWIG" repository is described as "The QuantLib extension modules" and has 111 stars. Below this, there is a heatmap visualization titled "490 contributions in the last year" showing contribution activity from September to August. The heatmap uses a color scale where lighter shades represent "Less" contributions and darker shades represent "More" contributions. A legend at the bottom right of the heatmap indicates this scale. On the left side of the page, there is a sidebar with a profile picture of Luigi Ballabio, his name "Luigi Ballabio", his GitHub handle "lballabio", and a "Follow" button. Below the sidebar, there is a bio text: "One of the administrators and lead developers of the Quantlib project (<http://quantlib.org>). Also husband, father of four, ex-physicist, and amateur musician." There are also links to his location "Milan, Italy" and his website "<http://implementingquantlib.com>".

1.6 QuantLib Contents

- ◆ QuantLib consists more than 2000 files , include 15 modules 。
 - 1.Numeric types : define various data types , for example , interest rate、spread、volatility are real 。
 - 2.Currencies and FX rates : include 66 currencies , and FX rate conversion manager 。
 - 3.Date and time calculations : include 37country/region calendars 、 7 day counter types 。
 - 4.Pricing engines : 9 types pricing engines , include Asian option、Barrier option、Basket option 、 Cap/Floor 、 Cliquet option 、 Forward option 、 Quanto option 、 Swaption 、 Vanilla option 。 Each type consists analytic solution 、 binomial tree 、 finite difference and Monte-Carlo simulation implementation 。
 - 5.Finite-differences framework : 3 types implementation 。
 - 6.Short-rate modeling framework : include single factor models , Vasicek 、 CIR 、 Hull-White 、 Black-Karasinski 、 Extended CIR , and two factor model , G2 Model 。
 - 7.Financial instruments : 40+ financial instruments , include Swap、Vanilla Option、Exotic Option 、 Stock 、 Forward 、 Cap 、 Floor 、 Color 、 Bond 、 Future 、 Callable Bond and corresponding Quanto 、 Inflation Bond 。

- 8.Lattice methods : 1-dim , 2-dim binomial trees and trinomial trees 。
- 9.Math tools: including Distribution、Integration、Correlation、Interpolation、Matrix、Optimization 、Random Number 、Solver 、Statistics 。
- 10.Monte Carlo framework : Single and Multiple factor , European & American simulation objects 。
- 11.Design patterns : Singleton 、Observer/Observable 、Lazy Object 、Composite 、Curiously Recurring Template 、Acyclic Visitor 。
- 12.Stochastic processes : Geometric Brownian Motion 、Stochastic Volatility Model 、Square Root Process 、Ornstein-Uhlenbeck Process 、Hull White Process 、G2 Process 。
- 13.Term structures : Interest rate 、volatility 、inflation and credit term structure object 。
- 14.Models : including Equity Model 、Short Rate Model 、Volatility Model , Libor/Swap Market Model 。
- 15.QuantLib macros : Numerical limits and debugging Macros 。

1.7 Our Choice

- ◆ In this class, we use QuantLib C# version.

- Easy to use.
 - ✓ C++ is difficult for most people, C# is easier.
- Easy to build
 - ✓ Visual Studio is a user friendly IDE to write program.
- Easy to work
 - ✓ Window is a familiar environment to work.
- For ambitious or advanced readers, C++ is the native language of Financial Engineers.

- ◆ Where to get the source code?

- My Github web, <https://github.com/andydong1209/NTUSTFE>。

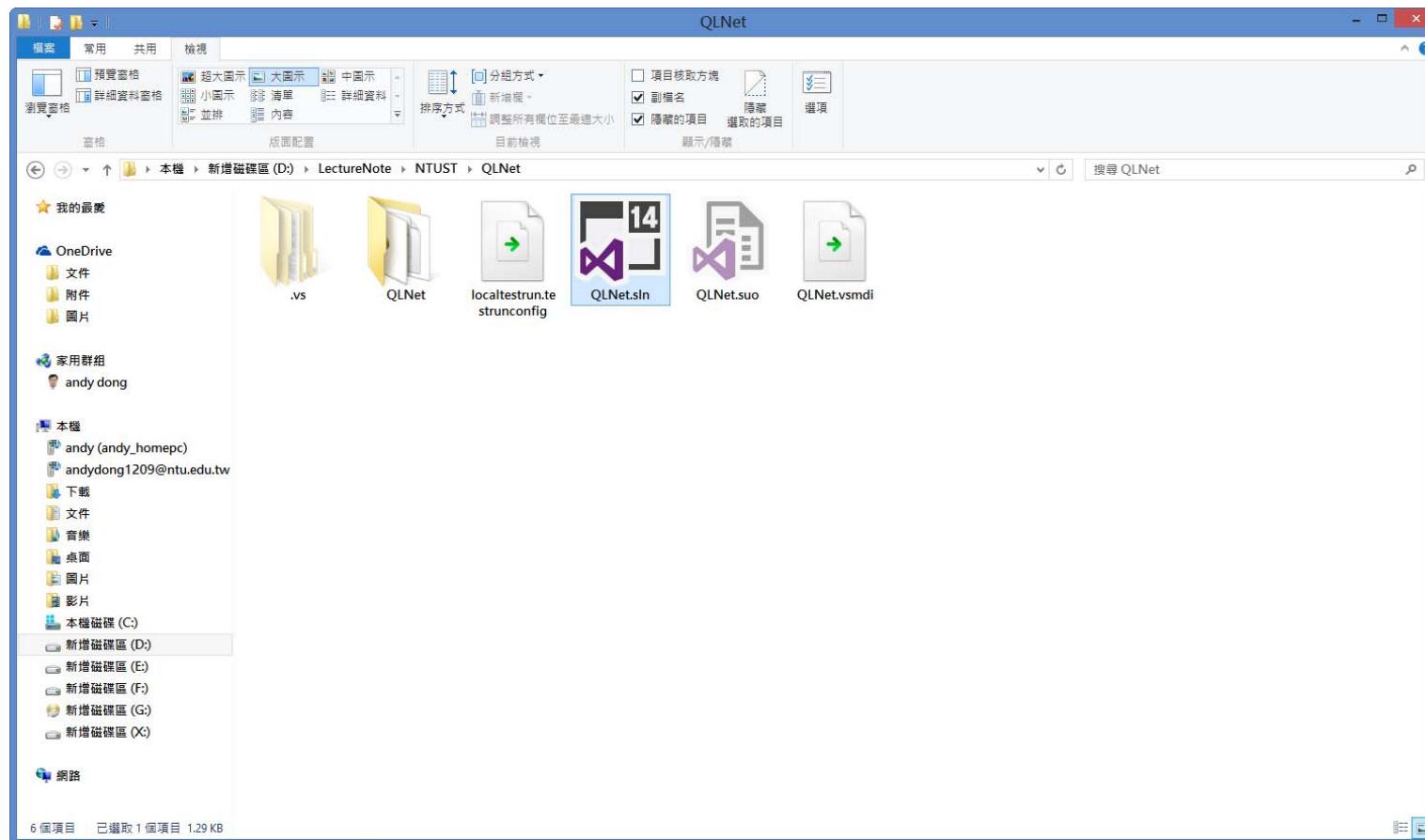
- ◆ We use Visual Studio 2015(2017) Community Edition as our developing environment.

- It can download from Web.

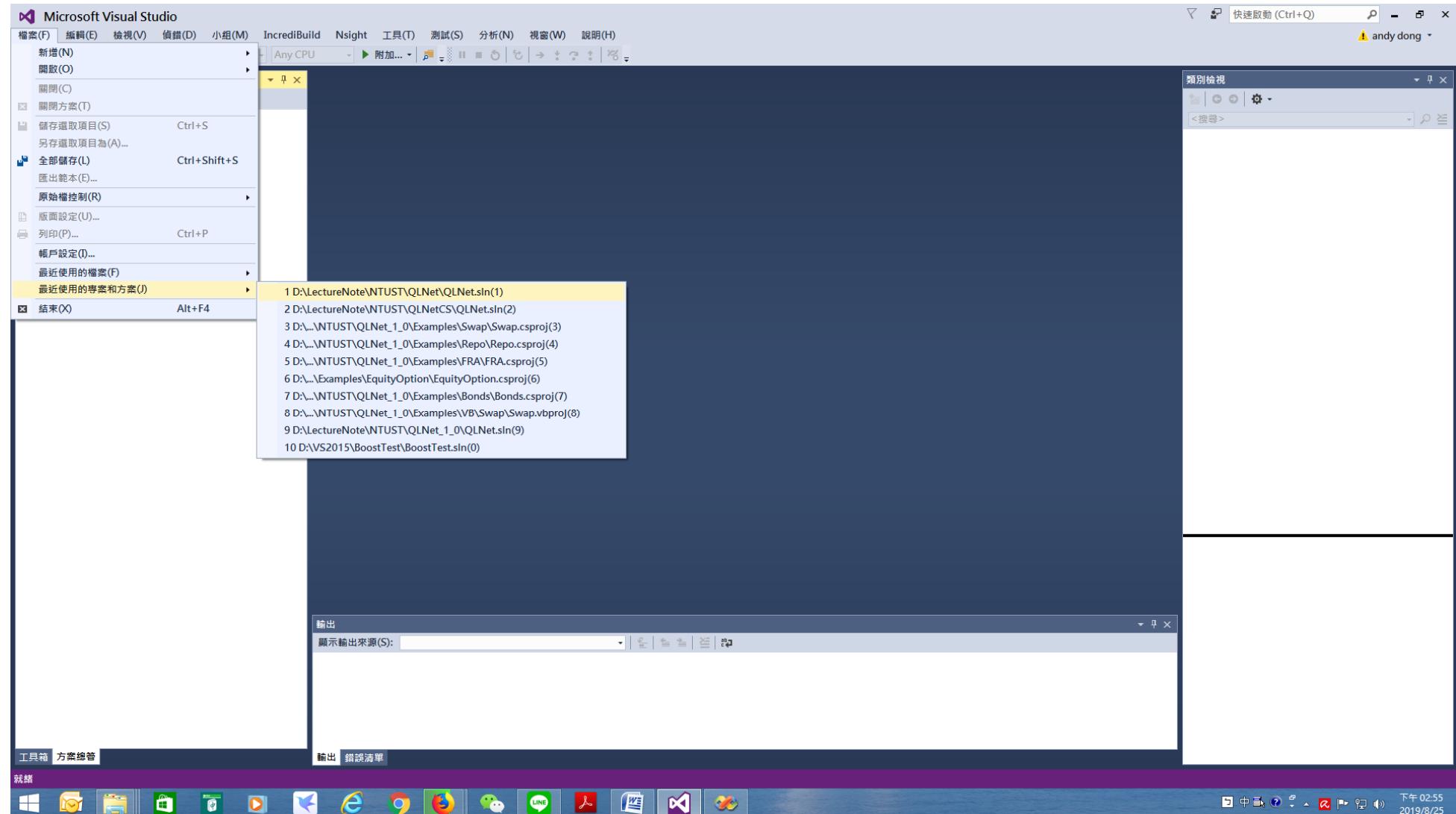
1.8 First Step: DateTime Example

◆ Open QLNet Solution。

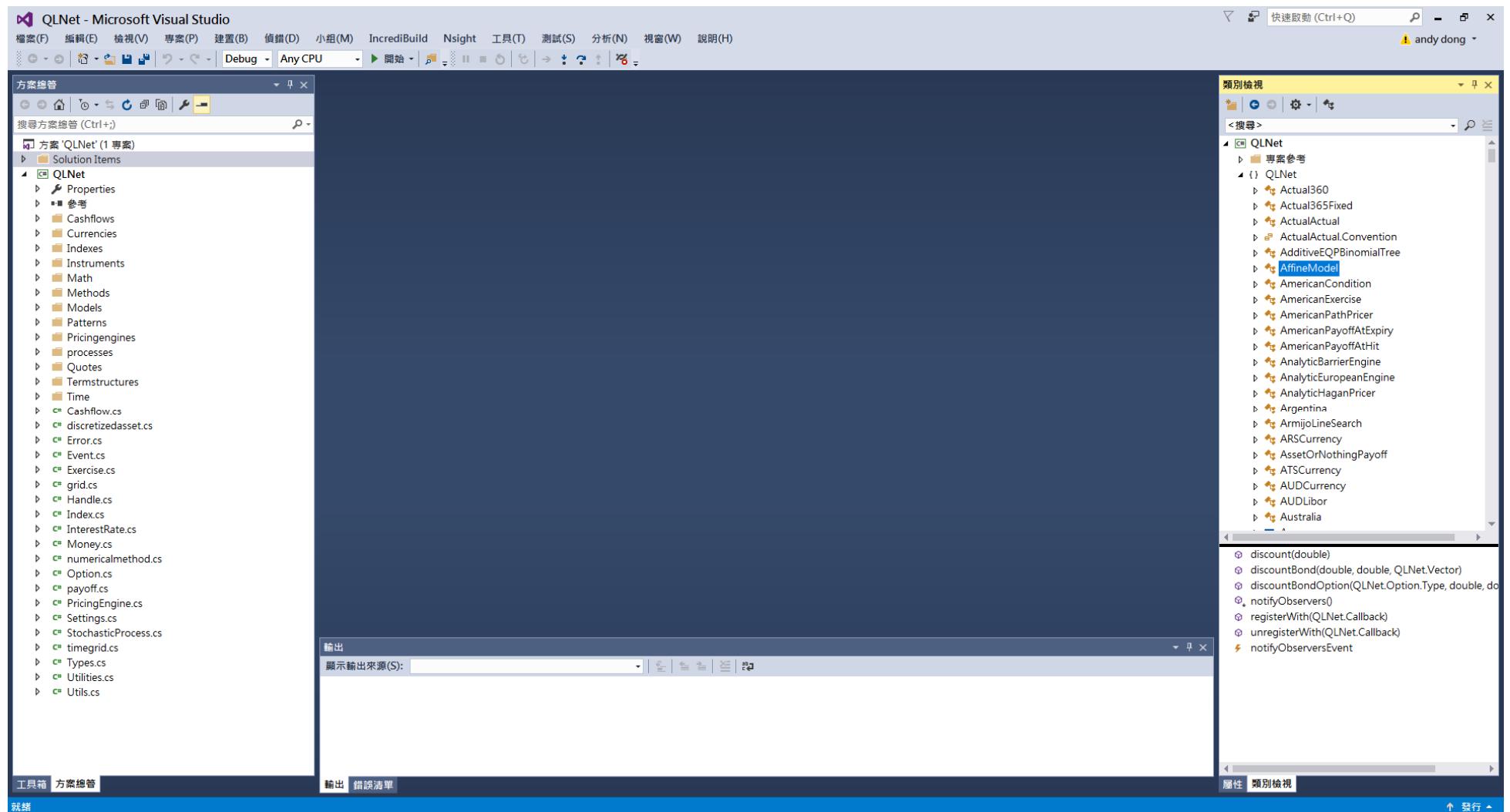
- In My Computer , D:\LectureNote\NTUST\QLNet ,



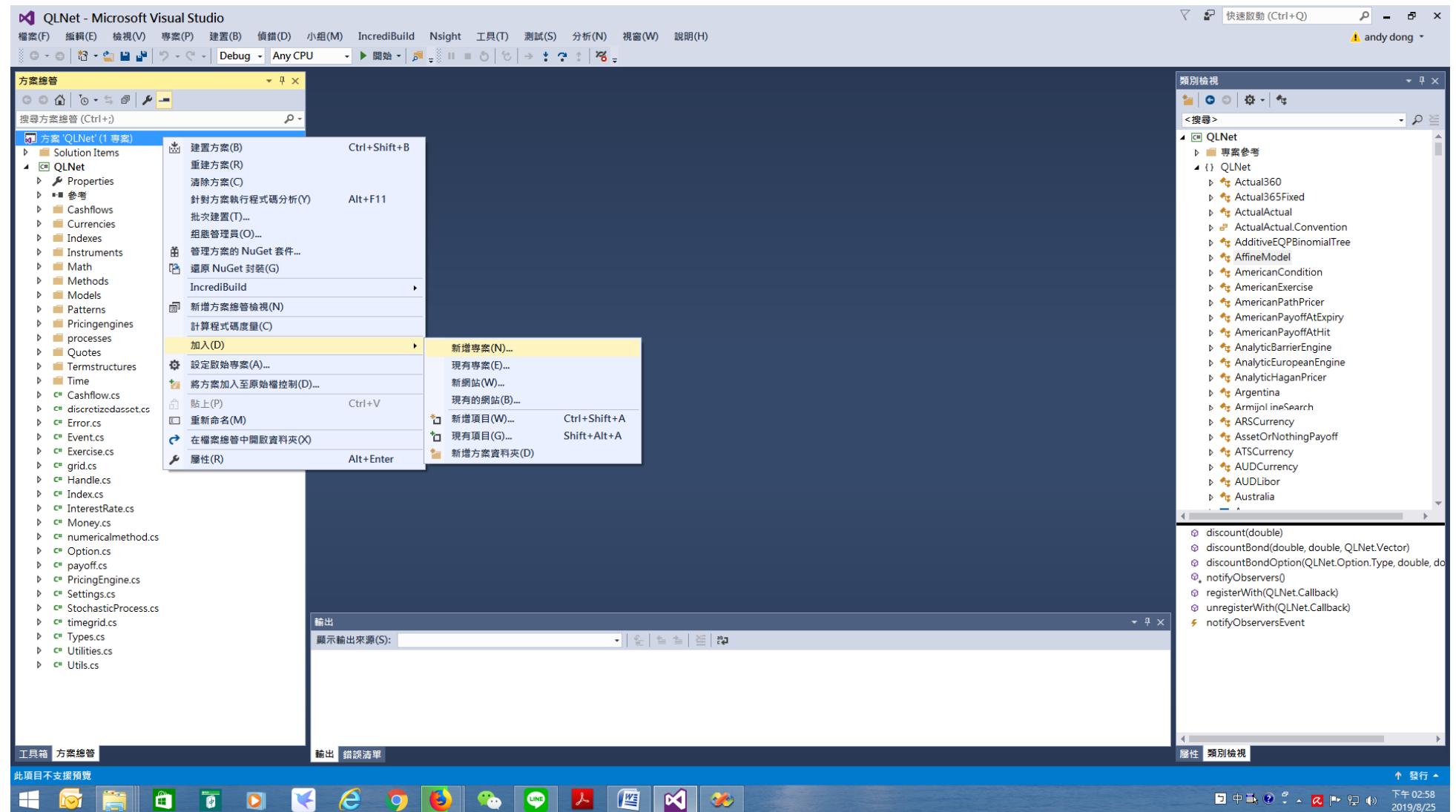
◆ Open with VS2015。



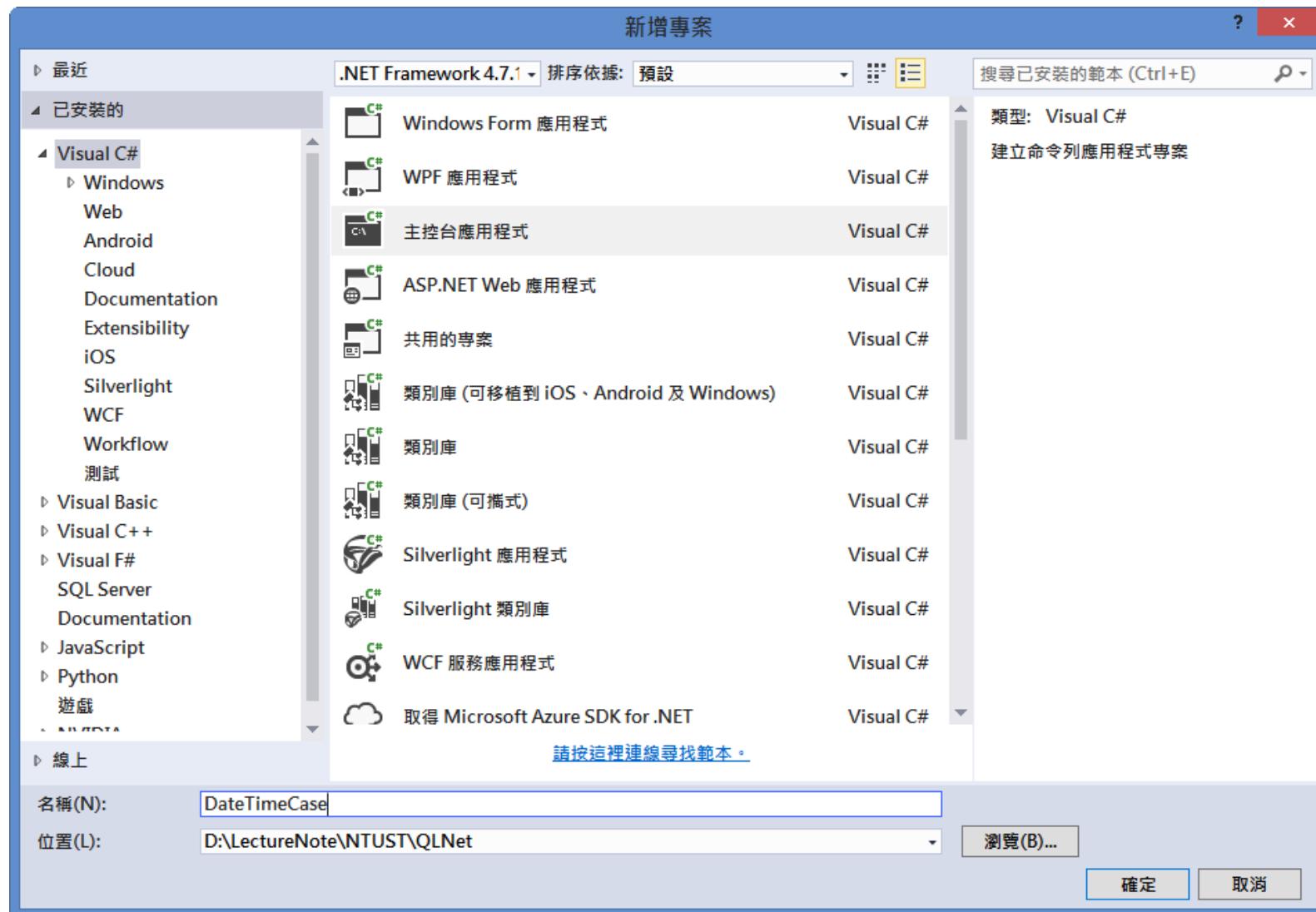
➤ Original Solution, with only one project, A DLL Library project.



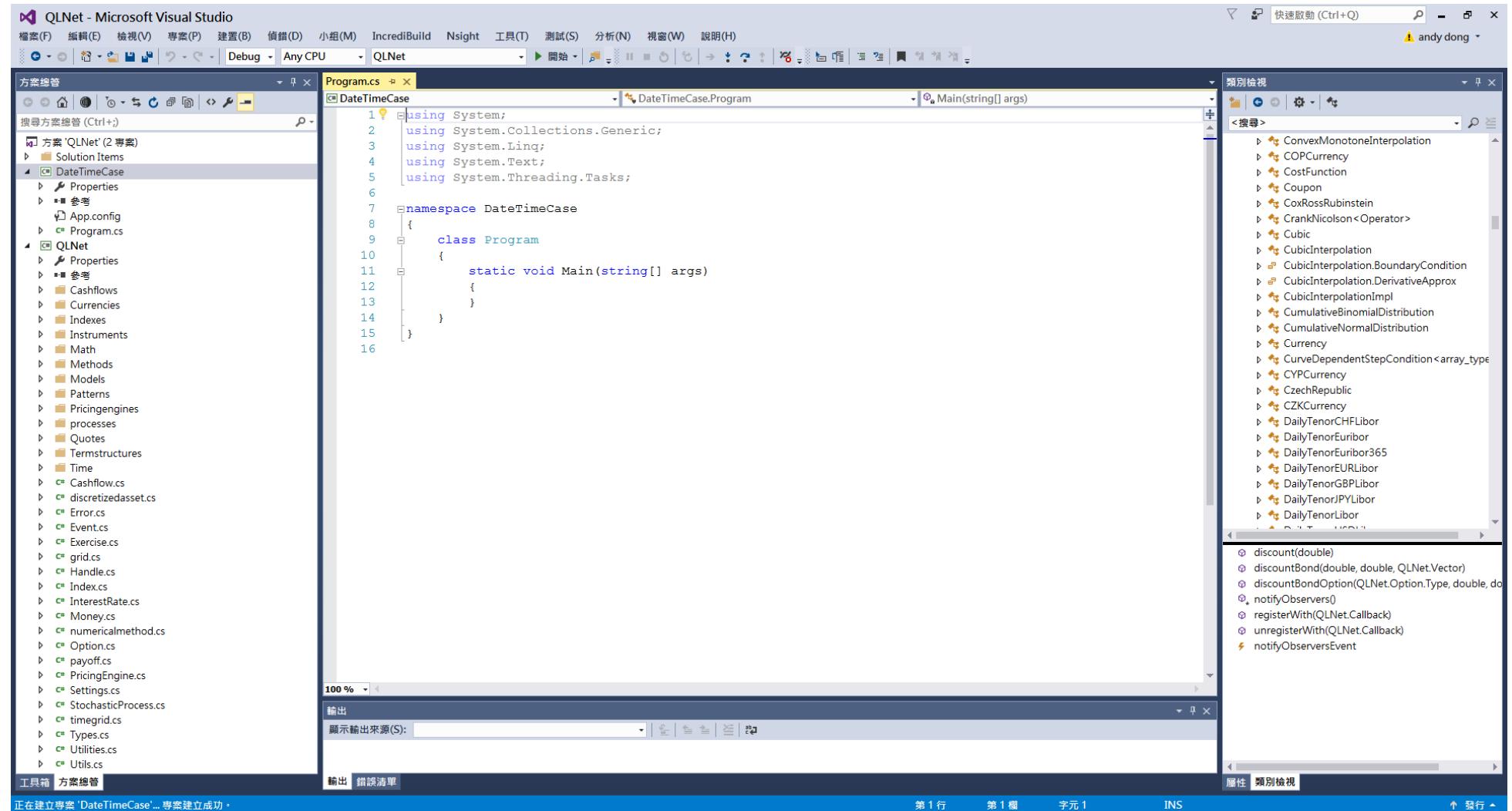
◆ Add New -- Project



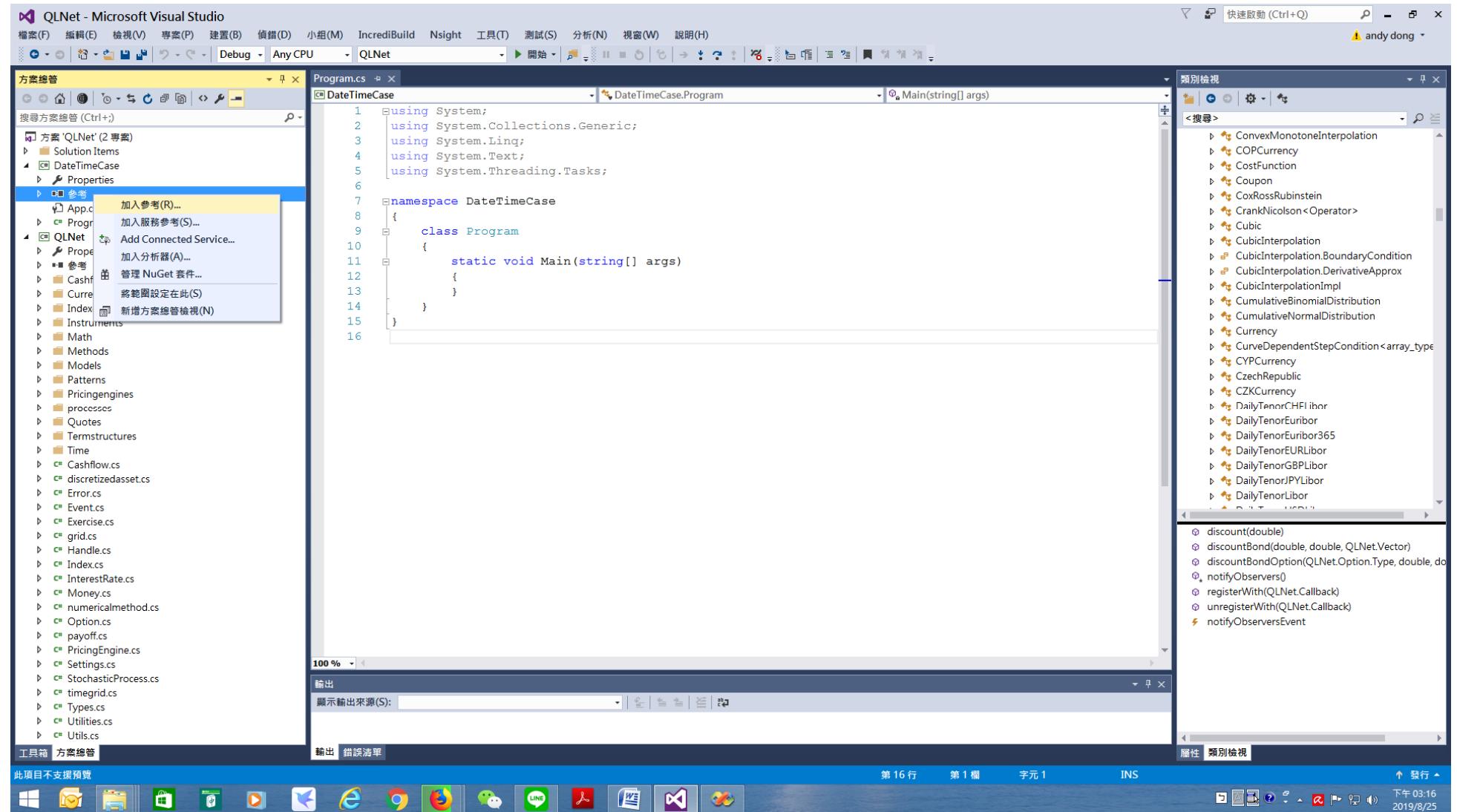
◆ Name : DateTimeCase , Console Application。



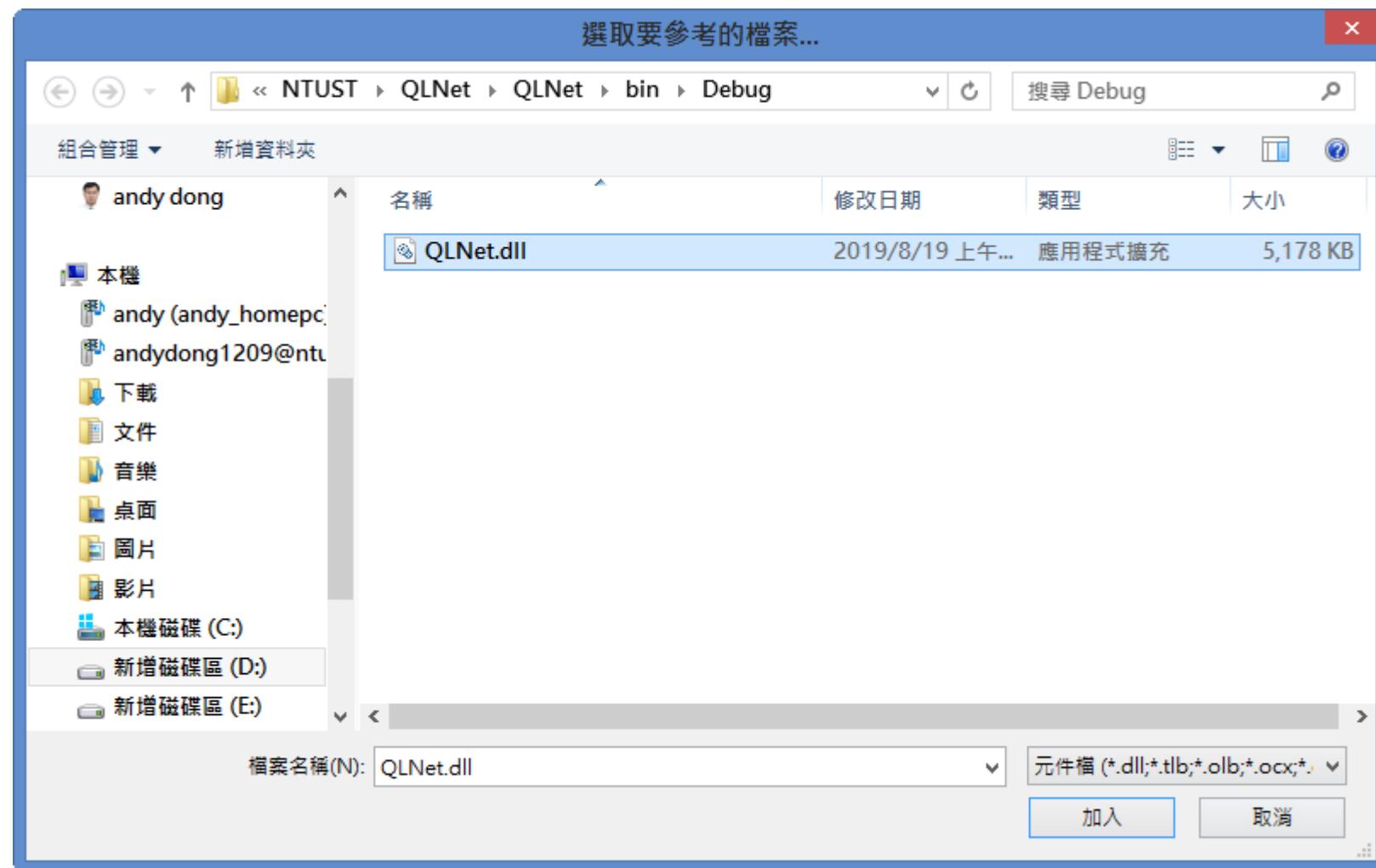
◆ The DateTimeCase project environment ◦



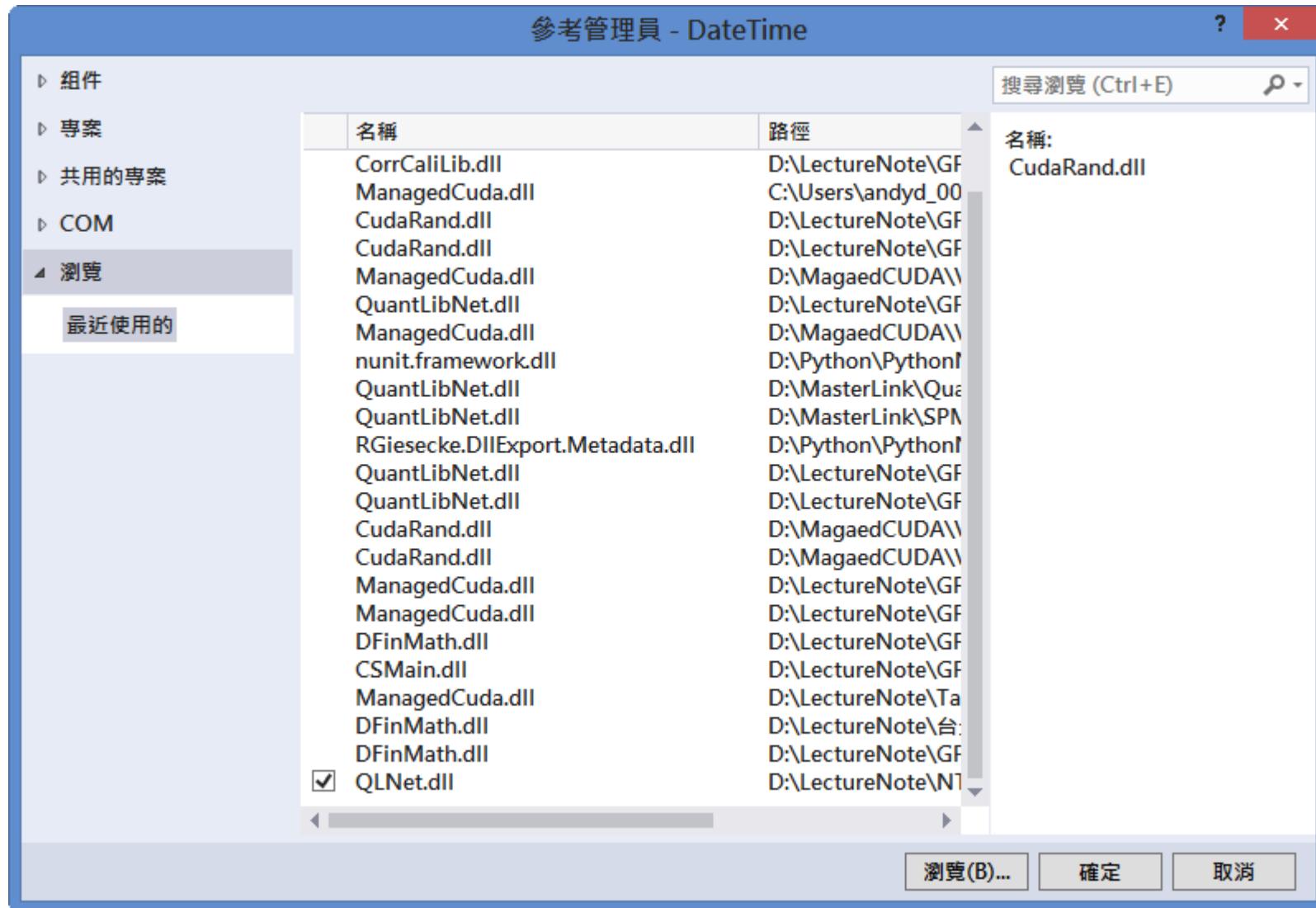
◆ Add Reference



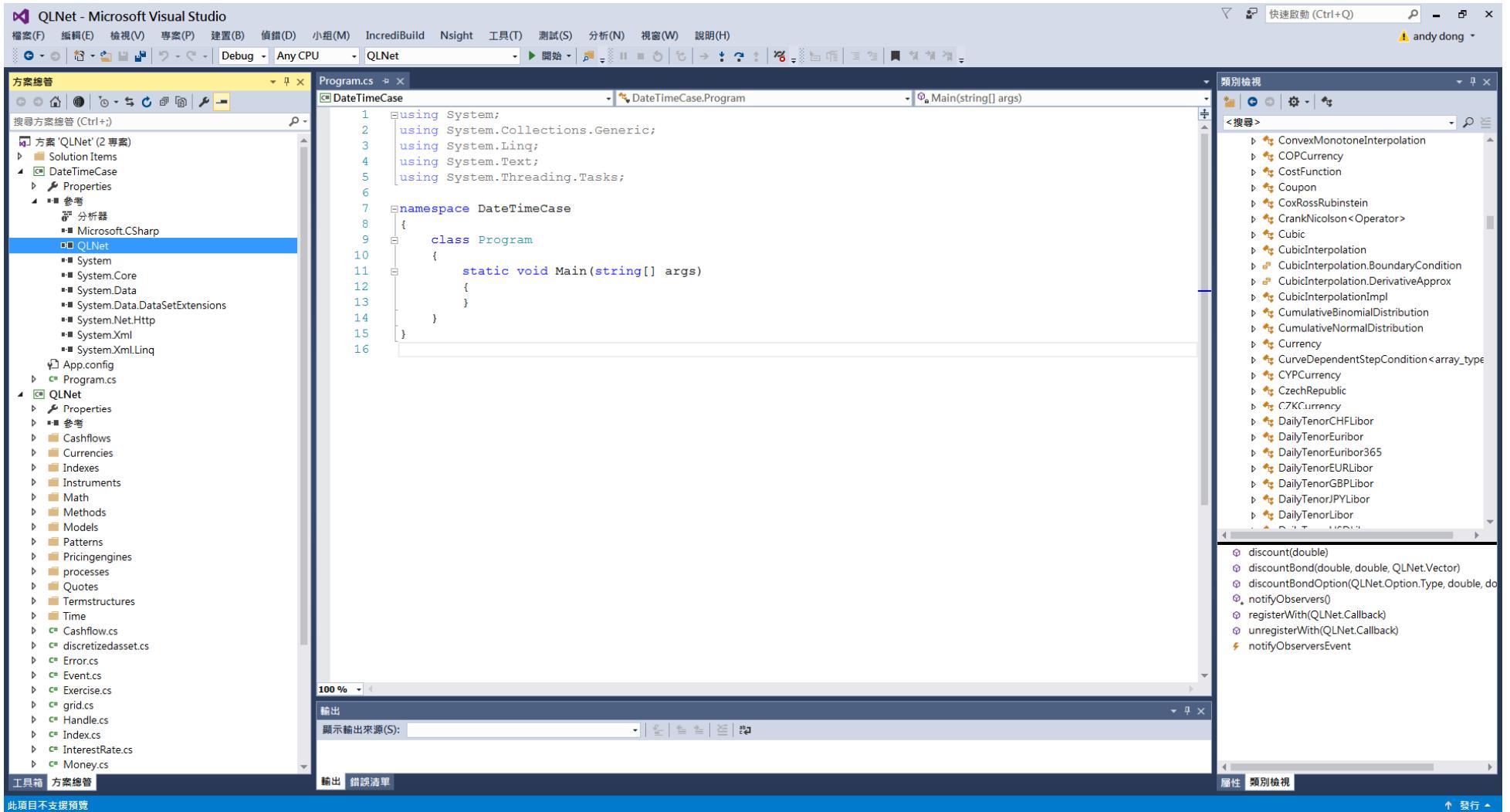
◆ Add QLNet.dll。



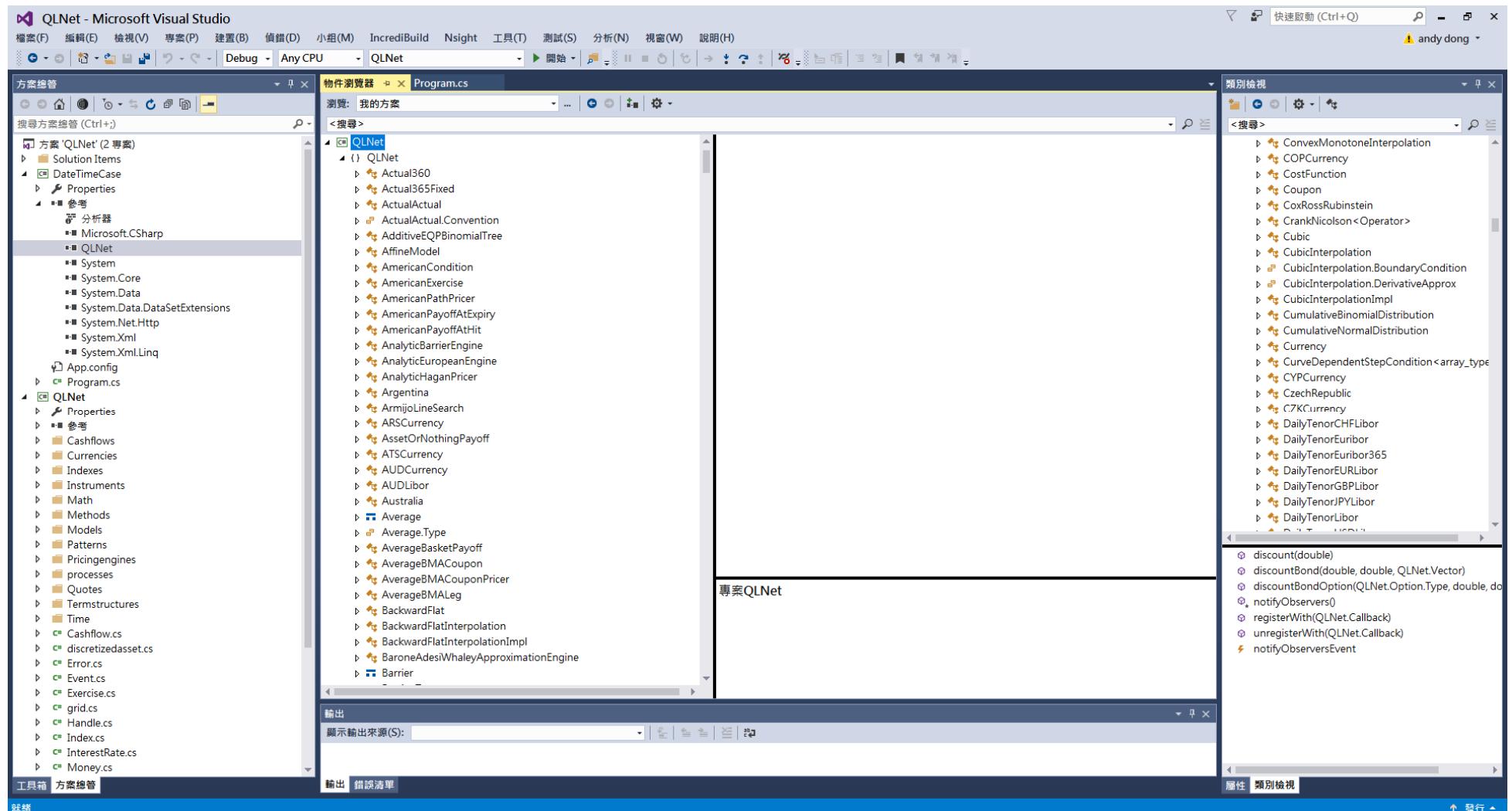
➤ Confirm 。



➤ See the DLL.



- Double click QLNet, in Object Browser find QLNet and its objects.



◆ Using Name Space ,

using QLNet;

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the 'QLNet' solution containing two projects: 'QLNet' and 'DateTimeCase'. The 'QLNet' project has numerous files under its 'Properties' and 'QLNet' folder, including 'App.config', 'Program.cs', and various C# files for cashflows, currencies, indexes, instruments, math, methods, models, patterns, pricingengines, processes, quotes, termstructures, time, and error handling.
- Object Browser:** Located on the right side of the IDE, it displays a tree view of the QLNet namespace. It lists many classes such as ConvexMonotoneInterpolation, COPCurrency, CostFunction, Coupon, CoxRossRubinstein, CrankNicolson<Operator>, Cubic, CubicInterpolation, CubicInterpolationBoundaryCondition, CubicInterpolationDerivativeApprox, CubicInterpolationImpl, CumulativeBinomialDistribution, CumulativeNormalDistribution, Currency, CYPCurrency, CzechRepublic, CZKCurrency, DailyTenorCHLibor, DailyTenorEURibor, DailyTenorEURibor365, DailyTenorEURLibor, DailyTenorGBPLibor, DailyTenorJPYLibor, DailyTenorLibor, and several discounting and observer-related methods like discount, discountBond, discountBondOption, notifyObservers, registerWith, unregisterWith, and notifyObserversEvent.
- Code Editor:** The main window shows the 'Program.cs' file with the following code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using QLNet;
7
8  namespace DateTimeCase
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14         }
15     }
16 }
17
```
- Status Bar:** At the bottom, the status bar shows '第 6 行 第 13 横 字元 13 INS'.

◆ Add Code , Build Solution ◆

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Solution Explorer:** Shows the 'QLNet' solution with its projects and files. The 'QLNet' project is selected.
- Object Explorer:** Shows the class hierarchy for the selected project.
- Toolbox:** Standard .NET components are visible.
- Task List:** Lists tasks such as 'Build Solution' and 'Run'.
- Properties:** Shows the properties for the selected item.
- Code Editor:** Displays the 'Program.cs' file with the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6  using QLNet;
7
8  namespace DateTimeCase
9  {
10     class Program
11     {
12         static void Main(string[] args)
13         {
14             DateTime timer = DateTime.Now;
15             Calendar calendar = new Taiwan();
16             Date todaysDate = new Date(8, Month.April, 1998);
17             Date settlementDate = new Date(10, Month.April, 1998);
18             Settings.setEvaluationDate(todaysDate);
19
20             Date bdate = new Date(31, Month.Jan, 2010);
21             Date edate = new Date(31, Month.Jul, 2010);
22             DayCounter dc = new Actual365Fixed();
23
24             double num = dc.dayCount(bdate, edate);
25             double yr = dc.yearFraction(bdate, edate);
26
27             Period pd1 = new Period(1, TimeUnit.Months);
28             Frequency f1 = Frequency.Weekly;
29             Period pd2 = new Period(f1);
30
31             Date MonthLater = bdate + pd1;
32             Date WeekLater = bdate + pd2;
33
34             Console.WriteLine("today: " + todaysDate.ToString());
35             Console.WriteLine("settlement: " +
36                             settlementDate.ToString());
37         }
38     }
39 }
```

- Output Window:** Shows the build output.
- Category Explorer:** A floating window showing a list of QLNet classes and methods, such as ConvexMonotoneInterpolation, COPCurrency, CostFunction, Coupon, CoxRossRubinstein, CrankNicolson<Operator>, Cubic, CubicInterpolation, CubicInterpolation.BoundaryCondition, CubicInterpolation.DerivativeApprox, CubicInterpolationImpl, CumulativeBinomialDistribution, CumulativeNormalDistribution, Currency, CYPCurrency, CzechRepublic, CZKCurrency, DailyTenorCHLibor, DailyTenorEURibor, DailyTenorEURibor365, DailyTenorEURLibor, DailyTenorGBLibor, DailyTenorJPYLibor, DailyTenorLibor, discount, discountBond, discountBondOption, notifyObservers, registerWith, unregisterWith, and notifyObserversEvent.

➤ Code

```
using System;
using QLNet;

namespace DateTimeCase
{
    class Program
    {
        static void Main(string[] args)
        {
            DateTime timer = DateTime.Now;
            Calendar calendar = new Taiwan();
            Date todaysDate = new Date(8, Month.April, 1998);
            Date settlementDate = new Date(10, Month.April, 1998);
            Settings.setEvaluationDate(todaysDate);

            Date bdate = new Date(31, Month.Jan, 2010);
            Date edate = new Date(31, Month.Jul, 2010);
            DayCounter dc = new Actual365Fixed();
```

```
double num = dc.dayCount(bdate, edate);
double yr = dc.yearFraction(bdate, edate);

Period pd1 = new Period(1, TimeUnit.Months);
Frequency f1 = Frequency.Weekly;
Period pd2 = new Period(f1);

Date MonthLater = bdate + pd1;
Date WeekLater = bdate + pd2;

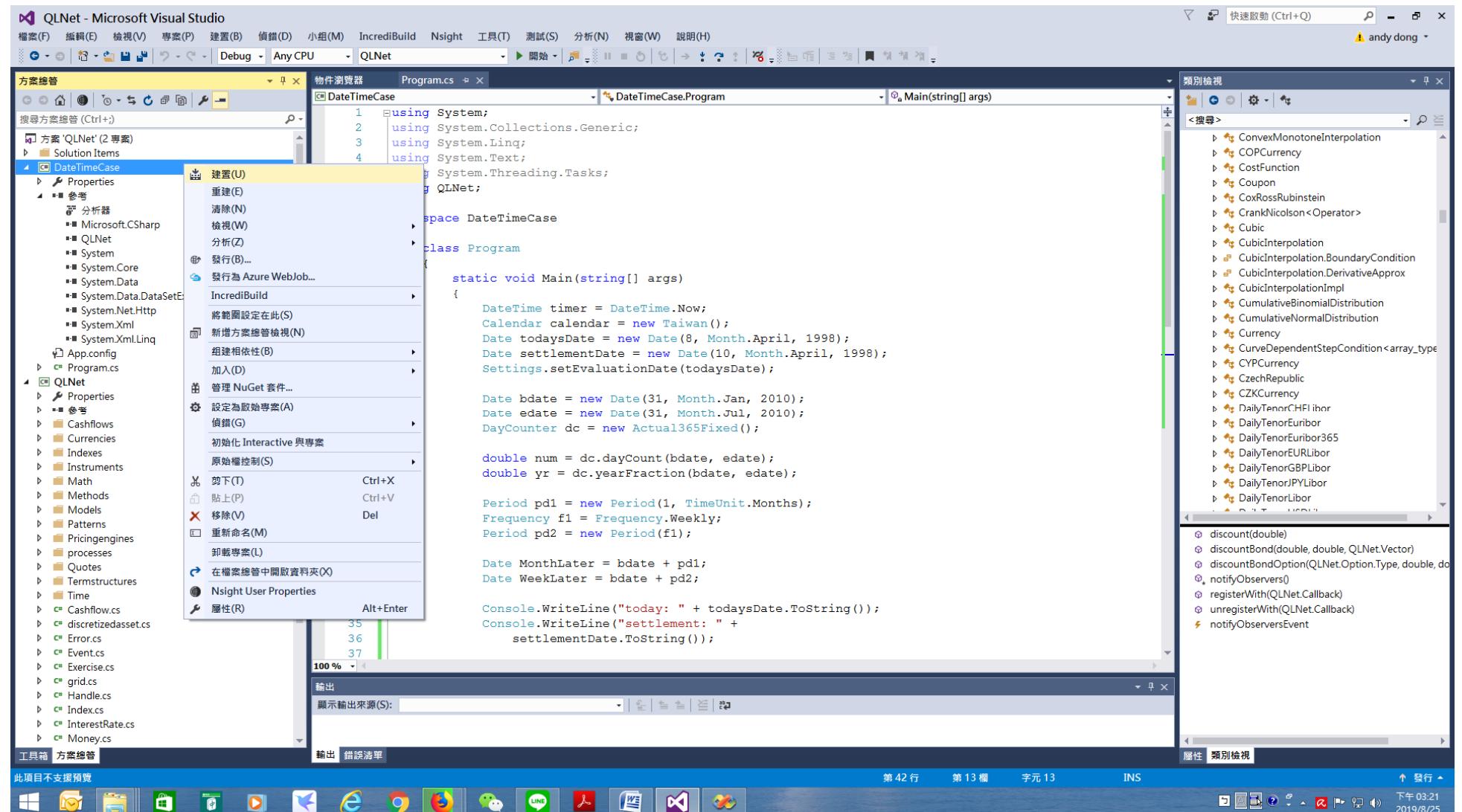
Console.WriteLine("today: " + todaysDate.ToString());
Console.WriteLine("settlement: " + settlementDate.ToString());

Console.WriteLine("\nPeriod begin date: " + bdate.ToString() + " end date: " + edate.ToString());
Console.WriteLine("Number of day: " + num.ToString());
Console.WriteLine("Year Fraction: " + yr.ToString());

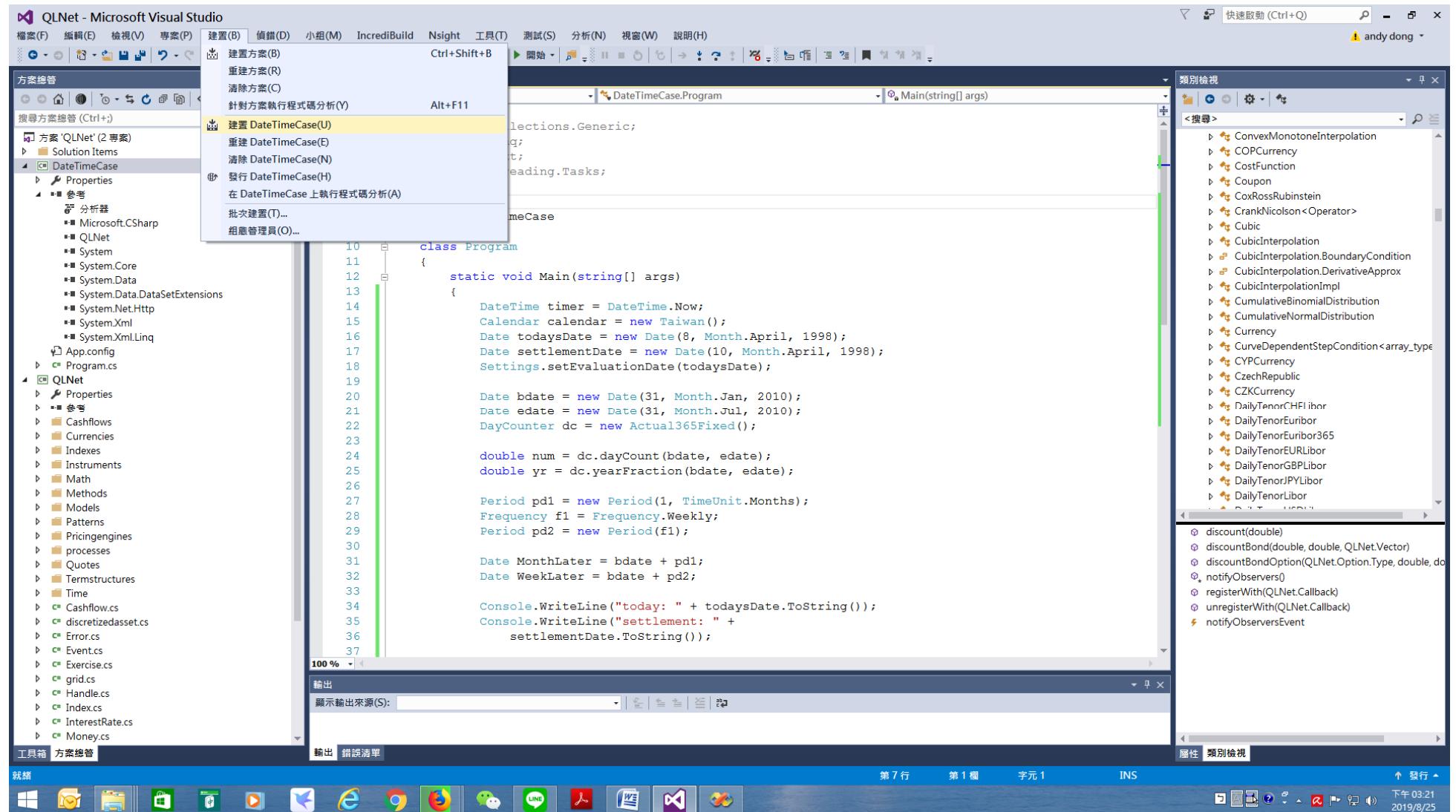
Date amonth = Date.advance(bdate, 1, TimeUnit.Months);
Console.WriteLine("\nA Month after begin date: " + amonth.ToString());
Console.WriteLine("A Month after begin date: " + MonthLater.ToString());
```

```
Console.WriteLine("A Week after begin date: " + WeekLater.ToString());  
  
Date setdate = Date.nthWeekday(3, DayOfWeek.Friday, 4, 2010);  
Console.WriteLine("\n3rd Friday of April 2010: " + setdate.ToString());  
  
Console.ReadKey();  
}  
}  
}
```

◆ Build from popup menu ◦



➤ Build from main menu。

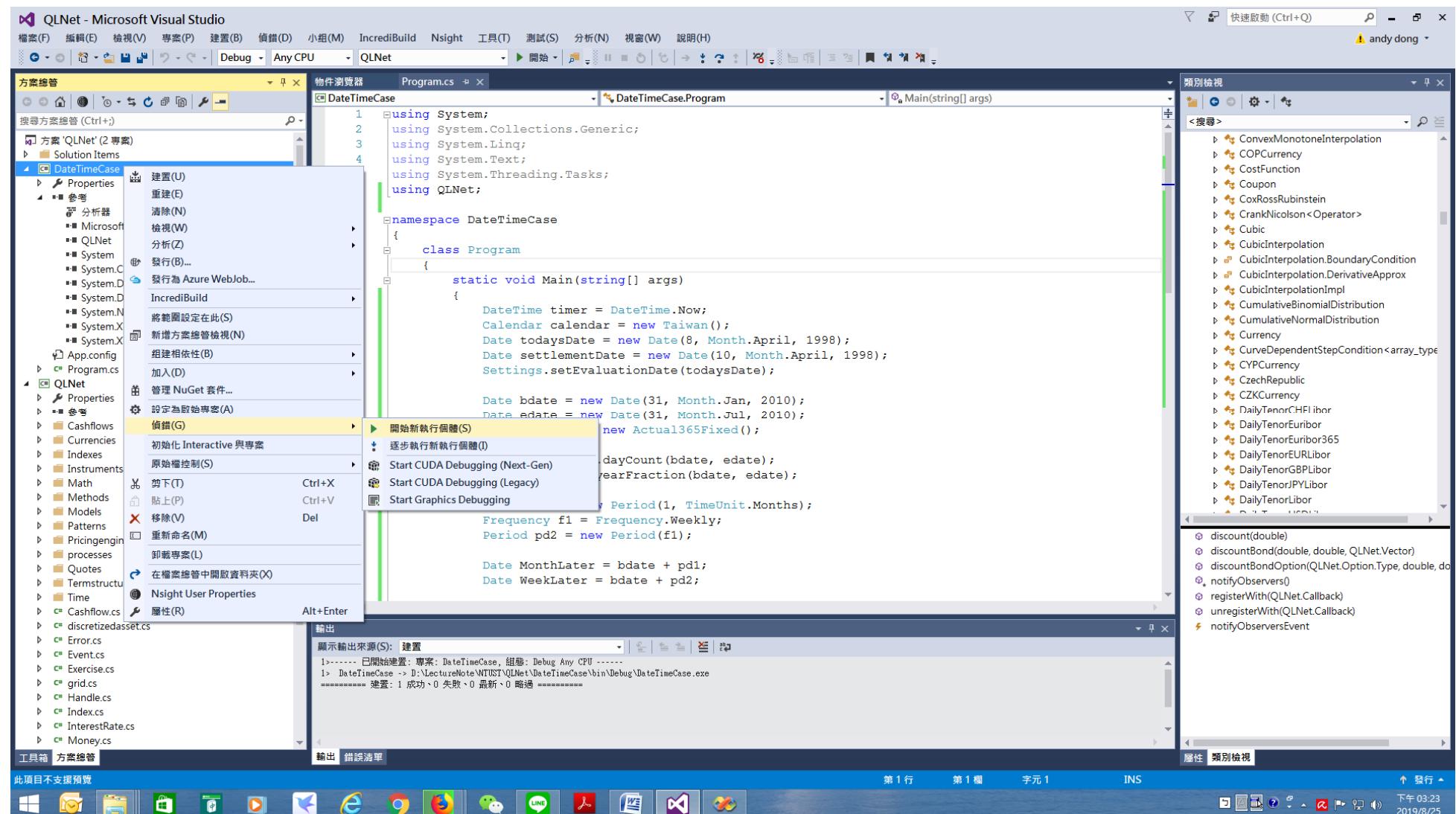


◆ Build OK

The screenshot shows the Microsoft Visual Studio interface with the following details:

- Title Bar:** QLNet - Microsoft Visual Studio
- Menu Bar:** 檔案(F)、編輯(E)、檢視(V)、專案(P)、建置(B)、偵錯(D)、小組(M)、Incredibuild、Nisight、工具(T)、測試(S)、分析(N)、視窗(W)、說明(H)
- Toolbox:** 方案總管、物件瀏覽器、Program.cs
- Code Editor:** Program.cs (DateTimeCase.Program) contains C# code for calculating dates and periods.
- Output Window:** 輸出 (顯示輸出來源(S): 建置)
1>----- 開始建置: 專案: DateTimeCase, 組態: Debug Any CPU -----
1> DateTimeCase -> D:\LectureNote\NTUST\QLNet\DateTimeCase\bin\Debug\DateTimeCase.exe
===== 建置: 1 成功、0 失敗、0 最新、0 略過 ======
- Category Explorer:** 显示了大量与金融相关的类名，如 ConvexMonotoneInterpolation, COPCurrency, CostFunction, Coupon, CoxRossRubinstein, CrankNicolson<Operator>, Cubic, CubicInterpolation, CubicInterpolationBoundaryCondition, CubicInterpolationDerivativeApprox, CubicInterpolationImpl, CumulativeBinomialDistribution, CumulativeNormalDistribution, Currency, CurveDependentStepCondition<array_type>, CYPCurrency, CzechRepublic, CZKCurrency, DailyTenorCHLibor, DailyTenorEURibor, DailyTenorEURibor365, DailyTenorEURLibor, DailyTenorGBPLibor, DailyTenorJPYLibor, DailyTenorLibor, USDLibor, discount(double), discountBond(double, double, QLNet.Vector), discountBondOption(QLNet.Option.Type, double, do..., notifyObservers(), registerWith(QLNet.Callback), unregisterWith(QLNet.Callback), notifyObserversEvent).

◆ Run--Debug ◇



◆ Output

```
file:///D:/LectureNote/NTUST/QLNet/DateTimeCase/bin/Debug/Dat...
today: 1998/4/8
settlement: 1998/4/10

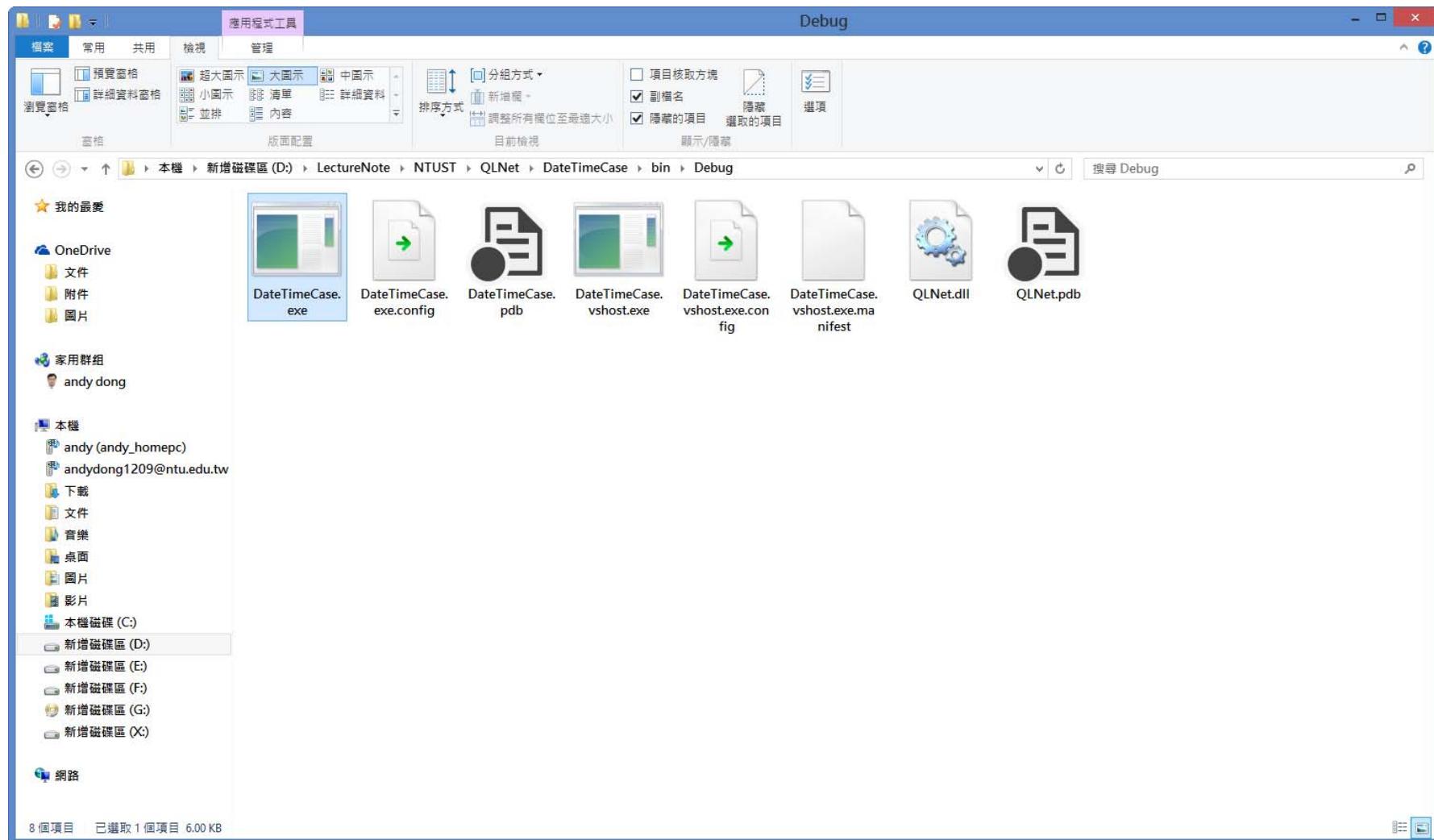
Period begin date: 2010/1/31  end date: 2010/7/31
Number of day: 181
Year Fraction: 0.495890410958904

A Month after begin date: 2010/2/28
A Month after begin date: 2010/2/28
A Week after begin date: 2010/2/7

3rd Friday of April 2010: 2010/4/16

微軟注音 半 :
```

◆ DateTimeCase.exe file location。



1.9 Resources on the Youtube

- ◆ Introduction to QuantLib. Part 1: The installation (Updated) by **Felix Lee**.

<https://www.youtube.com/watch?v=7LR7JK-ebcs>

- ◆ Introduction to Quantlib part 1 Build up an Option by **Carol Zheng**.

https://www.youtube.com/watch?v=e26P26kQsxc&list=PLvXg_ZPrPUfqSFEsI9baI8vP2agnLHCwD&index=2&t=0s

- ◆ Introduction to QuantLib is a talk by **Robert Hardy**.

<https://skillsmatter.com/skillscasts/9208-full-stack-quants-november?tc=3f7216>

- ◆ Introduction to QuantLib and Using QuantLib Programmatically is a talk by **Bojan Nikolic**.

<https://skillsmatter.com/skillscasts/9325-introduction-to-quantlib-and-using-quantlib-programmatically?tc=3f7216>