

Beginning Visual C#[®]

A Computer Programming Tutorial

2015 Edition

By

Philip Conrod & Lou Tylee

©2015 Kidware Software LLC



KIDWARE SOFTWARE

PO Box 701

Maple Valley, WA 98038

<http://www.kidwaresoftware.com>

<http://www.computerscienceforkids.com>

This copy of **Beginning Visual C#** is licensed to a single user. Copies of the course are not to be distributed or provided to any other user. Multiple copy licenses are available for educational institutions. Please contact Kidware Software for license information.

Course Description:

Beginning Visual C# is an interactive, self-paced tutorial providing a complete introduction to the Visual C# programming language and environment. The tutorial consists of 10 lessons explaining (in simple, easy-to-follow terms) how to build a Visual C# application. Numerous examples are used to demonstrate every step in the building process. The tutorial also includes detailed computer projects for you to build and try. **Beginning Visual C#** is presented using a combination of course notes (written in Microsoft Word format) and many Visual C# examples and projects.

Course Prerequisites:

To use **Beginning Visual C#**, you should be comfortable working within the Windows environment, knowing how to find files, move windows, resize windows, etc. No programming experience is needed. You will also need the ability to view and print documents saved in Microsoft Word format. This can be accomplished in one of two ways. The first, and easiest, is that you already have Microsoft Word (or a compatible equivalent) on your computer. The second way is that you can download the Microsoft Word Viewer. This is a free Microsoft product that allows viewing and printing Word documents - it is available for download at:

<https://www.microsoft.com/en-us/download/details.aspx?id=4>

Finally, and most obvious, you need to have Microsoft Visual Studio 2015 Community Edition. This is a separate product that must be obtained. It is available for free download from Microsoft. Follow this link for complete instructions for downloading and installing Visual Studio 2015 Community Edition on your computer:

<https://www.visualstudio.com/products/free-developer-offers-vs>

Installing Beginning Visual C#:

The course notes and code for **Beginning Visual C#** are included in one or more ZIP files. Use your favorite 'unzipping' application to write all files to your computer. (If you've received the course on CD-ROM, the files are not zipped and no unzipping is needed.) The course is included in the folder entitled **BeginVCS**. This folder contains two other folders: **BVCS Notes** and **BVCS Projects**.

The **BVCS Notes** folder includes all the notes needed for the class. Each file in this folder has a DOC extension and is in Microsoft Word format. The files are:

StartHere.doc	This file in Word format
Contents.doc	Course Table of Contents
Class1.doc	Class 1. Introducing Visual C#
Class2.doc	Class 2. The Visual C# Environment
Class3.doc	Class 3. Your First Visual C# Project
Class4.doc	Class 4. Project Design, Forms, Command Buttons
Class5.doc	Class 5. Labels, Text Boxes, Variables
Class6.doc	Class 6. UpDown Controls, Decisions, Random Numbers
Class7.doc	Class 7. Icons, Group Boxes, Check Boxes, Radio Buttons
Class8.doc	Class 8. Panels, Mouse Events, Colors
Class9.doc	Class 9. Picture Boxes, Arrays
Class10.doc	Class 10. Timers, Animation, Keyboard Events
Projects.doc	Additional Projects

The **BVCS Projects** folder includes all the Visual C# projects developed during the course.

How To Take the Course:

Beginning Visual C# is a self-paced course. The suggested approach is to do one class a week for ten weeks. Each week's class should require about 3 to 6 hours of your time to grasp the concepts completely. Prior to doing a particular week's work, open the class notes file for that week and print it out. Then, work through the notes at your own pace. Try to do each example as they are encountered in the notes. Work through the projects in Classes 3 through 10 (and the Bonus class). If you need any help, all completed projects are included in the **BVCS Projects** folder.

About The Authors

Philip Conrod holds a BS in Computer Information Systems and a Master's certificate in the Essentials of Business Development from Regis University. Philip has been programming computers since 1978. He has authored, co-authored and edited numerous beginning computer programming books for kids, teens and adults. Philip has also held various Information Technology leadership roles in companies like Sundstrand Aerospace, Safeco Insurance Companies, FamilyLife, Kenworth Truck Company, and PACCAR. Today, Philip serves as the Chief Information Officer for a large manufacturing company based in Seattle, Washington. In his spare time, Philip serves as the President of Kidware Software, LLC. Philip makes his home with his three lovely and “techie” daughters in Maple Valley, Washington.

Lou Tylee holds BS and MS degrees in Mechanical Engineering and a PhD in Electrical Engineering. Lou has been programming computers since 1969 when he took his first Fortran course in college. He has written software to control suspensions for high speed ground vehicles, monitor nuclear power plants, lower noise levels in commercial jetliners, compute takeoff speeds for jetliners, locate and identify air and ground traffic and to let kids count bunnies, learn how to spell and do math problems. He has written several on-line texts teaching Visual Basic, Visual C# and Java to thousands of people. He taught a beginning Visual Basic course for over 15 years at a major university. Currently, Lou works as an engineer at a major Seattle aerospace firm. He is the proud father of five children and proud husband of his special wife. Lou and his family live in Seattle, Washington.

About The Authors

Philip Conrod holds a BS in Computer Information Systems and a Master's certificate in the Essentials of Business Development from Regis University. Philip has been programming computers since 1978. He has authored, co-authored and edited numerous beginning computer programming books for kids, teens and adults. Philip has also held various Information Technology leadership roles in companies like Sundstrand Aerospace, Safeco Insurance Companies, FamilyLife, Kenworth Truck Company, and PACCAR. Today, Philip serves as the Chief Information Officer for a large manufacturing company based in Seattle, Washington. In his spare time, Philip serves as the President of Kidware Software, LLC. Philip makes his home with his three lovely and "techie" daughters in Maple Valley, Washington.

Lou Tylee holds BS and MS degrees in Mechanical Engineering and a PhD in Electrical Engineering. Lou has been programming computers since 1969 when he took his first Fortran course in college. He has written software to control suspensions for high speed ground vehicles, monitor nuclear power plants, lower noise levels in commercial jetliners, compute takeoff speeds for jetliners, locate and identify air and ground traffic and to let kids count bunnies, learn how to spell and do math problems. He has written several on-line texts teaching Visual Basic, Visual C# and Java to thousands of people. He taught a beginning Visual Basic course for over 15 years at a major university. Currently, Lou works as an engineer at a major Seattle aerospace firm. He is the proud father of five children and proud husband of his special wife. Lou and his family live in Seattle, Washington.

Acknowledgements

I would like to thank my three wonderful daughters - Stephanie, Jessica and Chloe, who helped with various aspects of the book publishing process including software testing, book editing, creative design and many other more tedious tasks like textbook formatting and back office administration. I could not have accomplished this without all your hard work, love and support. I also want to thank my best friend Jesus who always stands by my side giving me wisdom and guidance.

Last but definitely not least, I want to thank my multi-talented co-author, Lou Tylee, for doing all the real hard work necessary to develop, test, debug, and keep current all the 'kid-friendly' applications, games and base tutorial text found in this book. Lou has tirelessly poured his heart and soul into so many previous versions of this tutorial and there are so many beginners who have benefited from his work over the years. Lou is by far one of the best application developers and tutorial writers I have ever worked with. Thanks Lou for collaborating with me on this book project.

Contents

Course Description	xiii
Course Prerequisites	xiii
System Requirements	xiii
Installing and Using the Downloadable Solution Files.....	xiv
Installing Beginning Visual C#	xiv
How To Take the Course.....	xiv
Forward by Alan Payne, A Computer Science Teacher.....	xv

1. Introducing Visual C#

A Brief History of Visual C#	1-1
Let's Get Started.....	1-4
Starting Visual C#.....	1-6
Opening a Visual C# Project	1-9
Running a Visual C# Project	1-14
Stopping a Visual C# Project	1-19
Stopping Visual C#	1-20
Summary.....	1-21

2. The Visual C# Design Environment

Review and Preview	2-1
Parts of a Visual C# Project	2-2
Parts of the Visual C# Environment.....	2-5
Starting a New Visual C# Project.....	2-6
Main Window.....	2-7
Solution Explorer Window.....	2-8
Design Window	2-9
Toolbox Window.....	2-10
Properties Window	2-11

Moving Around in Visual C#.....	2-13
Solution Explorer Window.....	2-13
Properties Window.....	2-14
Code Window	2-16
Summary.....	2-19

3. Your First Visual C# Project

Review and Preview	3-1
Steps in Building a Visual C# Project	3-2
Placing Controls on the Form.....	3-4
Example.....	3-5
Setting Control Properties (Design Mode)	3-7
Naming Controls	3-11
Setting Properties in Run Mode.....	3-13
How Control Names are Used in Event Methods	3-15
Writing Event Methods.....	3-17
Example.....	3-19
Summary.....	3-27

4. Project Design, Forms, Buttons

Review and Preview	4-1
Project Design	4-2
Saving a Visual C# Project	4-4
On-Line Help	4-5
The Form Control.....	4-8
Properties.....	4-8
Example.....	4-9
Events.....	4-10
Typical Use of Form Control.....	4-11
Button Control	4-12
Properties.....	4-12
Example.....	4-14
Events.....	4-17
Typical Use of Button Control.....	4-17
C# - The First Lesson	4-18
Event Method Structure.....	4-18
Some C# Programming Rules	4-19
Assignment Statement.....	4-21
Property Types.....	4-22
Comments	4-25
Project - Form Fun	4-26
Project Design.....	4-26

Place Controls on Form.....	4-26
Set Control Properties.....	4-27
Write Event Methods.....	4-30
Run the Project.....	4-35
Other Things to Try.....	4-36
Summary.....	4-37

5. Labels, Text Boxes, Variables

Review and Preview	5-1
Debugging a Visual C# Project	5-2
Syntax Errors	5-3
Run-Time Errors	5-6
Logic Errors	5-11
Label Control	5-12
Properties.....	5-12
Example.....	5-13
Events.....	5-15
Typical Use of Label Control	5-16
Text Box Control.....	5-17
Properties.....	5-17
Example.....	5-19
Events.....	5-20
Typical Use of Text Box Control	5-21
C# - The Second Lesson	5-22
Variables.....	5-22
Variable Names	5-23
Variable Types.....	5-24
Declaring Variables	5-25
Type Casting	5-28
Arithmetic Operators	5-29
String/Number Conversion Methods.....	5-33
String Concatenation.....	5-37
Project - Savings Account	5-38
Project Design.....	5-38
Place Controls on Form.....	5-38
Set Control Properties.....	5-39
Write Event Methods.....	5-42
Run the Project.....	5-45
Other Things to Try.....	5-46
Summary.....	5-49

6. UpDown Control, Decisions, Random Numbers

Review and Preview	6-1
Numeric UpDown Control	6-2
Properties.....	6-2
Example.....	6-4
Events.....	6-5
Typical Use of Numeric UpDown Control.....	6-5
C# - The Third Lesson	6-6
Logical Expressions	6-6
Comparison Operators.....	6-8
Logical Operators.....	6-11
Decisions - The If Statement.....	6-16
Random Number Generator.....	6-22
Project - Guess the Number Game	6-25
Project Design.....	6-25
Place Controls on Form.....	6-26
Set Control Properties.....	6-27
Write Event Methods.....	6-30
Run the Project.....	6-36
Other Things to Try	6-37
Summary.....	6-39

7. Icons, Group Boxes, Check Boxes, Radio Buttons

Review and Preview	7-1
Icons	7-2
Custom Icons	7-3
Assigning Icons to Forms	7-7
Group Box Control	7-8
Properties	7-8
Placing Controls in a Group Box	7-10
Example	7-11
Typical Use of Group Box Control	7-12
Check Box Control	7-13
Properties	7-13
Example	7-15
Events	7-15
Typical Use of Check Box Control	7-16
Radio Button Control	7-17
Properties	7-18
Example	7-20
Events	7-21
Typical Use of Radio Button Control	7-21
C# - The Fourth Lesson	7-22
Decisions - Switch Structure	7-22
Project - Sandwich Maker	7-25
Project Design	7-25
Place Controls on Form	7-26
Set Control Properties	7-27
Write Event Methods	7-34
Run the Project	7-44
Other Things to Try	7-45
Summary	7-47

8. Panels, Mouse Events, Colors

Review and Preview	8-1
Panel Control	8-2
Properties.....	8-3
Typical Use of Panel Control.....	8-4
Graphics Using the Panel Control.....	8-5
Graphics Methods	8-5
Graphics Objects	8-6
Colors.....	8-8
Example.....	8-11
Pen Objects	8-12
Graphics Coordinates.....	8-14
DrawLine Method	8-15
Graphics Review	8-17
Example.....	8-18
C# - The Fifth Lesson.....	8-23
Mouse Events.....	8-23
MouseDown Event.....	8-24
Example.....	8-26
MouseUp Event	8-29
Example.....	8-29
MouseMove Event.....	8-30
Example.....	8-31
Project - Blackboard Fun	8-32
Project Design.....	8-32
Place Controls on Form.....	8-33
Set Control Properties.....	8-34
Write Event Methods.....	8-38
Run the Project.....	8-45
Other Things to Try	8-47
Summary.....	8-50

9. Picture Boxes, Arrays

Review and Preview	9-1
Picture Box Control	9-2
Properties.....	9-2
Image Property	9-4
Example.....	9-6
SizeMode Property	9-11
Example.....	9-13
Events.....	9-16
Typical Use of Picture Box Control.....	9-17
C# - The Sixth Lesson	9-18
Variable Arrays.....	9-18
C# for Loops	9-21
Block Level Variables.....	9-26
Method Level Variables	9-27
Shuffle Routine	9-28
Project - Card Wars.....	9-34
Project Design.....	9-34
Place Controls on Form.....	9-35
Set Control Properties.....	9-36
Write Event Methods.....	9-41
Run the Project.....	9-50
Other Things to Try	9-51
Summary.....	9-52

10. Timers, Animation, Keyboard Events

Review and Preview	10-1
Timer Control	10-2
Properties	10-2
Events	10-3
Examples	10-3
Typical Use of Timer Control	10-11
C# - The Final Lesson	10-12
Animation - The DrawImage Method	10-12
Image Disappearance	10-17
Border Crossing	10-20
Image Erasure	10-25
Collision Detection	10-28
Keyboard Events	10-33
KeyDown Event	10-34
KeyPress Event	10-37
Project - Beach Balls	10-42
Project Design	10-42
Place Controls on Form	10-43
Set Control Properties	10-44
Write Event Methods	10-48
Run the Project	10-58
Other Things to Try	10-59
Summary	10-61

B. Bonus Projects

Preview.....	B-1
Project 1 - Stopwatch.....	B-2
Project Design.....	B-2
Place Controls on Form.....	B-2
Set Control Properties.....	B-3
Write Event Procedures	B-5
Run the Project.....	B-7
Other Things to Try	B-7
Project 2 - Times Tables	B-8
Project Design.....	B-8
Place Controls on Form.....	B-8
Set Control Properties.....	B-9
Write Event Procedures	B-12
Run the Project.....	B-15
Other Things to Try	B-15
Project 3 - Dice Rolling.....	B-16
Project Design.....	B-16
Place Controls on Form.....	B-16
Set Control Properties.....	B-17
Write Event Procedures	B-20
Run the Project.....	B-21
Other Things to Try	B-22
Project 4 - State Capitals	B-23
Project Design.....	B-23
Place Controls on Form.....	B-24
Set Control Properties.....	B-25
Write Event Procedures	B-28
Run the Project.....	B-34
Other Things to Try	B-35
Project 5 - Memory Game	B-36
Project Design.....	B-36
Place Controls on Form.....	B-37
Set Control Properties.....	B-38
Write Event Procedures	B-45
Run the Project.....	B-52
Other Things to Try	B-53

Project 6 - Units Conversion	B-54
Project Design.....	B-54
Place Controls on Form.....	B-54
Set Control Properties.....	B-55
Write Event Procedures	B-58
Run the Project	B-63
Other Things to Try	B-63

Project 7 - Decode	B-64
Project Design.....	B-64
Place Controls on Form.....	B-64
Set Control Properties.....	B-66
Write Event Procedures	B-69
Run the Project	B-74
Other Things to Try	B-75
Project 8 - Frown.....	B-76
Project Design.....	B-76
Place Controls on Form.....	B-77
Set Control Properties.....	B-78
Write Event Procedures	B-85
Run the Project	B-93
Other Things to Try	B-94
Project 9 - Loan Calculator	B-95
Project Design.....	B-95
Place Controls on Form.....	B-95
Set Control Properties.....	B-96
Write Event Procedures	B-99
Run the Project	B-100
Other Things to Try	B-101
Project 10 - Checkbook Balancer	B-102
Project Design.....	B-102
Place Controls on Form.....	B-102
Set Control Properties.....	B-103
Write Event Procedures	B-108
Run the Project	B-112
Other Things to Try	B-113
Project 11 - Portfolio Manager	B-114
Project Design.....	B-114
Place Controls on Form.....	B-114
Set Control Properties.....	B-115
Write Event Procedures	B-121
Run the Project	B-126
Other Things to Try	B-127
Project 12 - Bonus Pong Game.....	B-128

Course Description:

Beginning Visual C# is an interactive, self-paced tutorial providing a complete introduction to the Visual C# programming language and environment. The tutorial consists of 10 lessons explaining (in simple, easy-to-follow terms) how to build a Visual C# application. Numerous examples are used to demonstrate every step in the building process. The tutorial also includes detailed computer projects for you to build and try. **Beginning Visual C#** is presented using a combination of course notes (written in Microsoft Word format) and many Visual C# examples and projects.

Course Prerequisites:

To use **Beginning Visual C#**, you should be comfortable working within the Windows environment, knowing how to find files, move windows, resize windows, etc. No programming experience is needed

Finally, and most obvious, you need to have Microsoft Visual C#. This is a separate product that must be obtained. It is available as a free download from Microsoft. Follow this link for complete instructions for downloading and installing Microsoft Visual Studio 2015 Community Edition on your computer:

<https://www.visualstudio.com/products/free-developer-offers-vs>

System Requirements

You will need the following hardware and software to complete the exercises in this book:

- ⌚ Microsoft Windows 7, 8, or 10
 - ⌚ Microsoft Visual Studio 2015 Community Edition
 - ⌚ 1.6 Ghz Pentium or compatible processor
 - ⌚ 1 GB (32 Bit) or 2 GB (64 Bit) RAM (Add 512 MB if running in a virtual machine)
-

- ⌚ 10 GB of available hard disk space
 - ⌚ 5400 RPM hard drive
 - ⌚ DirectX 9 capable video card running at 1024 x 768 or higher-resolution display
 - ⌚ Video Monitor (1024 x768)
 - ⌚ DVD-ROM Drive
 - ⌚ Microsoft Mouse or compatible pointing device
-

Installing and Using the Downloadable Solution Files

If you purchased this textbook directly from our website you received an email with a special and individualized internet download link where you could download the compressed Program Solution Files. If you purchased this book through a 3rd Party Book Store like Amazon.com, the solutions files for the Beginning Visual C# Tutorial are included in a compressed ZIP file that is available for download directly from our website at:

<http://www.kidwaresoftware.com/BVCS2015-solutions.htm>

Please complete the online web form at this webpage above with your name, shipping address, email address, the exact title of this book, date of purchase, online or physical store name, and your order confirmation number from that store. We also ask you to include the last 4 digits of your credit card so we can match it to the credit card that was used to originally purchase this textbook. After we receive and verify all this information we will email you a download link for the source code and multi-media solution files associated with this book.

Warning: If you purchased this book "used" or "second hand" you are NOT licensed or entitled to download the Program Solution Files. However, you can purchase the Digital Download Version of this book at a highly discounted price which allows you access to the digital source code solutions files required for completing this tutorial.

Installing Beginning Visual C#:

The course notes and code for **Beginning Visual C#** are included in one or more ZIP files. Use your favorite 'unzipping' application to write all files to your computer. The course is included in the folder entitled **BeginVCS**. The **BVCS Projects** folder includes all the Visual C# projects developed during the course.

How To Take the Course:

Beginning Visual C# is a self-paced course. The suggested approach is to do one class a week for ten weeks. Each week's class should require about 3 to 6 hours of your time to grasp the concepts completely. Prior to doing a particular week's work, open the class notes file for that week and print it out. Then, work through the notes at your own pace. Try to do each example as they are encountered in the notes. Work through the projects in Classes 3 through 10 (and the Bonus class). If you need any help, all completed projects are included in the **BVCS Projects** folder.

Forward by Alan Payne

What is Beginning Visual C# and how it works.

These lessons are a highly organized and well-indexed set of lessons in the Visual C# programming environment. Visual C# is a programming environment which allows the user to drag and drop buttons, text boxes, scroll bars, timers and dozens of other visual "controls" to make programs which look like "Windows" programs. They provide a graphical user interface to the user - rather than a text only interface as would be the case if you make "Console C#" projects.

The tutorials provide the benefit of completed real-world applications - fully documented projects from the teacher's point of view. That is, while full solutions are provided for the teacher's (and learner's) benefit, the projects are presented in an easy-to-follow set of lessons explaining the rationale for the form layout, coding design and conventions, and specific code related to the problem. The learner may follow the tutorials at their own pace while focusing upon context relevant information. Every bit of the lesson is remembered as it contributes to the final solution to a real-life application. The finished product is the reward, but the student is fully engaged and enriched by the process. This kind of learning is often the focus of teacher training. Every computer science teacher knows what a great deal of work required for projects to work in this manner, and with these tutorials, the work is done by an author who understands the classroom experience. That is extremely rare!

Graduated Lessons for Every Project ... Lessons, examples, problems and projects. Graduated learning. Increasing and appropriate difficulty... Great results.

With these projects, there are lessons providing a comprehensive background on the programming topics to be covered. Once understood, concepts are easily applicable to a variety of applications. Then, specific examples are drawn out so that a learner can practice with the Visual C# form designer. Conventions relating to naming controls and the scope of variables are explained. Then specific coding for the example is provided so that the user can see all the parts of the project come together for the finished product.

After the example is completed, then short problems challenge the user to repeat the process on their own, and finally, Projects provide a "summative" for the unit.

By presenting lessons in this graduated manner, students are fully engaged and appropriately challenged to become independent thinkers who can come up with their own project ideas and design their own forms and do their own coding. Once the process is learned, then student engagement is unlimited! I have seen student literacy improve dramatically as they cannot get enough of what is being presented.

Indeed, lessons encourage *accelerated* learning - in the sense that they provide an enriched environment to learn computer science, but they also encourage *accelerating* learning because students cannot put the lessons away once they start! Computer Science provides this unique opportunity to challenge students, and it is a great testament to the authors that they are successful in achieving such levels of engagement with consistency.

My history with the Kidware Software products.

I have used single license or shareware versions for over a decade to keep up my own learning. By using these lessons, I am able to spend time on things which will pay off in the classroom. I do not waste valuable time ensconced in language reference libraries for programming environments and help screens which can never be fully remembered! These projects are examples of how student projects should be as final products - thus, the pathway to learning is clear and immediate in every project.

By following these lessons, I was able to come up with my own projects - An Equation Solver which allows a student to solve any equation that they are likely to encounter in high school, a dice game of Craps, a Financial Calculator covering all grade 12 Financial Math applications, and finally, the game of Mastermind - where I presently have a "Mastermind Hall of Fame" for the best solutions by students over the years. I have made several applications for hardware interfacing in Computer Technology class. *I could do all of this only because of these lessons by Kidware Software!*

The exciting thing is that all of the above can now be done in Visual C#, when I learned to do the programming initially using Kidware Software's "*Learn Visual Basic*". For me to go from one language to another is now an inevitable outcome! With these lessons, I am able to concentrate on the higher order thinking skills presented by the problem, and not be chained to a language reference in order to get things done!

If I want to use or expand upon some of the projects for student use, then I take advantage of site-license options. I have found it very straight forward to emphasize the fundamental computer science topics that form the basis of these projects when using them in the classroom. I can list some computer science topics which everyone will recognize, regardless of where they teach - topics which are covered expertly by these tutorials:

- ⌚ Data Types and Ranges
- ⌚ Scope of Variables
- ⌚ Naming Conventions
- ⌚ Decision Making
- ⌚ Looping
- ⌚ Language Functions - String, Date, Numerical
- ⌚ Arrays, Control Arrays
- ⌚ Writing Your own Methods and Classes and more... it's all integrated into the tutorials.

Any further topics found in secondary school topics (recursive functions, sorting algorithms, advanced data structures such as Lists and Linked Lists, Stacks, Queues, Binary Trees, etc...) derive directly from those listed above. Nothing is forgotten. All can be integrated with the lessons provided.

Quick learning curve for teachers! How teachers can use the product:

Having projects completed ahead of time can allow the teacher to present the design aspect of the project FIRST, and then have students do all of their learning in the context of what is required in the finished product. This is a much faster learning curve than if students designed all of their own projects from scratch. Lessons concentrating on a unified outcome for all makes for much more streamlined engagement for students (and that is what they need,

especially in grades 9 and 10), as they complete more projects within a short period of time and there is a context for everything that is learned.

After the process of form-design, naming controls and coding has been mastered for a given set of Visual C# controls, then it is much more likely that students can create their own problems and solutions from scratch. Students are ready to create their own summative projects for your computer science course!

Meet Different States and Providences Curriculum Expectations and More

Different states and provinces have their own curriculum requirements for computer science. With the Kidware Software products, you have at your disposal a series of projects which will allow you to pick and choose from among those which best suit your curriculum needs. Students focus upon design stages and sound problem-solving techniques from a computer-science perspective. In doing so, they become independent problem-solvers, and will exceed the curricular requirements of secondary schools everywhere.

Computer Science topics not explicitly covered in tutorials can be added at the teacher's discretion. For example, recursive functions could be dealt with in a project which calculates factorials, permutations and combinations with a few text boxes and buttons on a form. Students learn to process information by collecting it in text boxes, and they learn to code command buttons. That is all that is required for this one example of a project-extension. The language, whether it is Visual C#, or Visual Basic, or Java, etc... is really up to the teacher!

Useable projects - out of the box !

The specific projects covered in the Beginning Visual C# tutorials are suitable for grade 9 and above:

Project 1 - Stopwatch	B-2
Project 2 - Times Tables	B-8
Project 3 - Dice Rolling	B-16
Project 4 - State Capitals	B-23
Project 5 - Memory Game	B-36
Project 6 - Units Conversion	B-54
Project 7 - Decode	B-64
Project 8 - Frown	B-76
Project 9 - Loan Calculator	B-95
Project 10 - Checkbook Balancer	B-102
Project 11 - Portfolio Manager	B-114
Project 12 - Bonus Pong Game	B-128

As you can see, there is a high degree of care taken so that projects are age-appropriate.

You can begin teaching the projects on the first day. It's easy for the teacher to have done their own learning by starting with the solution files. Then, they will see how all of the parts of the lesson fall into place. Even a novice teacher could make use of the accompanying lessons. The lessons will provide more than just the coding of the solution - they will provide the correct context for the coding decisions which were made, and provide help in the investigation of related functions. Students then experiment with projects of their own making.

How to teach students to use the materials.

Teachers can introduce the style of presentation (*lesson, examples, problem, projects*) to the students in such a way that they quickly grasp how to use the lessons on their own. The lessons are provided so that students may trust the order of presentation in order to have sufficient background information for

every project. But the lessons are also highly indexed, so that students may pick and choose projects if limited by time.

Highly organized reference materials for student self-study!

Materials already condense what is available from MSDN (*which tends to be written for adults*) and in a context and age-appropriate manner, so that students remember what they learn. The time savings for teachers and students is enormous as they need not sift through pages and pages of on-line help to find what they need.

How to mark the projects.

In a classroom environment, it is possible for teachers to mark student progress by asking questions during the various design and coding stages. Teachers can make their own written quizzes easily from the reference material provided, but I have found the requirement of completing projects (mastery) sufficient for gathering information about student progress - especially in the later grades.

Lessons encourage your own programming extensions.

Once concepts are learned, it is difficult to NOT know what to do for your own projects.

Once having done my own projects in one language, such as Visual C#, I know that I could easily adapt them to other languages once I have studied the Kidware Software tutorials. *I do not believe there is any other reference material out there which would cause me to make the same claim! In fact, I know there is not as I have spent over a decade looking!*

Having used Kidware Software tutorials for the past decade, I have to say that I could not have achieved the level of success which is now applied in the variety of many programming environments which are currently of considerable interest to kids! I thank Kidware Software and its authors for continuing to stand for what is right in the teaching methodologies which work with kids - even today's kids where competition for their attention is now so much an issue.

Regards,
Alan Payne
Computer Science Teacher
T.A. Blakelock High School
Oakville, Ontario
<http://chatt.hdsb.ca/~paynea>
