Conclusion

If you are reading this conclusion and if you have read carefully the entire book, then please **accept our well-deserved congratulations**! We are certain that you have earned valuable knowledge in the principles of programming that will stick for life. Even if the years pass, even if technology evolves and computers are far from their current state, the fundamental knowledge of data structures in programming and the algorithmic way of thinking as well as the experience gained in solving programming problems will always aid you, if you work in the field of information technology.

Did You Solve All Problems?

If you have **solved all problems from all chapters**, in addition to reading carefully the entire book, then you can proudly **declare yourself a programmer**. Whatever technology you pick up from now on will be child's play. Now that you have grasped the basics and fundamental principles of programming, you'll easily learn to use databases and SQL, develop Web applications and server-side software (e.g. with ASP.NET and WCF), write HTML5 applications, develop for mobile devices and whatever else you'd like. You have a **great advantage over the majority of programmers** who do not know what a hash-table is, how searching in a tree works and what algorithm complexity is. If you have really made the tremendous effort to solve all problems from the book, then you have most certainly **reached a level of fundamental understanding of the concepts of programming** and a programmer's way of thinking, which will aid you for many years.

Have You Encountered Difficulties with the Exercises?

If you haven't solved all exercise problems or at least the vast majority of them, **turn back and solve them**! Yes, it does take a lot of time, but that's the way to learn programming – with a lot of work and effort. You won't learn programming without practicing it diligently!

If you have encountered difficulties, **use the discussion forum** of the courses on fundamentals of programming at the Software Academy, which follow this book: http://forums.academy.telerik.com. Several hundred people have taken these courses and the majority of them have solved all problems and shared their solutions. So, examine them, try solving the problems and then try again without using any guides.

Many **lectures and video tutorials** have been uploaded on the book's Web site (http://www.introprogramming.info). We have **free PowerPoint slides and videos** in English and Bulgarian for each chapter of the book. They will be of great use to you, especially if this is the first time you are getting involved in programming. If you decide to teach C#, programming or data structures and algorithms, the slides and exercises will help you focus on the training and save time preparing the content. It's worth checking them out.

Also, check out the **free courses** available from Telerik Software Academy (http://academy.telerik.com). All of their lectures' study materials and video recordings have been made available for free download on each course's respective Web site. These courses are an excellent follow-up to your progress as software engineers and professionals in software development. All materials (lecture slides, exercises, demos) and some video recordings, both at this book's and at Telerik Academy's Web site, are **available in English**.

How Do You Proceed After Reading the Book?

Maybe you are wondering how you should continue your development as a software engineer. You've laid solid foundations with this book, so it won't be difficult. We can give you the following instructions:

1. **Choose a language and a programming platform**, e. g. C# + .NET Framework, Java + Java EE, Ruby + Rails or PHP + CakePHP. There's nothing wrong with giving up C#. Focus on the technologies your platform supports; you'll learn the corresponding language quickly. For example, if you choose Objective-C and iPhone / iPad / iOS / Xcode programming, the algorithmic way of thinking you have acquired with this book will help you make progress.
2. **Read a book on databases** and learn how to model your application's data using tables and relations between them. Learn how to build queries for selecting and updating data in **SQL**. Learn how to work with a database server, like **Oracle**, **SQL Server** or **MySQL**. The next natural course of action is to acquire some **ORM technology**, like ADO.NET Entity Framework, Hibernate or JPA. You might also try the **NoSQL database** systems available in the public clouds.
3. **Acquire a technology for building dynamic Web sites**. Start with a book on **HTML, CSS, JavaScript** and jQuery, or with our free course on HTML5, CSS3 and JavaScript (http://html5course.telerik.com). Then explore the web development tools your platform supports, such as ASP.NET Web Forms / **ASP.NET MVC** using the .NET Platform and C#, Servlets / JSP / JSF using the Java platform, CakePHP / Symfony / Zend Framework with PHP, Ruby on Rails using Ruby or Django using Python. Learn how to make simple Web sites with dynamic content. Try creating a Web application for mobile devices using some mobile UI toolkit.
4. **Learn to write mobile applications**. Start for example with HTML5 and **Cordova**, try to deploy your apps in the large marketplaces maintained by Google, Apple, Microsoft and Amazon. Try to learn **native mobile**

development (e.g. Java and **Android** development or Objective C and **iOS** development). Create a mobile app (e.g. some game) and deploy it in some major marketplace. Thus you will pass through the entire design / develop /publish cycle and this will give you real-world mobile development experience.

5. Take up working on a **more serious project**, like a Web market or a program for managing warehouse or accounting software. This will give you the opportunity to encounter the practical problems of practical software development. You'll gain the more valuable practical experience and you'll see for yourself that coding advanced software is much more difficult than coding simple programs.
6. **Get a job at a software company!** This is very important. If you have really solved all problems from this book, you'll easily get a job offer. By working on practical software projects you'll learn a great deal of new software technologies, unlike your colleagues, and you'll come to realize that, even though you know a lot about programming, you are only at the very beginning of your career as a software engineer. You'll only get to tackle the **challenges of team work** in practice, and acquire the tools for dealing with them by working on actual software projects at an actual work environment. You'll have to work at least for a few years until you establish yourself as a software development professional. Then, perhaps, you'll remember about this book and you'll realize that you haven't gone wrong by starting with data structures and algorithms rather than directly with Web technologies, databases and mobile development.

Free Courses at Telerik Software Academy

You can save yourself a lot of trouble and nerves, if you decide to go through all of the above steps of your development as a software engineer at **Telerik Software Academy**. You'll learn under the guidance of Svetlin Nakov and instructors with practical experience in the software industry. The Academy is the easiest and absolutely free-of-charge way to lay the foundations of your development career, but it is not the only way. Everything depends on you!

Good Luck to Everyone!

On behalf of the entire panel of authors, we wish you endless success in your career and personal life!

Svetlin Nakov,

Manager of the "Technical Training" Department, Telerik Corporation,

Telerik Software Academy – http://academy.telerik.com

August 24th, 2013