A Great Introductory Book For Beginning Programmers

* Objects First Approach. Students learn to design with objects from the start. In more traditional approaches, students first learn programming basics in the context of procedural programming in the small. Since this frame of reference is essentially useless when attacking large-scale problems, students must later re-learn how to approach problems. Instructors can present material from a point of view that will make sense throughout the curriculum. Presentation and justification of programming principles and good techniques is easier.

* Emphasis on the distinction between class specification and implementation. Students learn to develop coherent class specifications early on, and to build components that conform to carefully defined, consistent specifications. The result is more maintainable, error-free code.

* Early emphasis on testing and test-driven implementation. Students develop a habit of testing as part of the implementation process. Testing is essential to ensure quality programs.

* Current presentation of object-oriented design and Java. Students benefit from seeing general approaches to commonly occurring design patterns in a specific, well-defined context. This will also make it easier for students to get the point when such topics are introduced in upper-level design and software engineering courses. Other features include an emphasis on event-driven interfaces, rather than traditional procedural I/O; informal use of standard UML notation

* Optional interactive exercises are designed for use with the open-source DrJava integrated development environment (IDE) - a popular tool for compiling and testing programs
This book, in my opinion, is probably one of the few better introduction books that should be widely used in many intro computer courses. I have programmed for many years, and I have learned that I have made the same mistake in the past that I know many others have done, when they first started programming. The problem I am talking about is when some people program, they just go up to the keyboard and screen and just type, which I have learned from this book that it is very wrong to do. I have learned from this book that when one programs, he/she has to have some kind of design implementation (UML) set and ready before going to turn the designs to code. As an introductory book, this will teach beginner programmers the correct way to program, without the sloppy coding techniques.

The first two chapters are a good explanation to how computers process code and the meaning of object-orientation. The next couple of chapters gives a brief overview of what Java is and how Java ties in with object-oriented design. If one already has Java experience, I think he/she can just skip these chapters, because it may get repetitive to the eyes. For beginner Java-ers, the book gives excellent pure examples in Java, without any confusing documentation or coding.

Then comes Chapter 10 and on... These chapters covered are for Java programmers who have already passed the Beginner's stage, but feel they need to 'dip their feet' in Java water a little more before declaring themselves as proficient Java-ers. These chapters cover excellent computer terminology and basic computer aspects (lists, sorts, searches, abstractions) and classic algorithmic ideas. The last couple of chapters deal with introduction to Exceptions (Java) and GUI Applets, after dealing with TUI most of the book.

So in closing, a good book for introduction, but if you do have experience in Java, then you still would find the book useful, with a handful of excellent defined computer programming terminology and examples. Also, I took the class with one of the authors (Dr. Jaime Nino) and he is truly an exceptional professor at heart who loves his programming.

For More 5 Star Customer Reviews and Lowest Price:
Introduction to Programming and Object-Oriented Design Using Java by Frederick A. Hosch - 5 Star Customer Reviews and Lowest Price!