

Preface to 1st Edition

Control abstraction was the message of the first programming revolution seen in high-level programming languages such as Algol and Pascal. The focus of the next revolution was data abstraction, which proposed languages such as Modula and Ada.

The object-oriented revolution began slowly in the 1960s with the programming language Simula, but moved onto more languages such as Smalltalk, Objective-C and C++. Java is almost a hybrid between Smalltalk and C++, and has gained widespread acceptance due to its association with the Internet, its availability to a large user base and reusable libraries for programming in a graphical environment.

Our programming lineage has passed through Pascal, C and C++. As with many other programmers, good run-time checks with automatic memory management and a reusable API made Java a very attractive option. After a half-day on the original Java Whitepaper and the early Java online tutorial, we were sold on the Java bandwagon and already writing code. In another two days' time, we were using the Abstract Windowing Toolkit (AWT) package for graphical applications. In situations where there is no large investment into older languages, we are quite happy to abandon them completely.

Effective programming in Java comes from understanding three key areas – object-oriented concepts, the syntax and semantics of the Java programming language and the Java Application Programming Interface (API). This is our emphasis when we conduct professional courses, and in this book as well.

Much of the material in this book is based on previous courses which we have conducted over the past two years to the industry and the National University of Singapore (NUS). Courses conducted for the industry last about 5 to 7 days, depending on the amount of coaching that participants require. In the Department of Information Systems and Computer Science at NUS, a course on “Object-Oriented Methods” runs over 13 weeks.

As you might have noticed, we have taken to Java as ducks to water. Java has allowed us to think about and specify object behavior. This results in executable code which is merely secondary. What is important is the clean specification of object behavior. Similarly, in getting accustomed to working with objects, we believe that you will enjoy it too.

Preface to 2nd Edition

Since publishing the first edition almost 10 years ago, we have seen Java being used in many high school and university programming courses. Further, many projects now use Java as the implementation language. Similarly, at the Institute of Systems Science, we have seen professional developers warming up to Java for the first time in 1998, to those who use Java in their daily work in 2007.

We have thus updated the material to cover J2EE topics such as JDBC, RMI, Serialization and Java Servlets. We have also added a chapter on Generics as the Java language evolved to allow this elegant feature.

For those who might be embarking on a Java journey now, we wish you a pleasant journey and a well-used road map. Many have taken this journey before and are enjoying the fruits of their learning investment.

