

Fast Track to OOP and Java for Business Applications

a special course for creative and adventurous professionals

Java is becoming one of the most widely used programming languages for client-server and web-based applications. The successful programmer of the next decade needs a command of Java in his or her professional repertoire.

Java and COBOL are so radically different that making the transition is daunting for programmers whose first or main programming language is COBOL, a virtual *culture shock*. Since not every COBOL programmer will feel comfortable with that transition, it's unrealistic to expect all members of a large staff of COBOL programmers to become proficient in object-oriented programming using Java. Nevertheless for those programmers who have been successfully designing and developing very high-quality well-structured programs in COBOL and who are eager to adopt new ways of thinking about programming, this course offers that opportunity.

Our deep experience both in working with business application developers and in object-oriented programming gives us insights to ease the COBOL programmer into the new creative thought patterns of OOP and the peculiar rules of the Java language. We don't claim that Java is the ideal language for application development. We point out Java's shortcomings and discuss frankly not only how best to circumvent them but also whether it's worth the bother in certain cases. This course emphasizes techniques and principles of good practice that quickly get participants started developing *good* Java programs.

JAV-09 focuses on typical business application programs and data. Upon successfully completing this course the student is well prepared for our short follow-on courses in special topics, such as web applets (JAV-12), exception handling (JAV-16), and multi-threaded flow (JAV-13).

LENGTH: 6 half-day sessions

PREREQUISITES: JAV-08 ("Java without Objects"); we strongly recommend an interval of at least four weeks for practice and reinforcement between the two
or
Proficiency in both COBOL and C programming

GENERAL DESCRIPTION:

We begin by examining in depth *data type* and *data representation*, laying a foundation for class hierarchies, for the object-oriented paradigm, and for relevant Java facilities including inheritance, encapsulation, and overloading. Within the limitations of Java, our examples emphasize using OOP to model *application domains* rather than just the container data structures and user-interface objects emphasized by most courses and textbooks.

The second session confronts Java's confusing dichotomy between *primitive* built-in data types and so-called "reference types", one of the most difficult concepts for programmers learning Java. We begin introducing classes as data structures, first looking at elementary numeric classes, including **Money**, **Date**, and **Percentage**. We show that after going to a great deal (compared with COBOL) of initial trouble to establish those classes, we've achieved major advantages over COBOL in program maintainability and reliability that will be repaid many times over.

The third session focuses on *character-string* manipulation, emphasizing techniques that support user-friendly interfaces. We develop Java techniques for accomplishing the functionality of COBOL's specialized string-manipulation verbs. We also explore *arrays* and other homogeneous containers and some Java class library members that support them.

In the final session we consider *composite data item* classes (sometimes called "business objects") with inheritance, drawing the distinction between gen-spec (is-a) hierarchy and whole-part (has-a) hierarchy. Examples include **Person**, **Account**, **Product**. We introduce *polymorphic function invocation* and show how it greatly simplifies decision logic.

We conclude by exploring the organization of large Java programs into *packages*, classes, interface specifications, and pseudo-classes.