Welcome to this course on Java development. The course will provide you with a thorough introduction to the Java programming language using object oriented principles. The course will also give you a good grounding in the Java Standard Edition (SE) library, including database access, file handling, and so on.

We hope you enjoy the course!

Contents

2

This slide lists the first few chapters in the course. (The list continues on the next page as well).

Chapter 1 introduces the core principles of Java as a programming language, and shows how to use the Java SDK and Eclipse Integrated Development Environment (IDE).

Chapter 2 explores fundamental Java syntax such as variables, data types, and packages. The chapter also introduces some commonly used standard Java classes, such as `String` and `Math`.

Chapter 3 looks at the flow of control constructs in Java, i.e. how to write loops and if-statements. The chapter also summarizes the main operators in Java, such as +, –, ++, etc.

Chapter 4 is where we start looking at object orientation properly. We show how to define classes, create objects, enforce proper initialization of objects, and so on.

Chapter 5 looks at arrays. An array is a fixed-size collection of items, which you access by index position (starting at 0). Arrays are the simplest way to hold a series of objects in memory.

Chapter 6 looks at some additional Java language features, including variable-argument function lists and enumerations.

Chapter 7 explores various additional Java SE classes, including the `Console` class, the `StringBuilder` class, and regular expressions.

Contents (Continued)

8. Collections and generics
9. Inheritance
10. Interfaces
11. Exceptions and assertions
12. Inner classes
13. File handling
14. Accessing databases using JDBC
15. Multithreading

This slide continues the list of chapters in the course (also see the previous slide).

Chapter 8 introduces the concept of generics, and shows how they are used in conjunction with collection classes. For example, we show how you can create a list of persons, a list of products, etc.

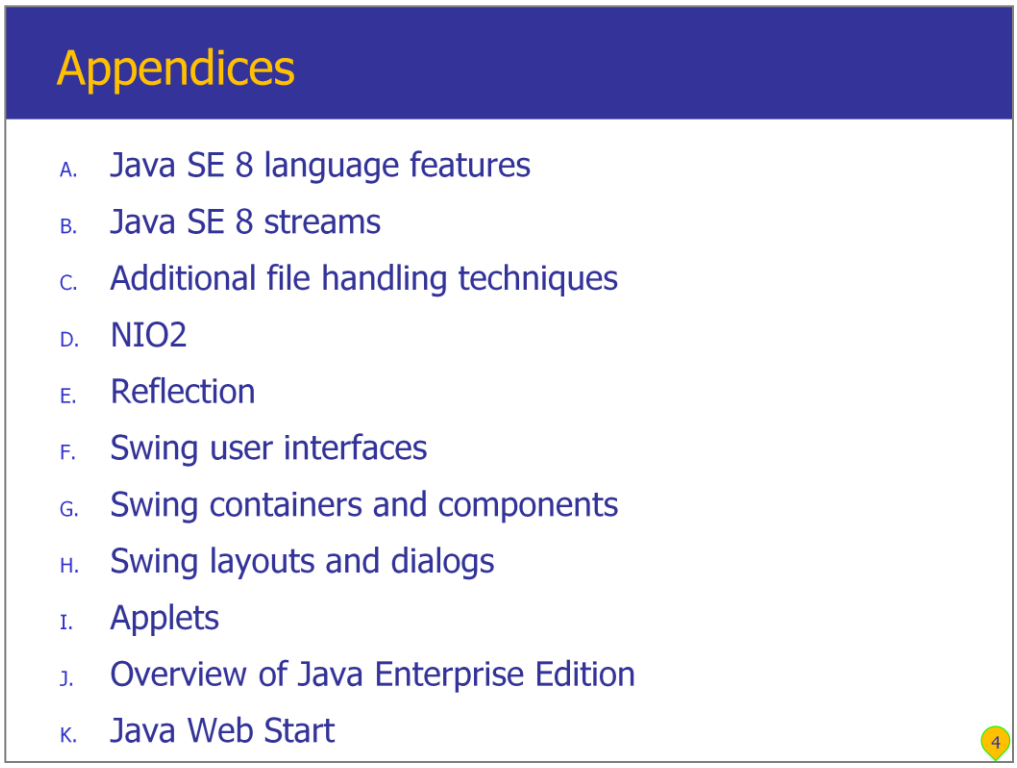
Chapters 9 and 10 are important object-oriented chapters. We describe how to use inheritance to define a class in terms of how it differs from an existing class. We also describe how to use interfaces to specify the capabilities that another class can implement. These are essential techniques in almost every Java application you'll ever write.

Chapter 11 deals with error handling and error detection. We look at how to throw and catch exception objects when something goes wrong, and how to use assertions to stipulate certain conditions are true in your code (this is useful when you're doing unit testing).

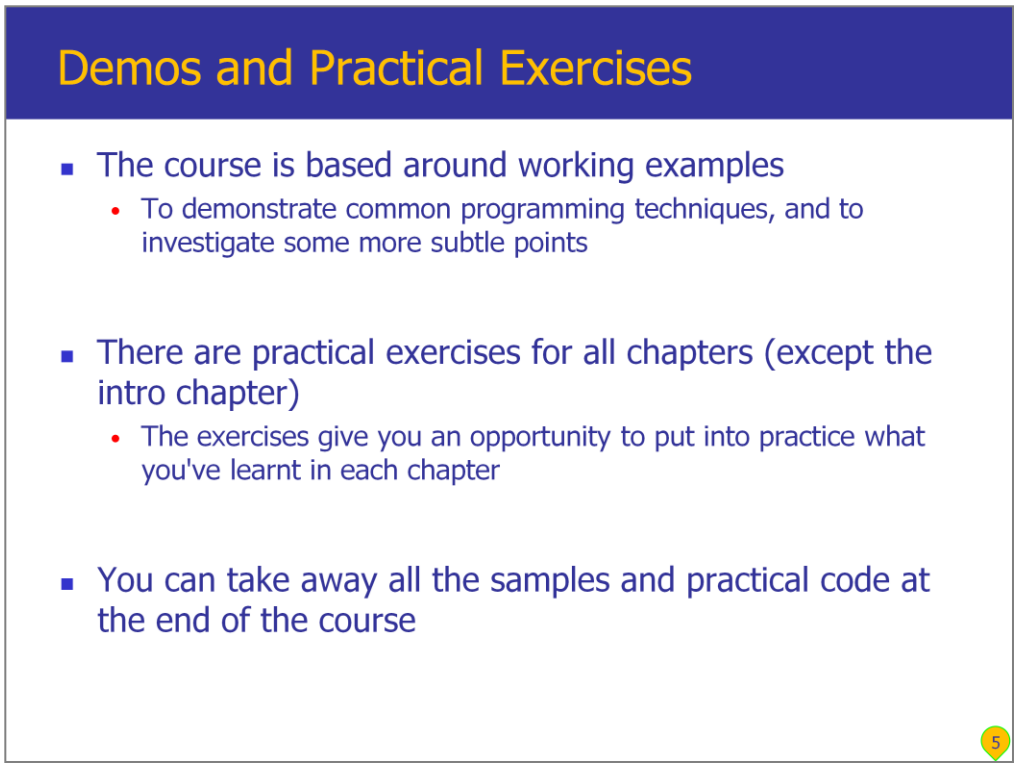
Chapter 12 introduces the concept of inner classes, whereby you can define one class inside the curly brackets of another class. This might seem exotic but it's actually very widely used, as you'll see.

Chapters 13 and 14 look at data management. In chapter 13 we show how to access the local file system, and in chapter 14 we show how to use JDBC to save and query data in relational databases.

Chapter 15 is the final core chapter in the course, and addresses the important issue of multithreading.



This course includes several appendices, which provide auxiliary information to accompany the core chapters. Feel free to peruse these appendices during or after the course. If you'd like to get more information on any of these topics, let the instructor know and we may be able to include some extra discussion on these topics as time permits during the course.



This course is extensively hands-on. For each chapter there's a full set of working demos, plus a lab with full solution code provided.

The standard installation folder for the code is `C:\JavaDev`, and there are separate sub-folders for the demos, lab student code, lab solution code, and databases as follows:

- `C:\JavaDev\Demos`
- `C:\JavaDev\Student`
- `C:\JavaDev\Solutions`
- `C:\JavaDev\Databases`

At the end of the course, you can take away all the demo code and lab code.

We like these courses to be interactive, so if there's anything you don't understand, or if there's anything you'd like to explore in more detail, just ask!