A Java Programming Two Course Sequence

Michael James Payne
Associate Professor of Computer Information Systems and Technology
1421 Knoy Hall, Room 219
Computer Technology Department
Purdue University
West Lafayette, IN 47907-1421
mjpayne@tech.purdue.edu
(765) 494-2566

Track: Information Systems Curriculum

A Java Programming Two Course Sequence

Abstract

Java is very hot right now in the development world. With this, we need to be considering to offer at least one course in our IS or IT course sequences. In this paper, I am going to discuss the transition in our school from a single Java course to a two course Java sequence. I will then discuss the each of the two courses and their contents. After that, I will discuss the reason for the two-course sequence. I will discuss the key topics to be covered in each of these two courses. Finally, I will mention the possibility of even a third course.

Introduction

Java can be found in many curriculums now. The key reason for this is that it is very hot right now and it is also a great language for students to learn the concepts of object oriented programming since it is truly an object oriented programming language. A second course is not found as often at schools. I found this after browsing all the schools with IS departments listed on the IS World web site (2). I believe that one is course is not enough. After consulting this past year on a major Java project, a single course in Java is just not enough to prepare our students to go out and take on real world Java projects.

Java 1 – Our First Java Course

We started our first course in Java two years ago. It was actually started by one of my colleagues. The outline for that course is as follows:

1. Introduction to Java
2. The Java Programming Environment
3. Fundamental Programming Structures in Java
4. Objects and Classes
    a. Introduction to OOP
    b. Using Existing Classes
    c. Starting to Build Your Own Classes
    d. Packages
    e. Class Design Hints
    f. UML diagrams and models
5. Inheritance
    a. First Steps with Inheritance
    b. Casting
    c. Abstract Classes
    d. Interfaces
        i. Properties of Interfaces
        ii. Interfaces and Callbacks
    e. More on Object: the Cosmic Super class
    f. The Class (Run-Time Type Identification)
    g. Protected Access
    h. Design Hints for Inheritance
6. Interfaces and Inner Classes
7. Graphics Programming and Printing
8. Event Handling
9. User Interface Components
10. Applets
11. Data Structures
12. Exceptions and Debugging

The main objective of this course is to prepare students and help them have a good understanding of object-oriented programming. The student should also be able to write moderately Java programs after completing the course.

As to similar courses, after reviewing several courses on the Web, the above outline seems to be a pretty common setup for the first course in Java (1, 2, 3, 4). The course has been very successful and the enrollment continues to rise.

However, there is one area in the course I would like to discuss for possible removal. This is the area of creating GUIs with Java. After attending a professional course in Java and working on a live project in Java, I find that most GUI front ends are done in other languages such as Visual Basic. Even Web pages are generated by tools and not written in Java.

Java is a much too powerful language to be used to build GUIs. Maybe it is better to use that time to cover other topics in the first course rather that GUIs. Below is an alternate outline for a first Java course.

1. Introduction to Java
2. The Java Programming Environment
3. Fundamental Programming Structures in Java
4. Objects and Classes
    a. Introduction to OOP
    b. Using Existing Classes
    c. Starting to Build Your Own Classes
    d. Packages
    e. Class Design Hints
    f. UML diagrams and models
5. Inheritance
    a. First Steps with Inheritance
    b. Casting
    c. Abstract Classes
    d. Interfaces
        i. Properties of Interfaces
        ii. Interfaces and Callbacks
    e. More on Object: the Cosmic Superclass
    f. The Class (Run-Time Type Identification)
    g. Protected Access
    h. Design Hints for Inheritance
6. Interfaces and Inner Classes
7. Applets
8. Data Structures
9. Exceptions and Programming
10. Threads
11. Stream I/O and Files
12. Local Databases

The dropping of GUI opens up enough time for one to at least start on the above-mentioned topics: stream I/O and files, local databases. Both are very important topics in Java. The first allows the student to do some more processing with Java by actually using files, etc. The second topic actually shows them how to connect to at least a local database. By adding these two topics, the students would be better prepared for their first real project such as a semester project or a summer intern project.

Java 2 – Our New Second Java Course

We are about to start our second course in Java. It will be offered for the first time during the Spring 2001 semester. The key to this course is more advanced topics in Java that will really prepare the student to go out and take on live projects associated with the web. A probable outline for this course is as follows:

1. Threads and Multithreading
2. Stream I/O and Files
3. Networking
4. Database Connectivity w/JDBC

    a. Local Databases
    b. Other Databases
5. The Client/Server Environment
    a. Java Server Pages
    b. Servlets
        i. HTTP Basics
        ii. Life Cycle
        iii. Parameters
        iv. Sending HTML information
        v. Session Tracking
        vi. Database Connectivity

The key to this course is to give the student enough background to go into an organization with enough understanding to actually complete or be apart of a Java web project. There will be a continuation of files and databases, even enterprise databases such as Oracle. The addition of threads gives the students the opportunity to develop an application that is truly multi-threading. The addition of networks allows the students to actually deal with network connectivity in the classroom. The last area, client/server, gets students right into the area of java server pages and servlets. This will allow them to development and implement an application in a true client/server environment, possibly even in the enterprise environment over the web.

Java 3 – A Possible Third Course

There is still the possibility for a third Java course. This course would involve bringing together other areas in application develop in addition to Java. The prerequisite for such a course might include: Internet development, Java 1 and 2, Visual Basic 1 and 2, and even HTML. The key to this course is that it would be project-oriented class that could possibly even deal with live projects.

Conclusion

One course in Java in an IS curriculum is mandatory for the today's market. However, that one course is just not enough to prepare our students for the true industrial strength Java of the real world. So, I recommend a second course that will better prepare students for a true Java web project. There is even a possibility of third capstone course in Java or at least that would use Java and other application development areas but that is a topic for another paper or presentation.

References

1. Course Descriptions. (2000, September 5) *IS 483. Business Applications Using Java* [WWW Document]. URL http://www.csulb.edu/~isdept/coursedescriptions.htm

2. ISWorld Net. (2000, September 5) *ISWorld Net IS Undergraduate Name List* [WWW Document]. URL http://mcs.uww.edu/isprogug/namelist.htm

3. Memorial University Java Course Description (2000, September 5) *Introduction to Java* [WWW Document]. URL http://web.cs.mun.ca/~michael/java-course/

4. Polytechnic University Java Course Lecture Notes. (2000, September 5) *Java Lecture Notes* [WWW Document]. URL http://metalab.unc.edu/javafaq/course/