



Constructing a Java Programming Syllabus: Some Observations

Robert Joseph Skovira, Ph.D.
Robert Morris University, PA
rjskovira@worldnet.att.net

Putting a syllabus together for any course or class is always an experience and an intellectual struggle. A syllabus, especially in a programming course, is not merely (or should not be) a copying of the book chapter titles. A syllabus is a means of getting knowledge in the head (or perhaps in textbooks) out into the world, of creating and sharing knowledge in the world. A syllabus is a conceptual structure of a subject area. This essay discusses the construction of a syllabus for a class on Java programming as an example of the dimensionalities of a course syllabus.

For some, a course syllabus is a legal document. Some of the legalities show up in "policy" statements. For example, "Policies: Attendance policy follows the C&IS department's policy (25% missed). Attendance will be taken every session. The student is responsible for obtaining any missed notes and materials and assignments. Make arrangements (phone numbers or email) with other people to do this for you, should you need to miss a session."

For others, it is a plan, a pathway through a discipline, area of study, or subject matter. It is a template for, a conception of, significant notions of a particular subject matter. This plan or way through the thicket of a particular subject matter shows up in the weekly topics of study and discussion. This is usually the bulk of any syllabus. For example:

1. 27 August *Introduction to Course*
2. 3 September *Java Introduced*
Chapter 1 Introduction to Java and Visual J++, 3-44
Geography Lessons in Visual J++
JAR 1 due
3. 10 September *Basic Elements & Structures*
Ch 2 Java Building Elements, 45-78
Ch 3 Control Structures, 79-109
4. 17 September *Object-Oriented Thinking*
Ch 4 Methods, 111-138
Ch 5 Programming with Objects and Classes, 139-183
JAR 2 due

A course syllabus is an intentional document, a rhetorical device to alert an audience, the students, to the shape and form, the profile of a subject. As a rhetorical means, a syllabus may operate as a tool of discovery, a map of already explored parts, or a codified repository of elements to be learned anew by a new generation. This intentional approach first shows up in the course description and in the course objectives. The course description for the undergraduate Java course is: "INFS3151 Java Programming provides the opportunity for students to learn an object-oriented language and to learn object-oriented programming. The course is aimed at learning how to program in Java and developing Java applications and applets. Topics included are Object-oriented Programming, Classes, Objects, Instances, Methods, Applets and Applications, Control Structures in Java, Java Arrays (as Objects), Strings and Characters, Graphics, Multimedia, Exception Handling, Multithreading, Files and Streams, Networking." The objectives of the course are: "To study and learn object-oriented programming, To study and learn object-oriented program design concepts and techniques, To study and learn the object-oriented programming language, Java 2 (JDK 1.2.2), To study and learn about JDK (Java Development Kit, Sun Microsystems), To study and learn about the IDE Visual J++, To gain an understanding of self."

A syllabus is an existential document because it is created with an audience in mind, but one that is situated in a set of circumstances. It is a dynamic state-of-affairs. How it is an existential document shows up in ways much like the following statement: "Any part of this syllabus may be changed by the profes-

sor at any moment in time, depending on where you are, and if there is a valid reason, including the whim of the professor."

A course syllabus reflects a context, and represents situations of performance within that context. The context is an educational one, not merely a training context. There is a different sense as to the reasons why someone may be studying the subject matter, in this case Java. This does not mean that "training" is not addressed, because these concerns commonly show up in the definitions of the situation according to the students. They are studying Java for a variety of reason, from "required" to "want to use it in the workplace". Situations of performance show up in any syllabus in terms of the assignments developed and given and in the manner in which a student's performance is reported on. For example, in this case, a student, when asked to do an assignment, which is commonly a programming assignment, is required to submit the results of the assignment in a report. The report details the assignment problem picked (usually a student has a choice of one from several problems), the algorithm designed to accomplish the task, the code of the Java program, screen shots of the output, and a discussion of any and all problems the student encountered, including error messages, on the way to completing the assignment. This report is designed to insure the engagement of the student in the learning process of doing any assignment. It is integral to the nature of the course.

A course syllabus is also a model of a particular subject matter, purporting to reflect or represent a body of knowledge, a logical space of known objects or propositions, processes, procedures, and vocabularies.

A course syllabus is also an abstract list of things known to practitioners or a community of practice, but represents things to be learned.

A syllabus is a representation of how a professor sees the subject matter and how he or she thinks things ought to be understood.

A course syllabus is a definition of a subject. A course syllabus models the known states-of-affairs of a subject area. The modeling of the known state of affairs in a discipline or area of study is difficult to do, and depends upon and uses other recognized models. These other models commonly show up in textbooks. Usually, one these models is picked to be the basis for a course, but to design a course appropriately, many other models as possible ought to be consulted. For example, the textbook used in this Java course is: Liang, Y. Daniel. (2000). *Introduction to Java programming with Microsoft Visual J++ 6*. Upper Saddle River, NJ: Prentice Hall. However, these other textbooks were used to build the syllabus, and to understand the subject matter behind the categories found in the syllabus: Adams, Joel, Nyhoff, Larry R. and Nyhoff, Jeffrey. (2001). *Java: An introduction to computing*. Upper Saddle River, NJ: Prentice Hall; Anow, David M. and Weirs, Gerald. (2000). *Introduction to programming using Java: An object-oriented approach*. Reading, MA: Addison Wesley; Longman, Cornelius, Barry. (2001). *Understanding Java*. Harlow, England: Pearson Education; Garside, Roger and Mariani, John. (2003). *Java: First contact*, 2e. Pacific Grove, CA: Brooks/Cole Thomson Learning; Hughes, David. (2002). *Fundamentals of computer science using Java*. Boston: Jones and Bartlett; Savitch, Walter. (2001). *Java: An introduction to computer science and programming*, 2e. Upper Saddle River, NJ: Prentice Hall; and Wigglesworth, Joe and Lumby, Paula. (2000). *Java programming: Advanced topics*. Cambridge, MA: Course Technology/Thomson Learning

A syllabus is a management tool of a project of learning. A course syllabus is a cultural artifact, for it exists within institutions, and a social setting. It represents something of value within the social scene. So, it is a social fact which spawns other facts of behavior (grades for example). These other facts are emotional and cognitive facts.

A course syllabus is a philosophical treatise because it rests on ontological and epistemological assumptions. It is an ontology of a worldview. In this case, it is the world of Java and the world of object-oriented programming.

A course syllabus is a lexicon, an introduction to a vocabulary. The vocabulary is the vocabulary of Java programming language, and its semantics and syntax. It is also a taxonomy of categories. The categories name and collect experiences, things, events, and bric-a-brac into coherent classification

systems. A syllabus is an information architecture of a subject matter. In most (if not all) instances, a particular syllabus is only one of many possible architectures of a subject. Usually, specific textbooks offer a specific architecture to follow. A syllabus represents in the view of its creator an architecture designed to optimally traverse the subject matter.

0 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/proceeding-paper/constructing-java-programming-syllabus/32251

Related Content

Sustainability

Yannis A. Phillis (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 6935-6947).

www.irma-international.org/chapter/sustainability/113163

Information Society, Digital Divide, and E-Governance in Developing Countries

Sukaina Al-Nasrawi and Saleem Zoughbi (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 6827-6835).

www.irma-international.org/chapter/information-society-digital-divide-and-e-governance-in-developing-countries/113148

Towards Knowledge Evolution in Software Engineering: An Epistemological Approach

Yves Wautelet, Christophe Schinckus and Manuel Kolp (2010). *International Journal of Information Technologies and Systems Approach* (pp. 21-40).

www.irma-international.org/article/towards-knowledge-evolution-software-engineering/38998

Characteristics of Resources and the Impact on Projects

Loredana Arana (2015). *Encyclopedia of Information Science and Technology, Third Edition* (pp. 5029-5036).

www.irma-international.org/chapter/characteristics-of-resources-and-the-impact-on-projects/112951

Pareto Artificial Life Algorithm for Multi-Objective Optimization

Jin-Dae Song and Bo-Suk Yang (2013). *Interdisciplinary Advances in Information Technology Research* (pp. 100-115).

www.irma-international.org/chapter/pareto-artificial-life-algorithm-multi/74535