# ABAP OBJECTS: DESIGNING A PROGRAMMING COURSE FOR INFORMATION SYSTEMS STUDENTS USING SAP SOFTWARE

**Camille Rogers, Georgia Southern University, cfrogers@georgiasouthern.edu**

## ABSTRACT

*This paper discusses an ABAP Objects programming class for Information Systems students. ABAP is the proprietary language of the SAP enterprise resource system. The most recent version of the SAP platform includes ABAP Objects. This paper provides some helpful suggestions on what resources and topics to teach for this language either at an introductory or advanced level programming course using a classic procedural approach or an objects first approach.*

KEYWORDS: ABAP Objects, SAP, OO Programming, Procedural Programming, ERP, Business Programming

## INTRODUCTION

For many years the primary business programming language was COBOL. Still in use by some legacy systems, it is not as widely used in industry and is taught less frequently in business schools. Languages such as Java, .NET, and C++ are now the universal languages [4] [11] [12] used to teach students in all computing disciplines, with a one-size-fits-all mentality. This is not necessarily the appropriate curriculum for Information Systems majors as they need to learn how to develop software that focuses more on providing a business solution for organizations, in addition to learning fundamental programming syntax and concepts. Several Information Systems faculty have contemplated the dilemma of what to teach to their students [1] [9] [13] [13].

One possible remedy that is getting some attention in business schools, primarily outside the United States [2] [3] [10], is the ABAP programming language. Advanced Business Applications Programming or ABAP is the proprietary language of the SAP enterprise resource systems software. The most recent version of the SAP ERP Netweaver platform uses an object-oriented version known as *ABAP Objects*. Since SAP has roughly 67% of the large scale ERP market share, it is worth considering teaching this particular language to future business software developers. This paper will provide some insight as to the topics and resources being used to teach ABAP Objects in a programming course at one particularly university.

## TARGET AUDIENCE

The target audience or placement of this course in the curriculum is somewhat flexible. ABAP can be taught to introductory business programming students or to advanced classes. The nature of the language is such that ABAP Objects can be taught with or without the full capabilities of object oriented programming. Similar to COBOL, ABAP was initially developed as a procedural language but later adapted the newer OO programming paradigm. In fact, ABAP appears to have many of the same features and syntax of the COBOL language, and it would be very easy for ABAP students to pick up COBOL development or vice-versa, if needed.

This particularly course is taught to students at a university who have already completed a two course sequence in Java, in addition to an introductory course in SAP. The fundamentals of classic procedural programming and OO concepts such as inheritance, encapsulation, superclasses, subclasses, and interfaces were already introduced to students in the Java classes, and thus do not need to be re-introduced in this particular ABAP class. This approach allows the ABAP instructor to cover much more material in one semester. This does not preclude IS faculty however, from selecting and using ABAP as their first programming course, or as a two course sequence with classic procedural programming one semester and an objects content as the second.

## ABAP WORKBENCH

The ABAP Objects programming language serves many functions for the SAP ERP system and is designed specifically for the development of business applications and processes in such areas as materials management, financial, accounting, and human resource information systems. It uses procedural as well as object-oriented programming to handle event driven dialog programming, database retrieval and processing, and report generation.

The ABAP Workbench is the environment that contains all the software tools needed to develop, test, and launch ABAP based business solutions and

is located on the SAP server with entry through the SAP Easy Access menu (see Figure 1).

The primary tools used in the ABAP Workbench for this particular course are:

- ABAP editor – for developing basic program source code
- ABAP Dictionary – for defining and maintenance of data table structures
- Data Browser – for creating and maintaining data entries
- Screen Painter – for designing GUI input screens
- Menu Painter – for designing menu options and icons
- Class Builder – for creating global classes
- Object Navigator – displays all development objects by repository type (database tables, programs, transactions, classes, functions, etc..)
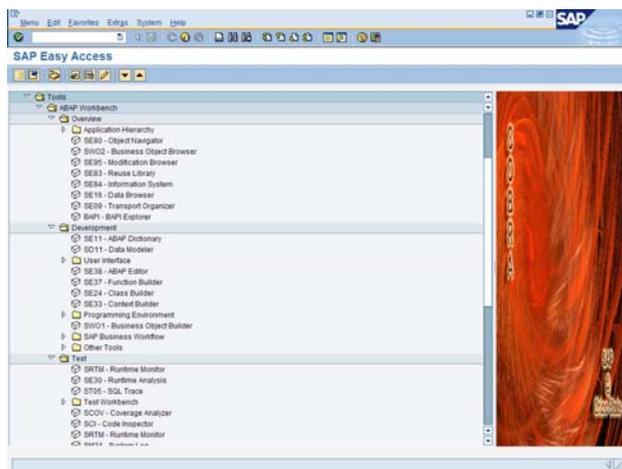


Figure 1: Overview of ABAP Workbench Tools

## COURSE CONTENT

The content of the course covers fundamental ABAP syntax as well as an overview of the many functions and object repositories that can be created. The specific content covered includes:

- Brief overview of the SAP system and business processes
- Structure of ABAP programs
- Introduction to ABAP Workbench & all of the software tools
- Defining and creating tables; data types, elements, and domains
- Parameters, data, and constants
- Basic ABAP syntax and report generation

- Math statements
- ABAP Debugger
- Structures, Transparent tables, and Internal tables; Open SQL
- Processing Files
- Procedural modularization using forms (subroutines)
- Designing input GUI screens, menus, and icons
- Dynpro/dialog programming
- Database inserts, updates, and deletes
- Classes and Objects

## COURSE MATERIALS

Finding course materials for teaching ABAP OBJECTS can be a challenge. This particular course has been taught three different semesters, each time using different materials. It is recommended that instructors who want to teach ABAP attend one of the SAP University Alliance summer workshops or SAP's own professionally taught courses. Many of the original books for ABAP were problematic to use for teaching because they were either written in German or used a technical approach not suitable for academics. They also did not include lectures, teaching notes, and end of chapter exercises. Some of the more recent material has improved tremendously. In the current iteration of the course, taught during the spring semester of 2008, the instructor and students are using the following resources:

- *Discover ABAP* [7]
- *Introduction to ABAP Programming: Course Workbook* [3]

The *Discover ABAP* [7] is a recent publication and it provides a thorough introduction to the ABAP language basics and the ABAP Workbench tools. It can also be used to teach basic programming concepts and data table manipulation. The book includes guided instructions and sample code that can be downloaded from the book website. It does not however provide end of chapter exercises or programming projects.

The *Introduction to ABAP Programming: Course Workbook* [3] developed by Dr. Todd Boyle of St. Francis Xavier University and his capstone project [2] is an excellent complimentary resource for this class. The course workbook includes twelve PowerPoint lecture/lessons with example projects and lab projects for students to complete for assessment. The material does not go into detailed ABAP basics,

but instead is meant as a tutorial introduction to the language with a hands-on business application.

Two additional materials are also used for this course as a supplement and are optional for the students to purchase. *ABAP Basics* [5] and *ABAP Objects* [6] are recent publications release by SAP Press and provide a more technical, but thorough coverage of the classic procedural and OO approaches to the language. These books are not necessarily designed for the beginning programming students. The *ABAP Basics* also provides good coverage of the Object Navigator tool (see Figure 2), which is designed more for the instructors use or by a project manager, and is not mentioned in either of the required resources for the class.
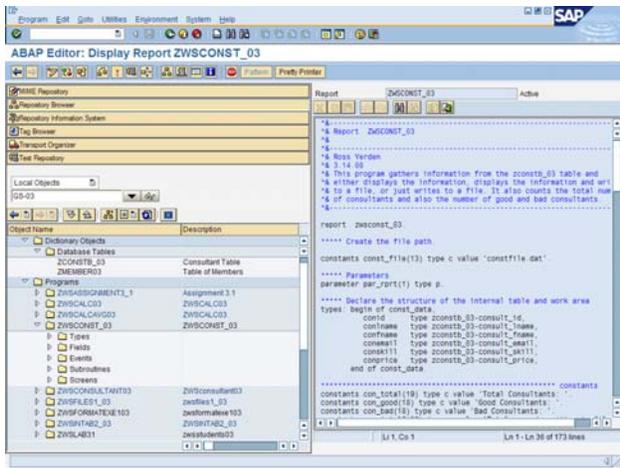


Figure 2: Object Navigator and ABAP Editor

## REFERENCES

[1]     Baker-Eveleth, L., Pendegraft, N., Stone, R.W. (2003). Integrating programming into the IS curriculum, *Proceedings of the 18th Annual Conference of the International Academy of the Information Management*, 133-138.

[2]     Boyle, T. (2007). Computers-for-edu: An Advanced Business Applications Programming (ABAP) Teaching Case, *Journal of Information Systems Education*, 18(2).

[3]     Boyle, T. (2007). Introduction to ABAP Programming: Course Workbook, 3rd edition, *SAP University Alliances curriculum.*

[4]     Cooper, S., D., Dann, W., Pausch, R. (2003). Teaching objects-first in introductory computer science, *Proceedings of the 34th SIGCSE technical symposium on Computer science education*, 191-195.

[5]     Farber, G., Kirchner, J. (2007). *ABAP Basics*. Galileo/SAP Press. German edition first published 2005 by Galileo Press.

[6]     Keller, H., Kruger, S. (2007). *ABAP Objects: ABAP Programming in SAP NetWeaver*, Galileo/SAP Press: Boston, MA.

[7]     Keller, H., Kruger, S. (2002). *ABAP Objects: An Introduction to Programming SAP Applications*. Addison-Wesley/Pearson Education Limited: Harlow, Essex. First published in German in 2001 by Galileo/SAP Press.

[8]     Kuhnhauser, K. H. (2008). *Discover ABAP: A Hands-on Introduction to ABAP from Concepts to Code*. Galileo/SAP Press: Boston, MA.

[9]     MacKinnon, R., Rogers, C., Kung, H., Gardiner, A., Whitworth, J., Rebstock, S. (2006). Creating an ERP Emphasis in the IS Curriculum, *Issues in Information Systems,* 7(1), 284-288.

[10]    McCarthy, B., Hawking, P. (2002). Teaching SAP's ABAP programming language to IS students: Adopting and adapting web-based technologies, *Informing Science*.

[11]    Napier, N. P., Johnson, R.D. (2004). Course assessment: Just how tough is programming in C/C++, *Proceedings of the 2004 International Conference on Informatics Education Research*, 75-83.

[12]    Rogers, C. (2007). Designing an Advanced Programming Class in Java that Incorporates Web Database Applications, *Journal of Computing Sciences in Colleges*, 23(2), 157-162.

[13]    Saxon, C. S. (2004). Incorporating advanced programming techniques in the computer information systems curriculum, *Issues in Information Systems*, 5(1), 288-293.

[14]    Seethamraju, R. (2004). Effectiveness of using an enterprise system (ES) software in business curriculum, *Proceedings of the 2004 International Conference on Informatics Education Research*, 84-91.