

AD-A283 665



SACRED HEART
UNIVERSITY

July 20, 1994

Administrative Grants Officer (AGO)
Office of Naval Research Resident Representative
33 Third Avenue
Lower Level,
New York, NY 10003-9998

RECEIVED
AUG 03 1994
ONR DRAPER

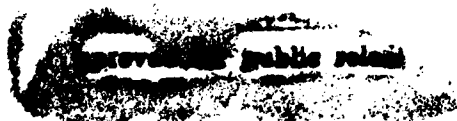
SDTIC
ELECTE
AUG 16 1994
S G D

Dear Sir,

This is the final report of the project "Ada in Introductory Computer Science Courses", BAA #91-18, Category #1 issued to Sacred Heart University Computer Science Department which is a part of the Faculty of Science, Math, and Computer Science. In addition to the time spent on developing CS050 and CS051, this grant presented many opportunities to meet other educators and defence personnel promoting the Ada language. I was able to participate in several Ada and Software Engineering related conferences and arranged for one to be held on our campus. I have met people from ASEET, strong Ada supporters such as Dr. Michael Feldman, educators who have adopted or are about to adopt Ada, and defense contractors in our local area. It has been a wonderful opportunity and experience that I am very grateful for!

The Computer Science Department approved the changes as proposed in the DARPA grant for our introductory courses in the CS curriculum. During the fall of 1992, two sections of CS050, Introduction to Structured Programming were changed to include the programming language Ada. During the spring of 1993, two sections of CS050 and two sections of CS051, Data Structures, were offered using Ada. The courses were taught using Modula-2 prior to fall 1992. The language transition from Modula-2 to Ada was not very difficult.

One of the CS051 sections in spring 93 was taught by an instructor initially unfamiliar with Ada. With Ada handouts and instructions, he successfully taught data structures to students with initial programming background in Pascal and Modula-2. These students were able to read and understand programs in Ada. My CS051 section consisted of students who had CS050 with me and used Ada as their first language. Sacred Heart traditionally receives transfer students from the technical colleges with Pascal as their first language. We also have a large older evening student population who



94 8 15 092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

092

94

8

15

**Best
Available
Copy**

take one or two classes per semester. At times their second programming language course is taken long after their first. Hence we were aware of students coming into CS051 with other structured languages such as Pascal and Modula-2.

An Ada Meridian Compiler was purchased to aid the instructor with work on a personal computer. Students were using a Dec 5500 operating under ULTRIX running Dec Ada.

During the fall of 1992 I was invited to participate in a REUSE EDUCATION WORKSHOP hosted by the West Virginia University, CARDS, ASSET, and AdaNET. The working group I joined was entitled, "Software Reuse in Computer Science Courses". The working group's charge was to identify the characterization, motivational and training needs of professor and students. The group presented an organization of Computer Science courses centered around a philosophy of software reuse. This was a wonderful opportunity to meet others interested in software reuse and computer science curriculum development to share ideas with. The results were published in the "Proceedings for the Reuse Education Workshop 23-24 September 1992".

I had asked for an extension to be able to present some of this work at an appropriate conference and to be able to look into Ada9X. The extension was granted. However, the Ada9X compilers that became available during the 93-94 school year could not run on our dec system or PC's that lack OS-2. However, I used the time to collect more information on Ada9X to implement in our courses in the future, and also found a conference to present the outcome of this project.

The ASEET sponsored conference in Albuquerque, New Mexico in January 1994 provided Ada9X tutorials by the ASEET team and by Norman Cohen. I was able to obtain much information during the 3 day conference that will help me add Ada9X implementations to these courses when Sacred Heart University finds a Ada9X compiler for our VAX 5500. During that time Dave Cook and Eugene Bingue talked about the possibility of coming to our campus to do an Ada workshop. To help educate instructors, students, and other interested parties, and to promote Ada9X, a 3 full day workshop entitled, "The New World of Ada9X", is scheduled for August 3-5, 1994. We are very excited about hosting this event. We are expecting software developers from UT/Norden Systems, UT/Sikorsky Aircraft and other defense contractors to be in attendance. With the help of Catherine McDonald from IDA/ASEET, this "hands-on" ASEET 1994 Ada9X Summer Workshop will be presented by Dr. Dave Cook from the USAF Academy Colorado Springs CO, Eugene Bingue from USAF Phillips Laboratory, Albuquerque NM, and Les Dupaix from Software Technology Support Center, UT, in our classroom laboratory at Sacred Heart University at a nominal cost of \$30 per participant to help defray the cost of slides. Sacred heart is very thankful for this opportunity to host this workshop.

I presented a paper entitled, "Ada, a Software Engineering Tool, in Introductory Computer Science Programming Courses at Sacred Heart University: Mutual Benefits". In the Education session of the Twelfth Annual National Conference on Ada Technology held on March 21-24, 1994, at Norfolk State University in Williamsburg, Virginia. A copy

of this paper is included in the final report. This gave me the opportunity to share my experience teaching these introductory courses with other educators, and to make available the developed course material

The courses went well during the 92-93 and 93-94 school year. The basic programming concepts normally covered in these introductory courses were covered. The students did not find Ada a difficult language to work with on the introductory level and began to learn about software engineering concepts. We are planning to continue using Ada as our introductory language and I would like to adopt it in other courses in the curriculum.

Syllabi, handouts, program assignments, articles, and textbook recommendations were developed for CS050 and CS051. They are included in the final report.

If you need more information, please contact me at:

Sacred Heart University
Faculty of Science, Math, and Computer Science
5151 Park Avenue
Fairfield, CT 06432-1000

phone (203)371-7791 or (203)371-7799

or by email honda@shu.sacredheart.edu

Sincerely yours,



Sandy Honda Adams,
Associate Professor

Accession For	
NTIS CRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	<i>per form 50</i>
By _____	
Distribution /	
Availability Codes	
Dist	Avail and/or Special
<i>A-1</i>	

FINAL TECHNICAL REPORT

ARPA Grant # MDA972-92-J-1031

**(Undergraduate Curriculum & Course Development
in Software Engineering and the Use of Ada)**

"Ada in Introductory Computer Science Courses"

SUBMITTED BY:

**Sandra Honda Adams
Computer Science
Sacred Heart University
5151 Park Avenue
Fairfield, Connecticut 06432-1000**

TO:

**Administrative Grants Officer (AGO)
Office of Naval Research Resident Representative
33 Third Avenue
Lower Level
New York, NY 10003-9998**

TABLE OF CONTENTS

Introduction	1
Summary	3
Time Table	4
Budget	6
CS050 Notes	7
CS051 Notes	10
Appendices	
Bibliography	A
Syllabus	B
Handouts	C
Projects	D
Exams	E
Articles	F

August 12, 1993

Department of The NAVY
Office of NAVAL Research
Resident Representative
33 Third Avenue, Lower Level
New York, New York 10003-9998

Dear Sir:

This is the final technical report of the project "Ada in Introductory Computer Science Courses", BAA #91-18, Category #1 issued to Sacred Heart University Computer Science Department which is a part of the Faculty of Science, Math, and Computer Science.

The Computer Science Department approved the changes as proposed in the DARPA grant for our introductory courses in the CS curriculum. During the fall of 1992, two sections of CS050, "Introduction to Programming", were changed to include the programming language Ada. During the spring of 1993, two sections of CS050 and one section of CS051, "Data Structures", were offered using Ada. The courses were taught using Modula-2 prior to fall 1992. The language transition from Modula-2 to Ada was not very difficult since Modula-2 contains many desirable features found in Ada. General principles of software engineering were also included in these courses.

Many of the features of Ada that lend themselves to software engineering principles (such as data abstraction, information hiding, modularity, reliability, uniformity, modifiability, reuse, completeness, and understandability) were implemented using the language Ada while teaching basic programming concepts. Ada's strong typing facilities, packaging concept, generics, and exception handling were features of Ada included in CS050 and CS051, showing how the above software engineering goals are achieved. Since all CS students in our Information Option and Scientific Option need to take Software Engineering during their Junior year, it was felt that general software engineering principles could be incorporated during

the first two programming courses. The students were given a history of Ada and made aware of the "software crisis" and the need for languages and techniques to handle "programming in large".

The students were provided with course syllabi for each of the courses, examples and homework assignments that explained program development, domain specification, modular style approach to programming, packaging, exception handling and generics (in the Data Structures Course).

As a simple example of what was done, the first assigned program required output of any design of the student's choice using several put "string" procedures. The students' programs were combined into a class package and distributed electronically. Consequently, the concepts of subprogram and packaging were introduced very early in the semester involving the students in an enjoyable application. With the students in a relaxed mode, they created many interesting designs. Having this as a base facilitated the explanation and development of such concepts such as reusability of the "with" context clause, main program development using a case selector statement, the information hiding aspect of the package body, the abstraction idea that can be imbedded in a package specification, etc.

Ada concepts that were taught in the first course, CS050 - user defined data typing (enumerated as well as other user defined types), subprograms (functions and procedures) and packages (specification and body), and exception handling - were used to implement such software engineering principles as reliability, modularity, reusability, information hiding and uniformity. CS051 included more of this plus visibility and scope rules, abstraction, and generics in the coverage of data structures such as stacks, queues, linked lists, and trees. All the programming concepts normally dealt with in our traditional CS050 and CS051 were covered in these revised courses.

During the spring semester, another section of CS051 taught by an instructor who normally teaches CS051, but who was not familiar with Ada, was given handouts and instructions in Ada. He successfully taught data structures to students who had prior background in Pascal and Modula-2. These students were able to read and understand programs in Ada. The students in my section of CS051 had taken CS050 in the fall using Ada as

their first language. Sacred Heart traditionally receives transfer students from the technical colleges and other universities who come with Pascal as their introductory language. We also have older married evening students who with their many responsibilities sometimes need to take a semester off. We were aware of students coming into the Data Structures course with other structured languages such as Pascal and Modula-2 and needed to make certain that using Ada in CS051 would not detract from the teaching of Data Structures. The results indicate that teaching data structures with handouts in all three languages (Ada, Pascal, Modula-2) made no difference in the learning process of data structures. In fact our students liked the added benefit of seeing the similarities and contrasts of the implementation of stacks, queues, linked lists and trees in several languages including Ada.

Some of the students entering the course knowing that Ada was a programming language used by the Department of Defense, seemed skeptical of learning Ada. After seeing some of the rich flavors of Ada and reading articles regarding the development of the air traffic control software in California and the development of banking software in European communities using Ada as the commercial language of choice, students did not seem to mind Ada. The students did not find Ada a difficult language to work with on the introductory level. We are planning to continue using Ada as our introductory language and I would like to adopt it in other courses such as "Compiler Design" and "Database Design" in the curriculum.

After careful evaluation of the courses, I am planning to use a different text book. Over the last several years there have been many text books written using Ada for introductory programming courses and there is a wide selection to choose from. Our students did not find the text I used very helpful in terms of programming examples. Another text has been selected for the fall 1993 classes. Other changes will include the introduction of "is separate" compilations. This was not included in the first year implementation, and will be helpful in modular design.

I would like to present this project and outcome to interested educators at an appropriate conference and will look for an opportunity during the upcoming school year.

An Ada Meridian Compiler was purchased to aid the instructor with work on a personal computer. Students were using a Dec 5500 operating under ULTRIX running Dec Ada.

During the fall of 1993 I was invited to participate in a **REUSE EDUCATION WORKSHOP** co-hosted by West Virginia University, CARDS, ASSET, and AdaNET. The working group I joined was entitled, "Software Reuse in Computer Science Courses". The working group's charge was to identify the characterization, motivational and training needs of professors and students. The group presented an organization of Computer Science courses centered around a philosophy of software reuse. This was a wonderful opportunity to meet and share ideas with others interested in software reuse and CS curriculum development. The results were published in the "*Proceedings for the Reuse Education Workshop 23-24 September 1992*".

If you need more information, please contact me at the following address:

Sandra Honda Adams
Sacred Heart University
Computer Science Department
Faculty of Science, Math, and Computer Science
5151 Park Avenue
Fairfield, CT 06432-1000

Sincerely yours,



Sandy Honda Adams,
Associate Professor