Jamie Penzien

Professor Woodring

CIS 343

7 December 2017

Ada Research Paper

The best way to describe the programming language Ada is to consider it a language compromised of other languages. In the 1980s, the need arose within the Department of Defense for a high-level language that would replace most of its embedded systems. Building the language was one of the most elaborate designs ever completed. Ada supports designs for the government, banking systems, commercial aviation, communications systems, computer-aided design, and manufacturing (6). The critical part of Ada that makes it useful for such systems is its ability to check for exceptions. That, along with Ada’s other strengths will be discussed in this paper.

Ada is a system that allows for high user customization. Programmers, instead of using a set of primitive types, can declare their own types. This allows for very specific routines based around what one’s goal is. Currently #28 on the Tiobe Index, Ada is a language with “simple syntax, structured control statements, flexible data composition, and a mechanism for detecting… run-time conditions” (4, 5). Ada is a structured programming language and each project, down to each function, can have its own types, variables, and procedures. The programs made in Ada can be compiled using GNAT. The following needs to be bound and linked to the .ali and .o files which will result in an executable file.

In the following code, you will see an example that will print out “Hello World!” to the screen (6).

with Text\_To; use Text\_To

procedure hello is

begin

put("Hello World");

end hello

First, we can see the use of with Text\_Io and Text\_Io. These allow us to Put and Get things to the prompt (3). These are like any other important statement, just for Ada instead. Next we have procedure hello is…end procedure. This is like how you would set up a class. A procedure begins and ends when you tell it to. Which brings us to the next line, the actual begin for hello. This will let the compiler know that the code for this class is beginning. And lastly, we have the one command that this code operates – printing hello. This file is saved in hello.adb, compiled with gnatmake hello.adb, and then run by performing ./hello.

Ada supports concurrency with it’s own form of a program called a task. They can communication amongst each other and be considered objects that share data. This language also allows the user to get ‘close’ to the data by providing the ability for a user to specify things in a bit-layout.

However, the most important aspect about Ada is its ability to catch exceptions. When forming a simple calculator program, Ada will warn you when any unplanned event could occur. This language, with its exception handling, promotes the writing of safe and good code that reduces potential errors. This object-oriented language is critical in certain scenarios (the government) because exception handling could potentially save lives. Some more interesting information about Ada includes: it is strongly typed, it allows generic programming, it has libraries available for use, and it supports structured concurrent programming, real-time programming, and can also interface to other languages (5).

While Ada may not be a language taking off and becoming wildly popular, it is still a language to be revered. This language was formulated through a long and thorough research process that implemented various qualities of languages that already existed. Not only should that impress you, but also knowing that this is the type of code our government could possibly use in some of their private work might excite you. Either way, Ada is an easily learnable and enjoyable language that is fun to work with especially given the way it handles exceptions. It makes one think outside of the box and consider ways the program could error-out before compiling.

References

“Ada 2012.” Ada 2012, www.ada2012.org/.

Brukardt, Randy. “Consolidated Ada 2012 Language Reference Manual.” Ada 2012 Language Reference Manual, web.archive.org/web/20160303001815/http://www.ada-auth.org/standards/ada12\_w\_tc1.html.

Kempe, Magnus. “A.10.1 The Packe Text\_IO.” Ada Home. <http://www.adahome.com/rm95/rm9x-A-10-01.html>

“Latest News.” TIOBE Index | TIOBE - The Software Quality Company, www.tiobe.com/tiobe-index/.

“The Ada Programming Language.” About Ada - Adacore, [www.adacore.com/about-ada](http://www.adacore.com/about-ada).

“The Ada Programming Language.” University of Michigan. http://groups.engin.umd.umich.edu/CIS/course.des/cis400/ada/ada.html