

## CIS 3190 A2: Ada Handout

My experience working with Ada was great, it was easy to learn and it was a lot better than Fortran. I needed a lot of hand holding on the first semester, whether it be from resources or getting a friend to explain the algorithm that we were reengineering. However, this time went a lot smoother the only time I needed to refer to something was to look up how you wrote particular things in Ada syntax.

One of the biggest challenges I faced doing this assignment was the algorithm itself, I looked at many of them and tried them out in the languages provided (the knights tour program had different algorithms available online in Python, C, Java) but none of them were complete. Almost all of them just crashed the moment any square besides the (0,0) was used as the starting point and none of the algorithms worked passed an 8 by 8 grid. Though, I eventually sucked it up and tried to implement my own way of choosing a starting point and using a given algorithm I found.

For the most part once I understood the algorithm and choose one that I was going to implement in Ada, coding it was pretty easy. Learning the difference from Ada syntax and what I'm used to (C) was the only time consuming part. Brackets aren't used much for conditions and the way you would do && and || statements in them were also different. Subprograms took a little while also for me to wrap my head around, the whole begin and end parts and how passing arguments worked. After that it was smooth sailing and I had a program that took in hardcoded values and ran the knight's tour. Implementing user input wasn't hard, just took some reading to learn how to scan in inputs. I then just had 3 consecutive command loops used to take user input for the 3 variables (board size and the x and y positions to start the knight on) and pass them into the procedure that solves the knight's tour. It took a bit of testing but I finally got it to work thought if a letter or symbol is entered then the program breaks, I haven't figured out error checking for that though. Finally, I just had to add file out put which was just creating a file and instead of using Put to output to the screen, use the file I created as a parameter to print it to a file.

After I was done, I tested the program and tried so see if I could incorporate some way to get 9 by 9 and 10 by 10 grids to also work with the program I wrote but there was nothing I could do. Maybe I should have found a much harder algorithm that works with all cases now that I know a lot more Ada and it'd be easier to code. I'm happy with my assignment and look forward to using Ada again in another assignment.

## Questions:

### 1. What were the greatest problems faced while designing the algorithm in Ada?

The hardest part designing the algorithm was trying to tell if the algorithm itself will work for all test cases. You didn't know until you wrote it out and tested it. Unfortunately, it was only after writing my Ada code did I realize that it only worked up to a 8 by 8 grid.

### 2. What particular features made Ada a good language?

The error messages that the compilers would return were really helpful. They were oddly specific and actually told you what lines the errors were occurring. Also, the way Ada asks you to specify pre and post conditions allows us to have less logic errors. The only errors I stumbled upon were syntax errors as I learnt the language.

### 3. Would it have been easier to write the program in a language such as C?

If I only had a short amount of time to write this program, then it would have been easier in C as I already know it. However, it's as easy and quick to write the program in Ada as it would be in C as the structure of the languages are similar and the code would also be almost the exact same disregarding the difference in syntax.

### 4. Given your knowledge of programming, was Ada easy to learn?

I found Ada's ease of learning to be quite easy. Thought at first glance it look like a much more elegant version of Fortran but the syntax of this language is so similar to C or Perl. I didn't have any sample code of Ada so I was stumped on understanding how to write in Ada but after reading the provided resources and this handy C to Ada comparison website. Ada looks like an older version of C code because a lot of the syntax and the way you write things are similar, just the minute things like how you write the conditions for an if or while loop, and also to define the value of a variable you would use := instead of just the equals sign in C.

### 5. What structures made Ada usable? (In comparison to C for instance)

The way functions and procedures worked really helped organize the code a little and also allowed me to use recursion to solve the knight's tour program. Also the structure of loops and if statements being really similar to C also helped me with my familiarity.

6. What did you dislike the most about Ada?

What I disliked about Ada was that you couldn't put arrays as parameters for another procedure/function easily. You need to create an array type first then use that when declaring the variable in the procedure/function and if you want to change or manipulate values within the array you have to declare it as an out variable at all.