Lecture C3: Ada syntax

Response to 'Muddiest Part of the Lecture Cards'

(51 respondents)

1) ***Why no PRS questions today, and why don't you recommend any problems we can try on our own***? (2 students)
There will be PRS questions in more or less every lecture. Today we unfortunaley never reached my last slide and it's PRS question. If you want to try out your Ada knowledge, there are homework assignments on each lecture (in the weekly unified problem sets) and the Feldman book is filled with good examples, that all are available via the Feldman-CD.

2) ***Can you type cast in Ada, how to change from Integer to Float***? (2 students)
Yes you can. Examples was shown in class, for example:

1.0 > FLOAT(0) -- change the Integer 0 into being a floating point number
FLOAT(3) * 4.0 -- change the Integer 3 into being a floating point number
3 * INTEGER(4.0) -- change the Float 4.0 into being an integer

3) ***Difference between Ada.Text_Io and Text_Io***? (1 student)
They are both identical. The package Text_Io was used in 'Ada 83'. The package Text_Io is in 'Ada 95' a sub package to Ada. It is kept for consistency reasons, so that old Ada 83 programs still can be used using the new compileras and tools.

4) ***How many different ways are there to store numbers, do they have different functions for each***? (1 student)
Infinitely many ways. Each time you declare a new kind of number type, new functions can be instantiated.

5) ***Can you in a recursive function assign a number or name that already exist. E.g., number_1 := number_1 + 1;***? (1 student)
Yes, all values will be saved on a "stack" and therir values will not be lost. More about details of this on later lecture.

6) ***What is a data type, what is a Float, what is a String, what is an Integer***? (7 students)
A data type clarifies the context for numerical and other operations. Data types are usually associated with values in memory or variables. Any value in memory is just represented by zeros and ones, there is no distiction in hardware between data, instructions, integers, characters, memory addresses, ...

A Float is a predefined and implementation dependent declartion of a real type.

A String is an array of characters

An Integer is a predefined and implementation dependent declartion of a integer type.

7) ***Why does Ada only support one Integer type, while other languages (ie, C++) have several. I.e., long_int, unsigned_int, etc.*** ? (1 student)
That is not correct, Ada can have infinitely many, and Ada has strong typing! In Ada it is possible to have full and complete control over the size of Integers, compared to in C++ where a long might be 64 bits, or 128 or, ...

8) ***Where in code do you define variables? Can it be done among the executable statements***? (1 student)
Can be defined within a 'declare block', which can be among the statements.

9) ***What is formal parameters, what does Item do, what can Put do***? (and similar questions) (2 students)
Put can print characters/data on the screen.

A procedure can have a number of different Formal parameters. When you call the procedure, you will instatiate the Formal parameters with Actual parameters.

Ex: Ada.Text_Io.Put (FormalParameter => ActualParameter);

Typing the Formal parameters like Feldman does in his examples makes the code more 'readable'. It also has a nice feature in that the order you use for the parameters are of no interest. If you on the other hand don't want to type the Formal parameters in your code, you need to know the eaxt order of the parameters. Mora about this in upcoming lecture.

10) ***Must Skip_Line come immediately after Get always***? (1 student)
Must not, but should come before the next Get-instruction to avoid that Get getting garbage as input.

11) ***Difference between New_Line and Skip_Line***? (2 students)
New_Line has to do with PUT instructions. SKIP_LINE has to do with GET instructions.

12)***Will extra spaces in the code affect the compiler*** ? (1 student)
No.

13) ***Synatx to declare variables and constants***? (1 student)
Age : Integer := 25; -- this is a variable declaration and instantiation

Pi : **constant** Float := 3..1415926536; -- this is a constant declaration

14) ***Do sub procedures have to be put in the space between the main "procedure" and "begin"***? (1 student)
Yes for local procedures, or can be within a declare block.

15) ***Can you go over arithmetic and relational operations*** ? (5 students)
I can.

16) ***Can you have more than one "use" command at the top of the program***? (and similar) (3 students)
Absolutely!

```
with Ada.Text_Io;
with Ada.Integer_Text_Io;
with Ada.Float_Text_Io;
with Spider;
use Ada.Text_Io;
use Spider;
use Ada.Float_Text_Io;
use Ada.Integer_Text_Io;
```

17) ***Is there a list of vocabulary/syntax?***? (1 student)
Can be found at many different places, for exaple:
Appendix B in Fledman has the Ada Character Set, Delimiters and Reserved Words.
The Language Reference Manual has all the information. Available both online or on the Feldman CD.
The Feldman CD also has something called 'REFCARDS' which I will distribute in class tomorrow.

18) ***How good is Ada at interacting with database files*** ? (1 student)
If you can do it in C, you can do it in Ada :)

19) ***How do you compile, bind, and execute the program? Do I just press F2, F3, F4***? (1 student)
Using the AdaGIDE tool that is correct. And the combination "Alt + F2" will generate the 'list file'.

20) ***Will we be learning to use Ada for embedded systems in this class***? (1 student)
Chapter 13 in the LRM (Language Reference Manual) will not be covered in this course.

21) ***Restate the difference between the two ways you draw ADA in class today***? (1 student)
First code example solved the problem using 'straight line Ada'. The second solution used procedures, which made the program more readable, easier to maintain/change/use/...

22) ***What is ":=", "=>"***? (1 student)
:= is used for assignment. The following statement "Weight := 42;" Assignes variable Weight the value of 42. Weight has to be declared as an Integer for that operation to be allowed.

=> is pronounced 'arrow'. It is used when assigning actual parameters to formal parameters. for example:

Ada.Text_Io.Put (Item => Initial1);

23) ***With the giant ADA program, how does it know that you only want ADA written and not write A and D when you put the instructions in for how to do that above. Is it because those lines are indented***? (1 student)
It had nothing to do with the indention of the lines. Indention, spaces, empty lines are usually just there to make the code more readable. The statements between 'begin' and 'end Giant_Ada_2;' were the once that called the subroutines 'Draw_Giant_A' and 'Draw_Giant_D'. When the instruction 'Draw_Giant_A;' is executed, the lines between 'begin' and 'end Draw_Giant_A;' will be executed and then printing a A on the screen.

24) ***Why do we use AdaGIDE instead of GNAT***? (1 student)
AdaGIDE is only a GUI (graphical user interface). AdaGIDE uses Gnat for 'everything'.

25) ***If you are using 2 packages, call them ada.foo_io and ada.bar_io, and you Almost never use ada.bar_io, can y ousay use ada.foo_io to save typing, but still write out ada.bar_io.command the few times you need it?***? (1 student)
Yes. You got it perfectly right!!

26) ***How different can using the package get and how different is it than using statements in C++***? (1 student)
I do not understand this question ...

27) ***"No mud"*** (14 students)
Good :)