Lecture C16: Records

Response to 'Muddiest Part of the Lecture Cards'

(38 respondents)

1) ***In defining record 'My_First_Record' you just said ("Name", "Age", "Location") Is that acceptable? Shoouldn't you define each input separately?*** (1 student)
I assume it is the code for Concept question 1 you are refering to. There is no input operations at all in that file. The record is declared at the top, where each field in the record gets its name and what type each field should be.

type My_Record_1 **is record**
Name : String (1..10);
Age : Integer;
Location : String (1..10);
end record;

After those declarations we assigned 2 records and give them their values:

My_First_Record : My_Record_1 := ("John Doe ", 25,"Detroit Mi");
My_Second_Record : My_Record_2 := ("Jane Doe ", 30,"New York ");

Which meens that "My_First_Record" is a record of type 'My_Record_1' and at the same time gets assigned values for each field.

Within the main section of the procedure we then perform operations on these two reocrds, My_First_Record and My_Second_Record.

2) ***In your code, you often use Put("Blah"), but when I've tried this command, I get a compilation error. I have to use Ada.Text_Io.Put("Blah"); Why?*** (1 student)
That is correct. You will always have to use the correct IO routines for the kind of types you want to read/print. I very often exclude the lines e.g, "With Ada.Text_IO; use Ada.Text_IO;" in my examples to save space and time.

3) ***How can we make fields within fields?*** (1 student)
Take a look at the Ada file "complex_persons.adb" that was distributed in class. It defines a record with in a record. That is, the field "Name" is in it self a record with 3 fields for "Title, First name, and Surname".

```
--variable declarations

type Inner_Record is record

          Field_1 : Integer;

          Field_2 : Character;

end record;



type Outer_Record is record

          Field_A : Integer;

          Inner_Record_Name : Inner_Record;

        end record;



Outer_Record_Name : Outer_Record;



-- main program

Outer_Record_Name.Inner_Record_Name.Field_In_Inner_Record
```

The '.' Operator allows the programmer to access fields of the record. The same operator used in succession can be used to access fields of inner records.

4) ***What is the point of using arrays, how are they different from records?*** and similar questions (4 students)

Arrays are used to store homogenous data (data of the same type). A record on the other hand, can have data of different types as its fields. You can, for instance, have:

- an array of records

- a record containing an array as a field

5) ***When someone enters info into a record, do you have to previously declare what record name it goes under (like A_Person.____) What if you dont know how many people you want to enter?*** (1 student)

If you don’t know the number of records you want to enter, there are two possible approaches:

a. Declare an array of records (static allocation)
b. Use dynamic memory allocation

The size of the array is the largest number of records that you might want. This approach however wastes a lot of memory. An alternative approach is to use dynamic memory allocation, which we will cover next semester.

6) ***how do we sort records or find records based on a single field?*** (1 student)

Instead of checking for the element value, check the field value. For example:

```
-- variable declaration

        type My_Aircraft is record

          Id : Integer;

          Latitude : Float;

          Longitude : Float;

        end record;
```

```
-- sorting procedure

        for I in 1 .. Num_Of_Aircraft - 1 loop

            for J in I+1 .. Num_Of_Aircraft loop

                      if Input_Array(I).Latitude > Input_Array(J).Latitude then

                                Temp:= Input_Array(I);

                                Input_Array(I) := Input_Array(J);

                                Input_Array(J) := Temp;

                        end if;

          end loop;

        end loop;
```

7) ***Why are two records with the same data of different types?*** (1 student)
As far as the compiler is concerned, they are defined as different types and can have different operations associated with them.

8) ***What does "(others => space);" do?*** (1 students)
The others=> space operations is called the ‘aggregation’ operation. It fills the remaining fields with blank spaces.

9) ***Is there a way to display the entire record by a single command?*** (1 student)
You can write a procedure to display each of the fields and call the procedure. Other than that, there is no predefined Ada library function that will allow you to display all the elements in a record.

10) ***Is there a way to make actual tables in Ada? Without teadious hand formating, like with borders and stuff?*** (1 students)

11) ***What is the purpose of Nchar in the second code example?*** (1 student)
Nchar keeps track of how many characters have been entered as input. That information is never used in the code, so it should have been removed.

12) ***How can I get an input from the user and stay on the same line? that is, I want to print something on the same line as something I get?*** (1 student)
Move the Skip_Line command to the line after you display your outputs. This will allow you to display

output on the same line.

13) ***Give me pointers, we are so close now!*** (1 student)
Yes we are... and pointer will be covered in beginning of spring term.

14) ***Handling files, making columns and screen formating?*** and similar questions (4 students)
Will be covered more next term.

15) ***"No mud"*** (15 students)
Good :-)