Lecture C2: Ada: Characters and Strings

Response to 'Muddiest Part of the Lecture Cards'

(10 respondents)

1) *How do you know how many bits make up a piece of data?* (1 student)
The number of bits is a function of its type. The Type'First and Type'Last, tell you the range of values that the data type contains. From that, you should be able to compute the number of bits used.

2) *Does* `Question'Last` *give the last used position in the string array or just the last position possible? If it gives just the last position possible what is the point of having both* `Question'last` *and* `Question'Length`? (1 student)

Attributes for array types

`T'first`	gives first index value for array type T
`T'last`	gives last index value for array type T
`T'length`	gives number if components in array type T
`T'range`	gives the index interval for array type T

Where T is the name of a constrained array type. Instead of a type name, the name of an object of an array type can be used:

`A'first`	gives first index value for array A
`A'last`	gives last index value for array A
`A'length`	gives number of components in array A
`A'range`	gives the index interval for array A

First and Last are used to find the first and last index numbers in an array, they can for example be used to make a loop run through all the index values for an array.

```
for I in Question'First .. Question'Last loop
   null;
end loop;
```

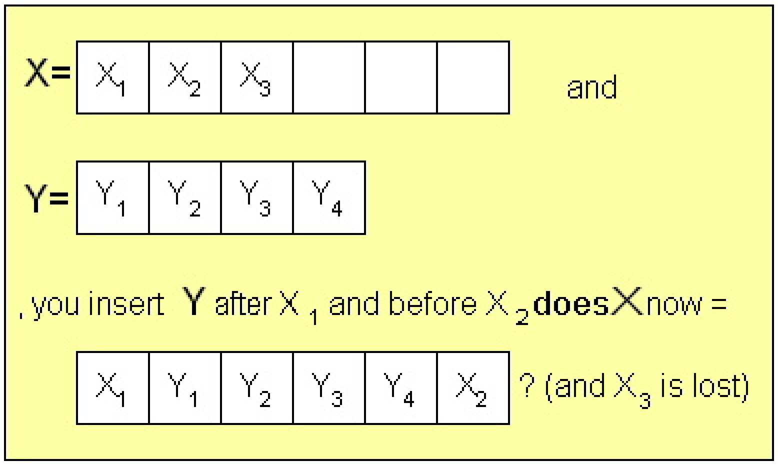
A more elegant solution is to use Range. The loop parameter will run through all the index values for the array.

```
for I in Question'range loop
   null;
end loop;
```

The attribute Length is used to find the number of components in an array.

```
   Put ("Number of elements in the array are: ");
   Put(Question'Last);
```

3) *If:*


X=
X_1 X_2 X_3
and
Y=
Y_1 Y_2 Y_3 Y_4
, you insert Y after X_1 and before X_2 does X now =
X_1 Y_1 Y_2 Y_3 Y_4 X_2
? (and X_3 is lost)

The result of the operation is dependent on how you implement the insertion operation. In the example shown above, the insertion operation may generate a run-time error because the size of array X is of static size 6 and including 4 new elements to an array of size 6 that already has 3 elements. An alternative implementation of the insertion function may just ignore X_3, resulting in the final array appearing as shown in figure above.

4)
```
      with Ada.Text_Io;
      use Ada.Text_Io;

      procedure Cp_Class_2 is
         Quest1 : String(1..17) := "I had to wake up ";
         Quest2 : String(1..16) := "at 8:00am today!";
      begin
         Put(Quest1&Quest2);
         Put();
      end;
```
Except that unfortunate “`Put();`” which generates a compilation error (missing operand), this piece of code looks perfectly fine to me!

5) *<> this says string char with no range? Inserting a string → how does this work?*
The Array types we have studied so far has been constrained arrays, that is, the index constraints are specified in the type declaration.

In Ada we can also declare **unconstrained array types**. That means, when the array is declared, the index type is specified but there is no need to state limits for the index, instead the symbol < > is used.

When an object of an unconstrained array type is declared, the index constraints must be stated.

```
N : Integer := 100;
type Vector is array (Integer range <>) of Character;

Vector_1 : Vector (1 .. 10);
Vector_3 : Vector (1 .. N);


type Index_Type is range 1 .. 100;
type Another_Vector is array (Index_Type range <>) of Integer;
I : Index_Type := 2;

Another_Vector_1 : Another_Vector (40 .. 50);
Another_Vector_2 : Another_Vector (I .. I * 5);
Another_Vector_3 : Another_Vector (40..50):=(0,0,0, others => 1);
```

6) *What is ISO_10646 again?*
ISO 10646 is a character set standard.

LRM section 2.1: "The character repertoire for the text of an Ada program consists of the collection of characters called the Basic Multilingual Plane (BMP) of the ISO 10646 Universal Multiple-Octet Coded Character Set, plus a set of format_effectors and, in comments only, a set of other_control_functions; the coded representation for these characters is implementation defined (it need not be a representation defined within ISO-10646-1)."

More information to be found in the **LRM** (Language Reference Manual), sections **2.1** and **3.5.2**.

7) *Does Ada provide a method for dynamically handling strings? That is, a string of varying length like in Java.* (1 student)
Absolutely. Take a look at the package Ada.Strings.Unbounded (section A.4.5 in the **LRM**)

8) *For*
`Put_Line(Integer'Image(Integer'Val(Character'Pos('1'))));`
why do we have to use `Integer'Val`?

In this case, there is no need to use Integer'Val, because the *position number* of an integer value is equal to the value (section 3.5.4 in the LRM)

9) *For* `function Is_Character (Item : in Wide_Charcter ) return Boolean;`
if the letter is part of the character set, will Boolean return true? Or does the character have to be only in the Wide_Character for Boolean to be true?

Boolean is the type of the returned value. The function will return true if the position of the element in wide character set is between 0 .. 255 (which is the positions present in the character set)

10) No mud (2 persons)