# Section 02 - Subprograms Subprograms 01

Pedro Fernando Flores Palmeros

## 1 INTRODUCTION

Subprograms can be renamed by using the **renames** keyword and declaring a new name for a subprogram:

Code 1: renames syntax

```ada
procedure New_Proc renames Original_Proc;
```

ADA is known as safety-focused language. There are many ways this is realized but two important points are:

- Ada makes the user specify as much as possible about the behavior expected for the program, so that the compiler can warn or reject if there is an inconsistency.
- Ada provides a variety of techniques for achieve the two designs goals above. A subprogram parameter can be specified a mode, which is one of the following:
  - **in** Parameter can only be read, not wri´tten.
  - **out** Parameter can be written to, then read.
  - **in out** Parameter can be both read and written.

The default mode for parameters is **in**; so far, most of the examples have been suing **in** parameters.

In the Code.(2) is executed the case in which a procedure named `In_Out_Params` has two arguments, both of them are of type **in out**. Observer that **it is a procedure, hence it should**

**not use return, but it is modifying the variables and setting them as outputs**. It might be like using references like in C/C++.

Code 2: main.adb

```ada
with Ada.Text_IO; use Ada.Text_IO;

procedure In_Out_Paramters is
   procedure Swap(A, B: in out Integer) is
      Tmp : Integer;
   begin
      Tmp := A;
      A   := B;
      B   := Tmp;
   end Swap;
   A : Integer := 12;
   B : Integer := 44;
begin
   Swap(A,B);

   -- prints 44
   Put_Line(Integer'Image(A));
   end In_Out_Paramters;
```

An **in out** parameter will allow read and write access to the object passed as parameter, so in the example above, we can see that A is modified after the call to Swap.