Your First Ada Program

Presented by Quentin Ochem

University.adacore.com

```ada
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Main subprogram name (can be any Ada identifier)

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

End of main name (optional)

Variables declaration, only before begin

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Statements, only between begin … end

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

The Ada assignment is :=

The Ada equality operator is =

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

' introduces a special property, called attribute

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Value is an attribute transforming a String to a value of a type

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Image is an attribute transforming a value of a type to a String

with allow to use a library unit, here Ada.Text_IO for textual functions

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Get_Line reads a line on the command line

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

Put_Line prints text on the command line

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

& is the concatenation operator, used between String values

if … then delimits a decision, no need for parentheses

```
with Ada.Text_IO;

procedure Hello is
   A, B, C : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);
   C := A + B;

   if C = 0 then
      Ada.Text_IO.Put_Line ("RESULT IS 0");
   elsif C > 0 then
      Ada.Text_IO.Put_Line ("POSITIVE RESULT :" & Integer'Image (C));
   else
      Ada.Text_IO.Put_Line ("NEGATIVE RESULT :" & Integer'Image (C));
   end if;
end Hello;
```

elsif introduces an alternative decision

Quiz

Identify the Errors

```
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;

   A = Integer'Image (Ada.Text_IO.Get_Line);
   B = Integer'Image (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

```ada
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;

   A = Integer'Image (Ada.Text_IO.Get_Line);
   B = Integer'Image (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

“begin” is needed to introduce a sequence of statements

```
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;

   A = Integer'Image (Ada.Text_IO.Get_Line);
   B = Integer'Image (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

```
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;
begin
   A = Integer'Image (Ada.Text_IO.Get_Line);
   B = Integer'Image (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

The Ada assignment instruction is :=

Image converts a number into a string,
Value would convert a string to a number

```
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;
begin
   A := Integer'Image (Ada.Text_IO.Get_Line);
   B := Integer'Image (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

```ada
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);

   if A == B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

The Ada equality operator is =

```
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);

   if A = B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & A);
   end if;
end Hello;
```

A is not a String, need to be converted through Integer'Image (A)

```ada
with Ada.Text_IO;

procedure Hello is
   A, B : Integer;
begin
   A := Integer'Value (Ada.Text_IO.Get_Line);
   B := Integer'Value (Ada.Text_IO.Get_Line);

   if A = B then
      Ada.Text_IO.Put_Line ("A EQUALS B, VALUE IS " & Integer'Image (A));
   end if;
end Hello;
```

university.adacore.com