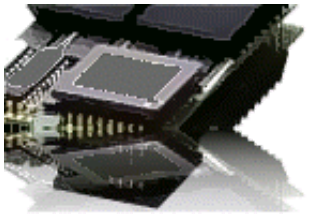

Brock University
Department of Computer Science

COSC 3P93: A0 -- Ada Warm-Up

Instructor: **Vlad Wojcik**

ASSIGNMENT OBJECTIVE:

To familiarize students with Ada 2005 programming techniques. This assignment will not be marked, and there is no due date/time. However, the material and code of this assignment will be useful in future assignments.

THE PROBLEM:

The file **random.pas**, reproduced below for your convenience, contains the code of an industrial-strength random number generator, written in Pascal, a direct ancestor of Ada.

```
PROGRAM Random(Output);

 CONST seed1 = 5; seed2 = 10000; seed3 = 3000;

 VAR x, y, z : integer;
     loop : integer;

 FUNCTION   Unif : real;
     Var tmp : real;
     Begin {Unif}
        x := 171*(x mod 177) - 2*(x div 177);
        if x<0 then x := x + 30269;
        y := 172*(y mod 176) -35*(y div 176);
        if y<0 then y := y + 30307;
        z := 170*(z mod 178) -63*(z div 178);
        if z<0 then z := z + 30323;
        tmp :=  x/30269.0 + y/30307.0 + z/30323.0;
        Unif := tmp - trunc(tmp)
     End;  {Unif}

 BEGIN
     x := seed1; y := seed2; z := seed3;
     for loop := 1 to 1000 do
     writeln(loop:4, ' ==> ', unif:7:5)
```

```
END.
```

The function Unif, being the random number generator, returns numbers seemingly of random nature, uniformly distributed within an interval [0, 1]. The value returned by Unif depends in a somewhat arcane way on three integer values x, y, z. These three values are declared outside the block defining the function Unif, and therefore retain their values between Unif calls. In this way Unif produces seemingly random, but reproducible number sequences.

At the very beginning of the program execution, x, y, z are given some seed values. Different seed values result in different, but reproducible, number sequences. This reproducibility is useful for debugging purposes; also, science demands that the results of every scientific experiment be reproducible.

Obtaining non-reproducible sequences is simple: the seed values should be established by perusing the system clock (accessible to Ada 2005 programmers via the package CALENDAR), prior to calling UNIF for the first time.

You are to create an Ada 2005 package, called RANDOM, that would contain an equivalent random number generator, also called UNIF. The users of the package should be able to call UNIF, and also should be able to set the seed values if they wanted reproducible sequences. Let us adopt the convention that not passing any seed values before calling UNIF amounts to the request for a non-reproducible number sequence.

The three values x, y, z should be hidden in the package body, following the canons of proper software engineering.

NOTE: The function trunc in Pascal takes a real argument and returns a real result, being the value of its argument with the fractional part truncated. {Curly braces like this contain comments}.

Instructor: Vlad Wojcik

Revised: 30 September, 2014 4:25 PM