EXAMPLE PROGRAM SINE COSINE TANGENT REPORT

1. Requirements

Produce an Ada program that will return the sine, cosine and tangent of a given angle.

2. Design

A top down analysis is presented in Table 1.

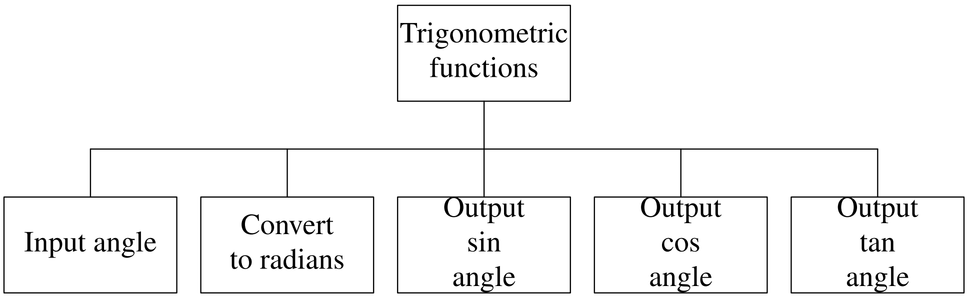

Table 1: *Top down analysis*

A single procedure:

1) `LOWER_2_UPPER` (top level procedure) - input angle; convert to radians; output sine, cosine and tangent+.

NAME	DESCRIPTION	TYPE	RANGE
INPUT_ANGLE	Input variable	ANGLE	0.0..90.0
DEG_2_RAD	Global constant	FLOAT	PI/180.0

where ANGLE is a subtype of the type FLOAT.

An appropriate Nassi-Shneiderman design is presented in Table 2. The design includes a single input variable:

Input angle INPUT_ANGLE
Convert to radians
Output sine INPUT_ANGLE
Output cosine INPUT_ANGLE
Output tangent INPUT_ANGLE

Table 2: *Nassi-Shneiderman Chart*

3. Implementation

Implementation as shown in Table 3:

```
-- SINE COSINE TANGENT
-- 5 August 1997
-- Frans Coenen
-- Dept Computer Science, University of Liverpool

with TEXT_IO, MATH_CONSTANTS, GENERIC_ELEMENTARY_FUNCTIONS;
use TEXT_IO, MATH_CONSTANTS;

procedure SIN_COS_TAN is
        subtype ANGLE is FLOAT range 0.0..90.0;
        package FLOAT_INOUT is new FLOAT_IO(FLOAT);
        use FLOAT_INOUT;
        package MATH_FUNC is new GENERIC_ELEMENTARY_FUNCTIONS(FLOAT);
        use MATH_FUNC;
        DEG_2_RAD   : CONSTANT FLOAT:= PI/180.0;
        INPUT_ANGLE : ANGLE;
begin
-- Input opposite and adjacent side
        PUT_LINE("Input angle: ");
        GET(INPUT_ANGLE);
        INPUT_ANGLE:= A*DEG_2_RAD;
-- Sin A
        PUT("sin is: ");
        PUT(sin(INPUT_ANGLE),FORE=>1,AFT=>5,EXP=>0);
        NEW_LINE;
-- Cos A
        PUT("cos is: ");
        PUT(cos(INPUT_ANGLE),FORE=>1,AFT=>5,EXP=>0);
        NEW_LINE;
-- Tan A
        PUT("tan is: ");
        PUT(tan(INPUT_ANGLE),FORE=>1,AFT=>5,EXP=>0);
        NEW_LINE;
end SIN_COS_TAN;
```

Table 3: *Implementation*

4. Testing

Use BVA and limit testing to verify input. This will also serve to check the operation of the trigonometric statements (with the inclusion of a mid range value, i.e. 45.0 in this case). A suitable set of test cases are presented in Table 4.

TEST CASE	EXPECTED RESULT		
INPUT_ANGLE	Sine	Cosine	Tangent
-0.1	CONSTRAINT ERROR		
0.0	0.00000	1.00000	0.00000
0.1	0.00175	1.00000	0.00175
45.0	0.70711	0.70711	1.00000
89.9	1.00000	0.00175	572.95721
90.0	1.00000	0.00000	CONSTRAINT ERROR
90.1	CONSTRAINT ERROR		

Table 4: *BVA, limit and arithmetic test cases*

A screen dump illustrating the output produced when the above test cases are run is given in Table 5.

```
kuban-323 $ sin_cos_tan
Input angle:
-0.1
Ada-runtime: Exception CONSTRAINT_ERROR raised in
/tmp_mnt/home/staff2/ra/frans/public_html/2CS21/week5/sin_cos_tan.ada
on line 15. Floating rangecheck error: 0 not in range 0..0.

kuban-324 $ sin_cos_tan
Input angle:
0.0
sin is: 0.00000
cos is: 1.00000
tan is: 0.00000

kuban-325 $ sin_cos_tan
Input angle:
0.1
sin is: 0.00175
cos is: 1.00000
tan is: 0.00175

kuban-326 $ sin_cos_tan
Input angle:
45.0
sin is: 0.70711
cos is: 0.70711
tan is: 1.00000

kuban-327 $ sin_cos_tan
Input angle:
89.9
sin is: 1.00000
cos is: 0.00175
tan is: 572.95721
```

Table 5: *Test case output*

```
kuban-328 $ sin_cos_tan
Input angle:
90.0
sin is: 1.00000
cos is: 0.00000
tan is:
Ada-runtime: Exception CONSTRAINT_ERROR propagated out of main.

kuban-329 $ sin_cos_tan
Input angle:
90.1
Ada-runtime: Exception CONSTRAINT_ERROR raised in
/tmp_mnt/home/staff2/ra/frans/public_html/2CS21/week5/sin_cos_tan.ada
on line 15. Floating rangecheck error: 0 not in range 0..0.
```

Table 5 continued

For completeness we should also carry out some data validation testing (Table 6).

```
kuban-232 $ sin_cos_tan
Input angle:
frans
Ada-runtime: Exception FLOAT_INPUT_ERROR_DIGIT_NOT_READ
propagated out of main.

kuban-233 $ rans
ksh: rans:  not found

kuban-235 $ sin_cos_tan
Input angle:
1e.2
Ada-runtime: Exception FLOAT_INPUT_ERROR_PERIOD_NOT_READ
propagated out of main.

kuban-236 $ .2
ksh: .2:  not found
```

Table 6: *Data validation test results.*