SE 2A04 Fall 2002

Lab Exercise 2

Instructor: William M. Farmer

Assigned:	27 September 2002
Demonstration due:	4 October 2002
Lab report due:	11 October 2002

The purpose of this lab exercise is to implement a simple abstract data structure as a module in two different ways, and then test both implementations with the same test module.

The following Oberon-2 interface is for a module that implements a single abstract two-dimensional vector in the real plane:

```
INTERFACE Vector;
  XCoordinate(): REAL;
    (* Returns the x-coordinate of the vector. *)
  YCoordinate(): REAL;
    (* Returns the y-coordinate of the vector. *)
  Magnitude(): REAL;
    (* Returns the magnitude of the vector. *)
  Angle(): REAL;
    (* Returns the angle from the x-axis to the vector measured
       in radians. *)
  SetXCoordinate(r: REAL);
    (* Changes the x-coordinate of the vector to r without
       changing the y-coordinate. *)
  SetYCoordinate(r: REAL);
    (* Changes the y-coordinate of the vector to r without
       changing the x-coordinate. *)
  SetMagnitude(r: REAL);
    (* Changes the magnitude of the vector to r without
       changing the angle. *)
  SetAngle(r: REAL);
    (* Changes the angle of the vector to r without
       changing the magnitude. *)
  Mul(r: REAL);
    (* Multiples the vector by r. *)
 END Vector.
```

Step 1

Write an Oberon-2 module named `TestVector` that "black box" tests any module named `Vector` implementing the module interface specified above.

Step 2

Write an Oberon-2 module named `Vector` that implements the module interface by representing the vector as a pair of cartesian coordinates. Write another Oberon-2 module named `Vector` that implements the module interface by representing the vector as a pair of polar coordinates. (Keep the two module files named `Vector.m` in separate directories.)

Step 3

Construct an Oberon-2 program named `TestCartesian` that uses `TestVector` to test the `Vector` module that uses cartesian coordinates, and construct another Oberon-2 program named `TestPolar` that uses `TestVector` to test the `Vector` module that uses polar coordinates.

Step 4

During the lab session on October 4, demonstrate both the `TestCartesian` and `TestPolar` programs.

Step 5

Before or during the lab session on October 4, send a copy of both of your `Vector` modules to your receiver partner, and get a copy of both of your sender partner's `Vector` modules. Construct Oberon-2 programs using `TestVector` to test both of your sender partner's programs.

Step 6

Write a lab report that includes the following:

1. A copy of the Lab Exercise 2 Marking Scheme (which is be available on the course Web site) stapled to the front of your report.

2. A copy of your `TestVector` module and a brief explanation of its design.

3. A copy of both of your `Vector` modules and a brief explanation of their design.

4. The results of the test of your `Vector` modules.

5. The results of the test of your sender partner's `Vector` modules.

6. A discussion of the test results and what you learned doing the lab exercise.

7. A discussion of any problems you found with the specification of the `Vector` interface.

8. A copy of the part of your log book relevant to this lab exercise.

The lab report is due no later than the beginning of the tutorial session on October 11.