# §10.5-10.7: Local Modules

•devo

9 Nov 2005
CMPT14x
Dr. Sean Ho
Trinity Western University

**Reminders:**

- journals in folder
- Hw (ch9 #30) due
- Quiz ch9 today

# Quiz ch9: 4 questions, 20 marks, 10 minutes

```
TYPE mySet = SET OF [0 .. 10];
VAR a, b : mySet;
a := mySet {1, 2, 9, 10};
b := mySet {2, 4, 6, 8, 10};
```

- Evaluate these two expressions: a*b, a/b
- Create a list of 100 points: each point has 3 REAL coordinates (x,y,z) and 3 CHAR colors (r,g,b)
  - Be sure to declare any types you may need
- How would you determine how many LOCs are used to store the above list of points?
- Name the 3 standard I/O libraries used to open/close files, and the differences among them
  - Hint: they contain e.g., Open, OpenRead, OpenOld

# Quiz ch9 answers: #1

```
TYPE mySet = SET OF [0 .. 10];
VAR a, b : mySet;
a := mySet {1, 2, 9, 10};
b := mySet {2, 4, 6, 8, 10};
```

- a*b = [3]
  - mySet { 2, 10 }  (set intersection AND)
- a/b = [3]
  - mySet { 1, 4, 6, 8, 9 } (symmetric set difference XOR)

- A side note on INCL/EXCL:
  - INCL (a, 5) <----> a := a + mySet {5};
  - Second parameter is an element, not a set

# Quiz ch9 answers: #2-3

- Create a list of 100 points: [6]

```
TYPE
    Point =
        RECORD
            x, y, z : REAL;
            r, g, b : CHAR;
        END;
VAR
    pointList : ARRAY [0 .. 99] OF Point;
```

- Storage size of pointList: [2]

```
SIZE (pointList)
```

- How would you initialize pointList? [+2]

# Quiz ch9 answers: #4

- Name the 3 standard I/O libraries used to open/close files, and the differences among them [6]
  - StreamFile: restricted streams
  - SeqFile: rewindable streams
  - RndFile: random-access files

# Programming as communication

- Computing scientists are toolsmiths:
  - We exist to serve/help the user
  - Hence good people skills are essential!
- Programming is:
  - not only communicating to the computer
  - it's also communicating to people:
    - End user
    - Other programmers reading your code
    - Yourself (reading your code at a later date)
- => Express yourself logically and clearly

# Essay / Paper

- Computing scientist as **Godly Christian Leader**:
  - Not just knowledge about tools, but
  - Wisdom of how to use tools
    - To serve others and
    - To give glory to God
- Write a short essay on a topic of your choosing about computers and society:
    - Approx 5 pages typed double-spaced 12pt 1in margins
    - Submit half-page topic by next Wed 16Nov
    - Paper due last day of class (7Dec)
      - **Electronic submission ok (email, eCourses)**

# Sample paper topics

- Censorship and free speech
  - Pornography, gambling, hate groups, etc.
- Violence in video games (Columbine etc.)
- Privacy: online banking, ID theft, etc.
- Blogs: effect on politics, social interaction, etc.
- File sharing: Napster, Gnutella, etc.
- Artificial intelligence: the nature of sentience
- Online dating (e.g. eHarmony): pros/cons
- Equity of access / rural digital divide
- ..... come up with your own topic!

# Tips for essay writing

- Your essay should be a position paper:
  - The topic should have at least two sides (e.g. pro/con)
  - You should state (in the introductory paragraph) what your position is (thesis)
  - You should have at least 2-3 points, each, both for and against your position
    - It is not necessary to rebut every point that contradicts your position:
    - Be honest about the faults/limitations of your thesis
  - Summary intro/conclusion paragraphs
  - Proper English (spelling, grammar) is important!

# Review of last time (9.11-10.4)

- RndFile: random-access files
  - OpenOld/OpenClean, NewPos/SetPos
- Scope, visibility, blocks
- Rules of thumb about variables/parameters
- Procedure variables

# What's on for today (10.5-10.7)

- Local modules
- Import and export of items from modules
- Qualified export

# Local modules

- Modules can contain:
  - IMPORTs
  - VAR/TYPE declarations
  - PROCEDURE declarations
  - A body (BEGIN ... END)
    - Also called the module's initialization section (c.f. library IMP-lementation module body)
  - Other modules!
- A local module is a construct for encapsulation:
  - Defines a scope of visibility for items

# Local module example

```
MODULE Parent;
VAR
    pVar1, pVar2 : REAL;

    MODULE Child;
        IMPORT pVar1;
        EXPORT cVar1;
        VAR
            cVar1, cVar2 : REAL;
    END Child;

BEGIN
END Parent.
```

Visible:
pVar1, pVar2
cVar1

Visible:
pVar1,
cVar1, cVar2

All vars exist and keep value throughout Parent module, even when not visible!

# Qualified export

- Cannot export two variables of same name to same scope of visibility:

```
MODULE Parent;
VAR myVar : REAL;
MODULE Child;
    EXPORT myVar;          (* bad!  (compile error) *)
    VAR myVar : REAL;
END Child;
```

- Solution: use qualified export:

```
EXPORT QUALIFIED myVar;    (* inside Child module *)
```

- Parent module is forced to use qualified form:

```
Child.myVar := 5.0;
```

# Example: sibling modules

```
MODULE Parent;

    MODULE Child1;
        EXPORT c1var;
        VAR c1var: REAL;
    END Child1;

    MODULE Child2;
        IMPORT Child1;
        VAR c2var: REAL;
    END Child2;

END Parent.
```

Visible in Child1:
c1var

Visible in Child2:
c1var (aka Child1.c1var),
c2var

Visible in Parent:
c1var (aka Child1.c1var)

# Summary of today (10.5-10.7)

- Local modules
- Import and export of items from modules
- Qualified export

- The book covers more details in 10.6-10.7; please read them on your own:
  - Dynamic modules
  - Opaque types
  - Make sure you understand the Fibonacci example

# TODO items

- Fall break rest of this week
- Lab 7 due next week:
  - 9.14 #(37 + 38) / (40 + 41)
- Reading: through §10.12 for Mon
- Paper topic due next Wed
- HW due next Fri: 9.14 #30
- Quiz ch10 next Fri
- Midterm ch8-10 Wed 23Nov (in 2 weeks)