SWEN421 – Lecture 1
Introduction to Ada

Note:

- These slides are intended as an outline/starting point, introducing key ideas, not as comprehensive notes
- You will need to read further to understand the details
- This is **not** a course in Ada programming – you are expected to learn most of the details of Ada yourself
- The course is about formal methods (program specification and verification), which happens to use Ada as a vehicle for exploring these ideas

What is Ada?

- Designed for US Department of Defence (DoD) to reduce cost of developing “comand and control” applications (specifically to replace a large number of languages then being used and eliminate the cost of developing/maintaining their implemenations)
- First version released in 1980, following a competitive development process, and standardised in 1983.
- Revised in 1995, 2005 and 2012 (search Ada Comparison Chart)
- Always intended for developing large scale, high integrity software
- Ada 2012 added many features to aid in testing and verification
- SPARK Ada restricts Ada to allow further verification

Main features of Ada

Ada belongs to the Algol/Pascal/Modula family:

- Uses word delimiters for compound statements (**loop** - **end loop**, **if** – **end if**, …) rather than curly brackets as in C-based languages.
- Uses **:=** for assignment and **=** for equals (not **=** and **==**).
- Declarations separated from statements.
- Variables/parameters declared as **var : type** (not **type var**).
- Distinguishes between **procedure** and **function** subprograms:
  - functions return values and are called in expressions,
  - procedures don't return values and are called as a statement.
- Very rich, strong type system – more later.
- Concurrency was there from the beginning.
- OO features added later.

Ada program structure

- An Ada program is a collection of packages – where a **package** is a kind of module structure, which is a collection of declarations for types, variables and subprograms
- Can control what names are exported by the package, and what names are imported from other packages

Ada Control Structures: Conditionals

- **if** condition **then** statements **end if**
- **if** condition **then** statements **else** statements **end if**
- **if** condition **then** statements **elsif** condition **then** …
- **case val is**

```
  when 0 =>
    Put_Line("Zero");
  when 1 .. 0 =>
    Put_Line("Poisitive digit");
  when 10 | 12 | 14 | 16 | 18 =>
    Put_Line("Even in tens");
  when others =>
    Put_Line("Something else");
end case;
```

Ada control structures: loops

- **while** x < y **loop** x := x+1; y := y-1; **end loop**;
- **for** k **in** lo .. hi **loop** A(k) := 0; **end loop**;
- **loop** x := A(i) + A(i+1); **exit when** x = 0; I := i+1; **end loop**;

Ada procedures

```
with Sqrt, Simple_IO;  use Simple_IO;
procedure Print_Root is
  x: Float;
begin
  Get(x);
  Put(Sqrt(x));
end Print_Root;
```

Ada functions

```
function Sum(x, y: Float) return Float is
  s: Float;
begin
  s := x+y;
  return s;
end Sum;
```

Ada parameter passing

```
procedure Sum(x, y: in Float; s: out Float) is
begin
  s := x+y;
end Sum;

procedure Add(x: in out Float; y: in Float) is
begin
  x := x+y;
end Add;
```