Programming in the Small

- Aim: to remind you how to read/write programs in Ada 95, Java and ANSI C
- Practicals will be in Ada 95
- For exams, you should be able to sketch solutions to problems in the three languages. However, we will not expect programs to be syntactically correct!
- We will expect a higher literacy in Ada 95

- We will also consider aspect of Modula-1 (for device driving) and occam2 (for its pure message-based IPC facility).

Fortran Example

```
DO 20 I = 1, 100
```

Loop to label 20, iterating I from 1 to 100

On the U.S. Viking Venus Probe, a programmer wrote

```
DO 20 I = 1. 100
```

The compiler interpreted this as an assignment statement and ignored the spaces

```
DO20I = 1.100
```

Variables need not be declared in Fortran, and those beginning with D are assumed to be of type real. 1.100 is a real literal!

An Overview of Ada

An Ada program consists of one or more program units:

- a subprogram (procedure or function) — can be generic
- a package (possibly generic) — used for encapsulation and modularity
- a task — used for concurrency
- a protected unit — a data-oriented synchronisation mechanism

Library units: package, subprogram

Subunits: subprogram, package, task, protected unit

Blocks

Ada is a block-structured language

```
declare
   -- definitions of types, objects,
   -- subprograms etc.
begin
   -- sequence of statements
exception
   -- exception handlers
end;
```

A block can be placed anywhere a statement can be placed

Points about Blocks

- objects declared in a block may only be used in that block (scope)
- any statement in the sequence of statement may itself be a block
- exception handlers can be used to trap errors arising out of the execution of the sequence of statements (they may be omitted)

Example

```
function Largest(X : Vector) return Integer is
  Max : Integer := 0;
begin
  for I in X'Range loop
    if X(I) > Max then
      Max := X(I);
    end if;
  end loop;
  return Max;
end Largest;
```

'Range is an **attribute** and = 'First - 'Last

What is wrong with this example?

The C Language

- Is sequential
- Main structuring units are functions (although files can be used to aid separate compilation)
- Block structured (called compound statements)

```
{
  < declarative part >

  < sequence of statement >
}
```

- Declarative part cannot contain functions
- Sequence of statements can contain compound statement

C Example

```
int largest(vector X, int len)
{
  int max = 0;
  int i;

  for (i = 0; i < len; i++) {
    // bounds start at 0
    if(X[i] > max) max = X[i];
  }
  return max;
}
```

Assignment statement is =

Equality is = =

Not so type secure as Ada. Typedefs are synonyms for types not derived types. However, the insecurities in C are well known.

An Overview of Java

- A class-based language
- Programming in the small component is similar to C but without explicit pointer values
- More type secure than C
- As with Ada, Java can have exception handler at the end of a block (but only if the block is a try block)
- Functions can only be declared in the context of a class

A Java Example

```
class SomeArrayManipulationClass
{
  public int largest(vector X)
  {
    int max = 0;
    int i;

    for (i = 0; i < X.length; i++) {
      // bounds start at 0
      // length is an instance variable of array objects
      if(X[i] > max) max = X[i];
    }
    return max;
  }
}
```

All arrays are objects

Java and Reference Types

- All objects in Java are represented as reference values
- Comparing two objects will compare their references not their values:

```
Node Ref1 = new Node();
Node Ref2 = new Node();
. . .
if(Ref1 == Ref2) { . . . }
```

  will compare the locations of the objects not their values; it is necessary to implement a `compareTo` method

- A similar situation occurs with object assignment; it is necessary to provide a `clone` method

Discrete Types

Ada	Java	C
Integer	int	int
	short	short
	long	long
	byte	
Boolean	boolean	
Character		char
Wide_Character	char	wchar_t
Enumeration types	*None*	*typedef enum*