Language Standardisation

```
int i; i = (1 && 2) + 3;
```

Is it valid C and what is the value of `i`.

How do we answer such questions?

1. Read the reference manual.
2. Try it and see!
3. Read the ANSI C Standard.

Language Standards

- Proprietry standards
- Consensus standards.
  - ANSI.
  - IEEE.
  - BSI.
  - ISO.

Language Standards

- Timeliness.
- Conformance.
- Obsolescence.
- Ambiguity and freedom to optimise.
- Machine dependence.
- Undefined.
- Deprecated.

Language Standards

What does the following mean?

```
x = y + z;

x = z + y;

x = a + b + c;

x = (a + b) + c;

x = a + (b + c);

a = a + b + c;

b = a + b + c;
```

Language Standards

What does the following mean?

```
x = x + g()
```

```
x = g() + x;
```

```
x = g() + g();
```

Language Standards

What does the following mean?

```
int x=1, y=1;

int g() { return ++y; }

int main() {
  x = ++x + ++x;
  y = g() + g();
  printf("x=%d y=%d\n", x, y);
  return 0;
}
```

Answer:

```
Linux              (gcc):   x=6 y=5
Mips               (gcc):   x=5 y=5
                   (cc):    x=6 y=5
DEC Alpha and Sun4 (gcc):   x=5 y=5
                   (cc):    x=5 y=5
```

Language Standards - PL/1

What does the following mean?

```
9 + 8/3
```

Is it?

1. 11.6666666....
2. Overflow
3. 1.6666666....

What does the following mean?

```
IF (1=1B) THEN ...
```

Language Standards - PL/1

`DEC(p,q)` means `p` digits with `q` are after the decimal point.

Type rules for DECIMAL in PL/1:

```
DEC(p1,q1) + DEC(p2,q2) =>
  DEC(1+MAX(p1-q1,p2-q2)+MAX(q1,q2),MAX(q1,q2))
DEC(p1,q1) / DEC(p2, q2) =>
  DEC(15, 15-((p1-q1)+q2))
```

So, for `9 + 8/3`, we have:

```
     DEC(1,0) + DEC(1,0)/DEC(1,0)
=>   DEC(1,0) + DEC(15, (15-((1-0)+0)))
=>   DEC(1,0) + DEC(15,14)
=>   DEC(1+MAX(1-0,15-14)+MAX(0,14), MAX(0,14))
=>   DEC(15,14)
```

So the calculation is as follows:

```
  9 + 8/3
= 9 + 2.66666666666666
=    11.66666666666666  -- OVERFLOW
=     1.66666666666666  -- OVERFLOW disabled
```

Language Standards - PL/1

Evaluation of: `IF (1=1B) ...`

```
      1   =   1B   -- DEC(1,0) and BIT STRING
=>   0001B = 1000B -- two BIT STRINGs
=>         0       -- FALSE
```