

The D Programming Language

Peter C. Chapin
CIS-3030

Overview

- Like C++ with Java influences + extra
 - C family syntax.
 - Provides low level concepts.
 - Bit manipulation (shifts, bit operators, etc)
 - Pointers
 - Structures (that are value types)
 - Provides high level concepts.
 - Classes (that are garbage collected reference types)
 - Templates and metaprogramming.
 - Nested functions, function literals.
 - Built in associative arrays, etc, etc.

Background

- Created by Walter Bright
 - ... apparently as a reaction against ugly C++.
 - Sacrifice C compatibility...
 - (D compiler can not compile C programs)
 - ... for the sake of a cleaner language.
 - Easily as rich and expressive as C++
 - Probably more so.
- Two versions.
 - D1 is stable; compiler only getting bug fixes.
 - D2 is under development; not entirely compatible.

Sample Date Class

- Method to advance a date by one day.

```
• class Date {  
    private int day_m, month_m, year_m;  
  
    void next()  
    {  
        if (day_m != days_in_month()) ++day_m;  
        else {  
            day_m = 1;  
            if (month_m != 12) ++month_m;  
            else {  
                month_m = 1;  
                ++year_m;  
            }  
        }  
    }  
}
```

Contracts

- Class invariants checked at each method call.

- `class Date {`

- `invariant {`

- `assert(1 <= month_m && month_m <= 12);`

- `assert(1900 <= year_m && year_m <= 2099);`

- `if (month_m != 2)`

- `assert(1 <= day_m &&`

- `day_m <= month_lengths[month_m - 1]);`

- `if (month_m == 2)`

- `assert(1 <= day_m && day_m <= 29);`

- `if (month_m == 2 && day_m == 29)`

- `assert(is_leap());`

- `}`

- `}`

Contracts Continued

- Pre and post conditions

- ```
int f(int x, int y, Date d)
 in {
 assert(y >= 0 && x > y);
 assert(d != null);
 }
 out(result) {
 assert(result > 1);
 }
 body {
 // Normal execution of function f.
 }
```

# Summary

- Interesting language
  - Powerful, reasonably clean, interesting features.
- Active community
  - <https://dlang.org>
  - <https://wiki.dlang.org>
- A bit haphazard
  - Some features don't seem well coordinated.
  - Some features don't seem that useful.
- Worth watching