

Prototypes

Victor Eijkhout, Harika Gurram,
Je'aime Powell, Charley Dey

Fall 2018

Prototypes and forward declarations

A first use of prototypes is *forward declaration*:

```
int f(int);  
int g(int i) { return f(i); }  
int f(int i) { return g(i); }
```

Prototypes for separate compilation

```
// file: def.cxx
int tester(float x) {
    .....
}
```

```
// file : main.cxx
int tester(float);

int main() {
    int t = tester(...);
    return 0;
}
```

Prototypes and header files

```
// file: def.h  
int tester(float);
```

The header file gets included both in the definitions file and the main program:

```
// file: def.cxx  
#include "def.h"  
int tester(float x) {  
    .....  
}
```

```
// file : main.cxx  
#include "def.h"  
  
int main() {  
    int t = tester(...);  
    return 0;  
}
```

Class prototypes

Header file:

```
class something {  
public:  
    double somedo(vector);  
};
```

Implementation file:

```
double something::somedo(vector v) {  
    .... something with v ....  
};
```

Strangely, data members also go in the header file.

Review quiz 1

For each of the following answer: is this a valid function definition or function prototype. Are any of them a constructor?

- `int foo();`
- `int foo() {};`
- `int foo(int) {};`
- `int foo(int bar) {};`
- `int foo(int) { return 0; };`
- `int foo(int bar) { return 0; };`
- `foo();`
- `foo() {};`