Binding

Binding is the process by which

- A routine invocation is attached to code that is executed
- An attribute reference is connected a memory location

Binding can be done

- At compile time — *static binding, using static linking*
- At program load time — *static binding, using dynamic linking*
- At run time — *dynamic binding*

Static binding in C

In the .h (header) file:

```
extern int foo (arg c);
extern float my_variable;
```

In the .c file:

```
int foo (arg c);
float my_variable;
```

Execution of a call in C

On the caller side:

- Push the actual parameters on the stack.
- Push the return address on the stack
- Jump to the code for the callee
  - *Address to jump to is "filled out" during linking*

On callee side:

- Push space for local variables and initialize, as needed
- Execute the statements in the routine's body
- Read the return address from the stack
- Pop local variables, return address and actual params
- Jump to return address