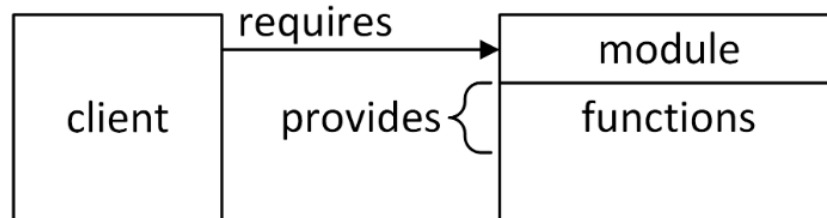


# Tutorial 6

- Modules
- Interfaces
- `stack.h`
- `stack.c`
- Problem: Reverse and Sum

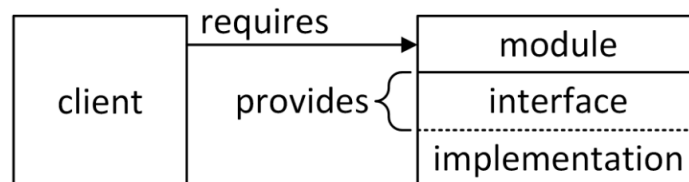
# Modules

- Modules are collections of functions that share a common aspect or purpose
- Modules provide modularity: write complex programs by combining simpler, well-defined programs
  - Re-usability
  - Maintainability
  - Abstraction



# Interface

- Modules are often divided into an **interface** and a **implementation**
- The interface contains a program purpose, function declarations (return type, name, argument types) and function documentation
- The interface is declared in a .h ("header") file and `#included` at the top of a client program to bring module functions into program scope



# stack.h

- Let's write a header file for a `stack` module that will implement an integer stack
- We need to include:
  - Purpose
  - `struct stack` type declaration
  - Function declarations
  - Function documentation

# stack.c

- Let's write an implementation for the `stack.h` interface

# Problem: Reverse and Sum

- Use the `stack.h` header file to write a program that reads `ints` from `stdin`, then prints out each number in reverse, adding each time the sum of all previous numbers