IMPLEMENTING POLYMORPHISM

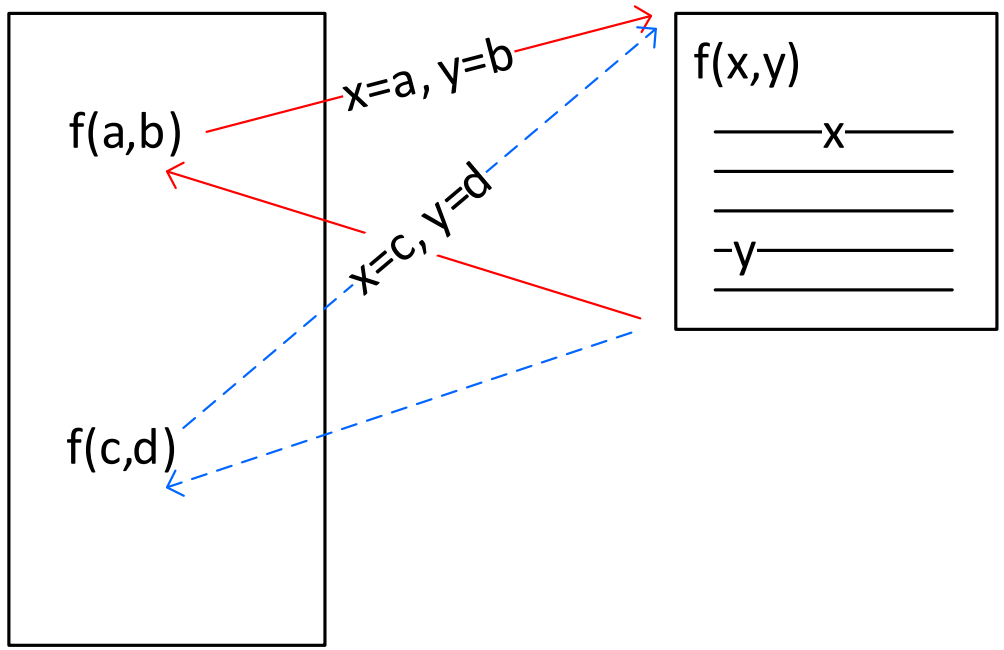
Dynamic, Runtime, or Late Binding

Dynamic Dispatch

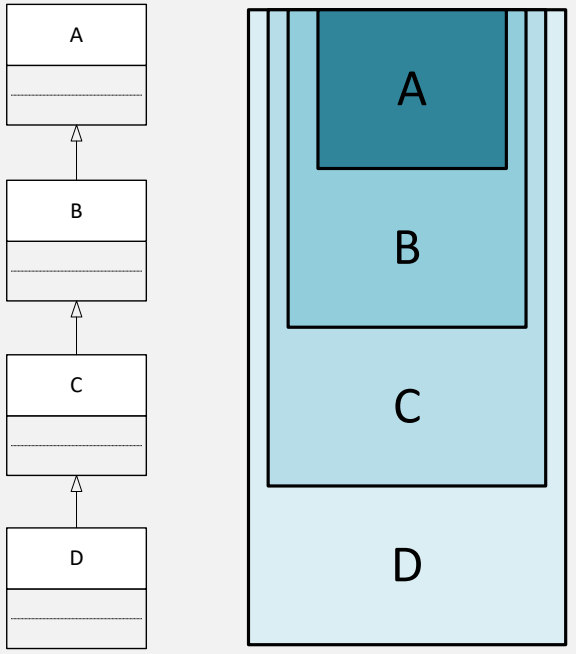
Delroy A. Brinkerhoff

NON-POLYMORPHIC FUNCTION CALLS

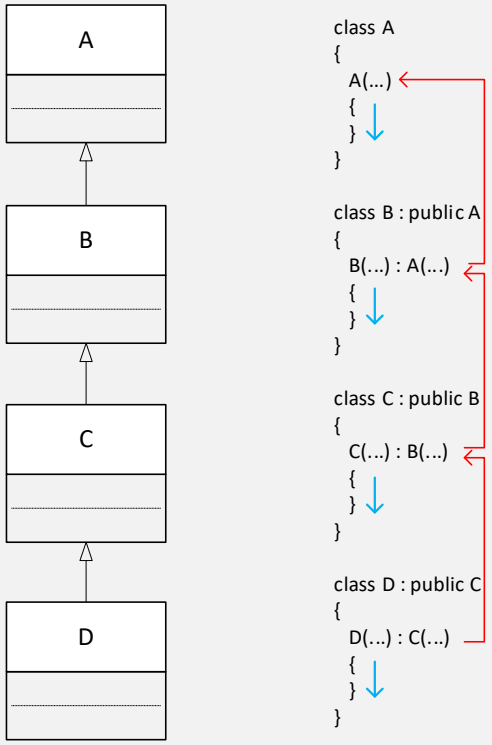
1. Saves the return address
   - the program counter
2. Passes arguments to parameters
3. Jumps to the function's entry point
   - sets the program counter
4. Executes the function's instructions
5. Returns to the address following the call
   - sets the program counter



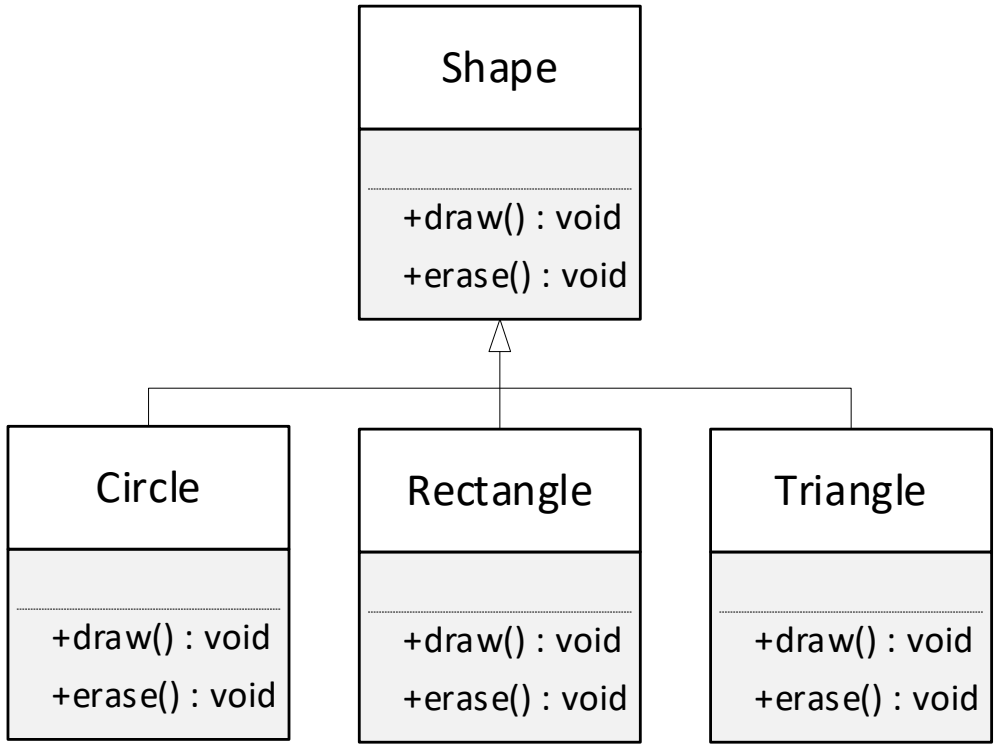
INHERITANCE AND OBJECTS



INHERITANCE AND CONSTRUCTORS




```
class D : public C
{
        void D(…) : C(…)
        {
                ….
        }
}
```



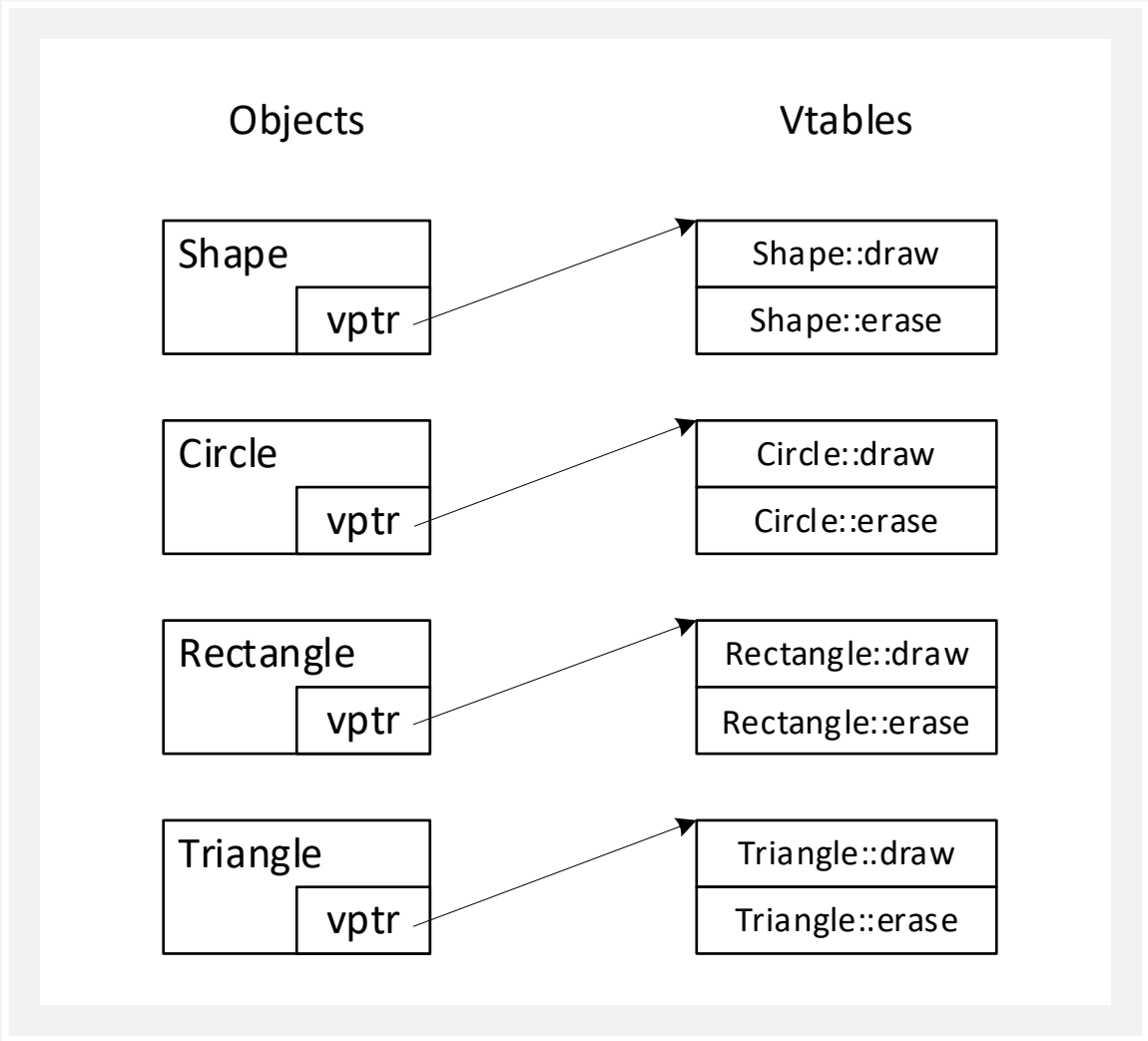
SHAPE HIERARCHY

- Features needed for polymorphism
  - Inheritance
  - Function override
  - virtual function

CLIENT CODE

```
Shape* s = new Circle;
        .
        .
        .
c->draw();
```

- Features needed for polymorphism
  - A pointer variable
  - an up-cast



IMPLEMENTING POLYMORPHISM

- When a class has virtual functions
  - The class has a virtual table
  - The table is a list of function pointers
  - Instances of the class have a virtual pointer
- Running a polymorphic function
  - Retrieves the address in the virtual pointer
  - Follow it to the class's virtual table
  - Search for the function by name
  - Run or dispatch the function