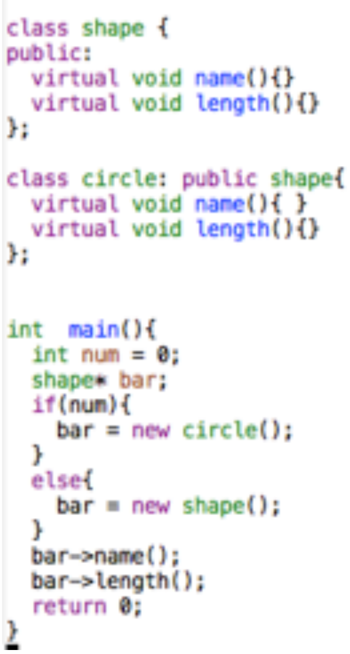
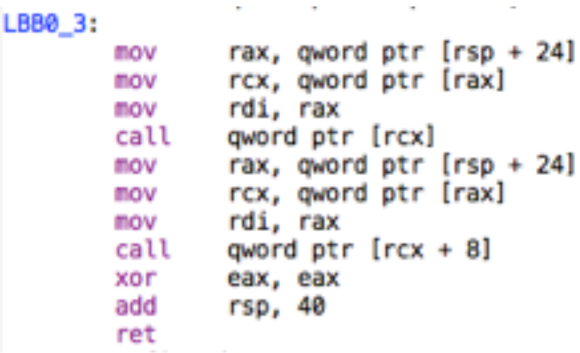
Pat Wongwiset
nw9ca
11/14/17
inlab09.pdf

Dynamic Dispatch



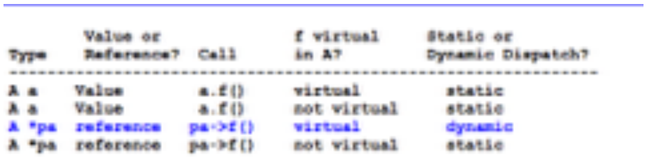
```
class shape {
public:
  virtual void name(){}
  virtual void length(){}
};

class circle: public shape{
  virtual void name(){ }
  virtual void length(){}
};


int  main(){
  int num = 0;
  shape* bar;
  if(num){
    bar = new circle();
  }
  else{
    bar = new shape();
  }
  bar->name();
  bar->length();
  return 0;
}
```



```
LBB0_3:
        mov     rax, qword ptr [rsp + 24]
        mov     rcx, qword ptr [rax]
        mov     rdi, rax
        call    qword ptr [rcx]
        mov     rax, qword ptr [rsp + 24]
        mov     rcx, qword ptr [rax]
        mov     rdi, rax
        call    qword ptr [rcx + 8]
        xor     eax, eax
        add     rsp, 40
        ret
```

The virtual method is powerful when the program cannot determine which method it should use until runtime. The program would pass the virtual method by calling from the address, not from the method's name directly. There exists a virtual table which stores all the virtual methods in the program and can be pointed by a register, rcx from the code above. The virtual method in the virtual table is offset by 8 bytes in the virtual table. Since all virtual methods are stored in the virtual table as a memory address, a subclass can change the behavior of the base class directly.

When does C++ use Dynamic Dispatch



Type	Value or Reference?	Call	f virtual in A?	Static or Dynamic Dispatch?
A a	Value	a.f()	virtual	static
A a	Value	a.f()	not virtual	static
A *pa	reference	pa->f()	virtual	dynamic
A *pa	reference	pa->f()	not virtual	static

http://condor.depaul.edu/ichu/csc447/notes/wk10/Dynamic2.htm

The program will use dynamic dispatch when the method can be passed by memory address.