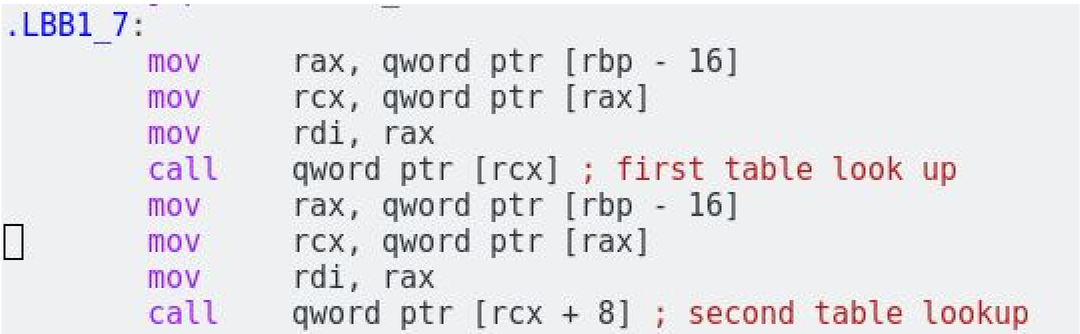
Holt Crews, jhc6we, 11/15/17, inlab09.pdf

Dynamic Dispatch

As we learned in lecture, when doing a dynamic dispatch via the virtual keyword, a virtual method table is created that is called upon at runtime to determine which function to use. These method tables have addresses to the method of the class and superclass. There are three pointer dereferences that have to take place in order for the program to locate the correct function. The virtual method has to follow the pointer to the object, then go to the virtual method table pointer, lookup that method pointer and then jump to that method. In assembly, the address of the virtual table is first moved to rax, then the address of the first function is loaded into rcx, then rax is moved to rdi, and then assembly calls the function at rcx if the first function of the table is need and increases by 8 ([rcx+8]) for every function after that. This behavior is demonstrated in the following code in which two virtual methods are called in main through dynamic dispatch. The assembly code accesses a virtual method table that has been generated prior and then within that table accesses the method necessary.



```
.LBB1_7:
        mov     rax, qword ptr [rbp - 16]
        mov     rcx, qword ptr [rax]
        mov     rdi, rax
        call    qword ptr [rcx] ; first table look up
        mov     rax, qword ptr [rbp - 16]
        mov     rcx, qword ptr [rax]
        mov     rdi, rax
        call    qword ptr [rcx + 8] ; second table lookup
```