

Introduction to Intel x86-64 Assembly, Architecture, Applications, & Alliteration

Xeno Kovah – 2014-2015
xeno@legbacore.com

All materials is licensed under a Creative Commons “Share Alike” license.

- <http://creativecommons.org/licenses/by-sa/3.0/>

You are free:



to **Share** — to copy, distribute and transmit the work



to **Remix** — to adapt the work

Under the following conditions:



Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

Attribution condition: You must indicate that derivative work
"Is derived from Xeno Kovah's 'Intro x86-64' class, available at <http://OpenSecurityTraining.info/IntroX86-64.html>"

Attribution condition: You must indicate that derivative work

"Is derived from Xeno Kovah's 'Intro x86-64' class, available at <http://OpenSecurityTraining.info/IntroX86-64.html>"

SimpleCppClass.cpp

Stack diagramming C++ classes

```
#include <stdio.h>
#include <string.h>

class Person {
protected:
    char name[24];
public:
    Person(char *_name) {
        memset(name, 0xCC, 24);
        strcpy(name, _name);
    }
    virtual void work() {
        printf("%s awaits the heat death of the universe\n", name);
    }
};

class Uber1337Haxor : public Person {
public:
    Uber1337Haxor(char *_name) : Person(_name) {}
    virtual void work() {
        printf("%s hax BIOS!\n", name);
    }
};

void main(){
    Person Corey("Corey");
    Person TheDude("TheDude");
    Uber1337Haxor Xeno("Xeno");
    Corey.work();
    Xeno.work();
    TheDude.work();
}
```

Work it out

- What does the stack diagram look like when there are C++ objects present?
- Debugger has the answer!