

```

// Calling superclass virtual functions
#include < coda-c.h>
int cc_toCap(int c) { return( c>='a' && c<='z'? c-32: c); }
void cs_cap(char* str) { while(*str) { *str = cc_toCap(*str); ++str; } }

typedef void OSig$(saySomething, char* greeting); // define virtual call signature

#define class XYZPerson
CodaClassZerosC();
CodaClass(XyzPerson, char, Root); // unused 'char' storage - 1 byte

// void XyzPeson_sayHello(XyzPerson self) { ... } // generate this function
void$(sayHello) { obj_(saySomething, self, "Hello, world!"); } // virtual call

// Generate this function and register it as virtual function for this class.
// static void XyzPeson_saySomething(XyzPerson self, char* greeting) { ... }
method$(void, saySomething, char* greeting) { printf("%s\n", greeting); }
#undef class // XyzPerson

#define class XyzShoutingPerson
CodaClassZerosC();
CodaClass(XyzShoutingPerson, short, XyzPerson); // unused 'short' storage - 2 bytes

method$(void, saySomething, char* greeting) {
    cleanOChar upper = Char_Value(greeting); cs_cap(upper); // cleanO: release 'upper' at scope end
    super_(saySomething, self, upper); // virtual call to superclass
}
#undef class // XyzShoutingPerson

int main() {
    Obj ppp = new0(XyzPerson);
    XyzPerson_sayHello(ppp);

    Obj sss = new0(XyzShoutingPerson);
    XyzPerson_sayHello(sss);
}

/* OUTPUT:
Hello, world!
HELLO, WORLD!
*/

```