

```

// Stephen M. Jones // 05/12/2024

// Coda-C example #6:  Defining a new class

// Task:  Create a new class 'Short' based on C data type 'short' with a virtual function.

// header // typedef short* Short; extern Short Class_Short; extern CodaCLASS Isa_Short;
#define class Short
    CodaClassZerosC();
    CodaClass(Short,short,Root);

ShortShort_Value(shortvalue) { Shortself=new0(Short); *self=value;return(self); }

static char* Short_xmlTag(Shortself) { return "short"; }

static$CONSTRUCTORvoidShort_boot() {
    CodaSig(xmlTag);
}
#undef class // Short

static void codax06() {
    clean0 Short sss=new0(Short); *sss=12345;
    printf("Short value.: %d\n",*sss);
    printf("Short xmlTag: %s\n",obj_(xmlTag,sss));
}

codax_register(codax06)

// Purpose:  Show how an object Class is defined.

```

```
// Coda-C adds the following:

// CodaClassZerosC() - defined as: CodaClassZeros(dtor,itor,kize,etor,ekeep,bits)
// CodaClassZeros() - defines zero enums for the current class like 'Short_dtor' etc
// CodaClass()       - Coda-C new class macro
// $CONSTRUCTOR      - C attribute to execute before main()
// CodaSig()         - register a virtual function inside a class definition

/*<stdout> example's output
Shortvalue.:12345
ShortxmlTag:short
*****/
```