

```

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

// Coda-C example #8:  Defining a new class by structure

// Task:  Create a new class 'Figure' with an initializer and destructor.

    structFigure_ {
        charname[64];
        Arraysides;
    };

#define class Figure
    static void Figure_itor(Figureself) {
        strcpy(self->name,"default");
        self->sides=new0(Array);
        printf("Initializer called.\n");
    }
    static void Figure_dtor(Figureself) {
        free0(self->sides);
        printf("Destructor called.\n");
    }
    CodaClassZeros(kize,etor,ekeep,bits);
    CodaClass(Figure,structFigure_,Root);
#undef class // Figure

// usage example

static void codax08() {
    clean0Figure fff=new0(Figure);

    printf("Figure name: %s\n",fff->name);
    printf("Figure sides type: %s\n",kind0(fff->sides));
}

codax_register(codax08)

// Purpose:  Show how more complex object Classes are defined.

```

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

// *_itor()          - each class may use an initializer
// *_dtor()          - each class may use a destructor

/*<stdout> example's output
Initializer called.
Figure name: default
Figure sides type: Array
Destructor called.
*****/
```