

```

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

// Coda-C example #7:  Defining a signature

// Task:  Define a signature.

    typedef double OSig(getArea)(Objobj); // place in header

    sig_(getArea);

    externConstChar sig_getArea; // place in header

// usage example

#define class Square
    CodaClassZeroSC();
    CodaClass(Square, double, Root);
    static double Square_getArea(Squareself) { returnself[0]*self[0]; }
    $boot(codax) { CodaSig(getArea); }
#undef class // Square

    #include <math.h>

#define class Circle
    CodaClassZeroSC();
    CodaClass(Circle, double, Root);
    static double Circle_getArea(Circleself) { returnM_PI*self[0]*self[0]; }
    $boot(codax) { CodaSig(getArea); }
#undef class // Circle

static void codax07() {
    Square sss=new0(Square); *sss=50.0;
    Circle ccc=new0(Circle); *ccc=25.0;
    printf("Square area: %6.5f\n", obj_(getArea, sss));
    printf("Circle area: %6.5f\n", obj_(getArea, ccc));
}

codax_register(codax07)

// Purpose:  Demonstrate how to define a Coda-C signature and use it.

```

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

// $boot()          - shorthand for class $CONSTRUCTOR function
// OSig()           - macro used to identify Coda-C signatures
// sig_()           - marco used to create a unique signature ConstChar global object

/*<stdout> example's output
Square area: 2500.00000
Circle area: 1963.49541
*****/
```