

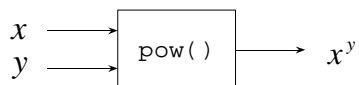
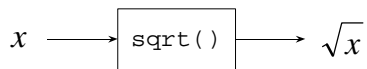
Implementing Complex Objects

- **Construct from primitives**
 - Like “calculator” object
- **Attributes**
 - Implement with data objects
- **Behavior**
 - Implement with functions

Functions

- **Have already seen main()**
 - Found in (almost) every program
- **Most programs have many**
 - Built-in (from libraries)
 - Like sqrt()
 - User-defined

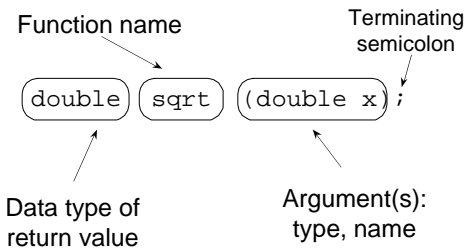
Function Operation



Function Interface

- **Name of the function**
- **Function argument(s)**
 - Zero or more
 - Argument type(s)
- **Function return value**
 - Return value type

Function Prototype



More Prototype Examples

```
double pow (double x, double y);

char MakeLower (char c);

int GetNextBadgeNumber ();
```

⋮

Header Comments

```

double RectArea (double width,
                 double height);
// Arguments:
//   width - width of rectangle
//   height - height of rectangle
// Returns: area of rectangle

```



⋮

Implementation

```

Function header
double RectArea (double width,
                 double height)
{
    Function body
    double area;
    area = width * height;
    return (area);
}

```

No semicolon

Return value



⋮

Function Call Operation

```

double sqr (double x);
void main ()
{
    cout << (sqr (1.5)) + 3.5;
}

double sqr (double x)
{
    return (x*x);
}

```



Function Argument

- **Formal arguments**
 - Like data objects inside function
 - Initialized by corresponding actual arguments
- **Actual arguments**
 - Passed in function call
 - Often expressions

Function with No Arguments

```
double Zero ();
void main ()
{
  cout << Zero ();
}

double Zero ()
{
  return 0.0;
}
```

Still need parentheses!

return Statements

- **Must have at least one**
 - For any function returning a value
- **Can have more than one**
 - Function exits immediately
- **Preferred practice**
 - Only one return, at end of body
