

Constructors Revisited

1/22/99 125

Constructors

In C++, the **constructor** is a special function automatically called when a class instance is declared

- Constructor's name is class name
- No explicit return type, not even *void...*

1/22/99 126

Multiple Constructors

- May be several reasonable ways to initialize a class instance
- Multiple constructors
- All have same name (name of class)
- Distinguished by number and types of arguments
- Example of “overloading” (more later)

1/22/99 127

Multiple Constructors (review)

```
// complex.h -- simple complex number class
class Complex {
public:
    // constructors:
    Complex( );                // = 0+0i
    Complex(double a, double b); // = a+bi

    // double->Complex constructor
    Complex(double r);          // = r+0i
    ...
};
```

1/22/99 128

Default Constructor

- If no explicit constructor is given, a default is supplied by compiler
 - Takes no arguments, does nothing
 - Not guaranteed to perform **any** initialization
 - Invisible
- If a class has one or more “non-default” constructor:
 - then NO “default” constructor will be supplied; variable declaration without initialization fails.

1/22/99 129

Guidelines for Constructors

- A constructor cannot return a value
 - so it must be declared without a return type
- A class may provide multiple constructors
 - Compiler will choose appropriate one, depending on context.
- Syntax for invoking a constructor

```
Complex a1;
Complex a3 = Complex(123.45);
Complex a2(10.0, -1);
```

But not this:

```
Complex a4();
```

1/22/99 130