

# The Class Construct – Part 1

Lecture 22

Sections 7.5 - 7.6

Robb T. Koether

Hampden-Sydney College

Mon, Oct 28, 2013

- 1 The Class Construct
- 2 The `Point` Class Example
- 3 Constructors
- 4 Assignment

# Outline

- 1 The Class Construct
- 2 The `Point` Class Example
- 3 Constructors
- 4 Assignment

# The Class Construct

## The Class Construct

```
class name
{
    public:
        member-function-prototypes

    private:
        data-member-declarations
};
```

- The **class** construct has the above form.

# The Class Construct

- Each member function appears as a prototype.
- Each data member appears as a declaration.
- The public part of the class construct describes the user interface.
- Note that the whole thing ends with a semicolon.

# Outline

- 1 The Class Construct
- 2 The `Point` Class Example**
- 3 Constructors
- 4 Assignment

# Point Class Example

## The Point Class

```
class Point
{
    // Data members

    private:
        double x; // x-coordinate
        double y; // y-coordinate
};
```

# The Point0 Class

- Example

- `Point0.h`
- `Point0Test.cpp`



# The Member Functions

- For convenience, we categorize member functions as follows.
  - Constructors
  - The destructor
  - Inspectors
  - Mutators
  - Facilitators
  - Operators
  - Other member functions

# Outline

- 1 The Class Construct
- 2 The `Point` Class Example
- 3 Constructors**
- 4 Assignment

# The Member Functions

- A **constructor** is called when a new object is created. It initializes the object.
- The **destructor** is called when an object passes out of scope.
- An **inspector** returns an attribute of the object, but does not change the object.
- A **mutator** changes one or more data members of the object.

# The Member Functions

- A **facilitator** is used to help implement an operator.
- An **operator** is a function that can be invoked through a symbol such as  $+$  or  $*$ .
- The other member functions are the ones that do not fall into any of the previous categories.

# Constructors

- A **constructor** is a function that creates an instance (an object) of a class.
- A constructor (automatically) allocates space in memory for the object.
- A constructor typically will initialize the object's data members.
- A constructor's name must be the same as the class name.
- A class may have many constructors, each with a different signature.

# Constructors

- A constructor has no return type.
- A constructor is invoked by declaring an object of that type.

# Point Class Example

## The Point Class

```
class Point
{
    public:

    // Constructors

    Point();
    Point(double xval, double yval);
    Point(const Point& p);
    :
};
```

# The Point1 Class

- Example

- Point1.h
- Point1.cpp
- Point1Test.cpp



# Outline

- 1 The Class Construct
- 2 The `Point` Class Example
- 3 Constructors
- 4 Assignment**

# Assignment

## Assignment

- Read Sections 7.5 - 7.6.