The Class Construct

Defining objects with attributes and behavior

Class Types

- Class construct
  - Allows programmers to define new data types for representing information
  - Class type objects can have both attribute components and behavior components
  - Provides the object-oriented programming in C++
- Example we shall consider is
  - RectangleShape

Terminology

- Client
  - Program using a class
- Object behaviors
  - Realized in C++ via member functions (methods)
    - RectangleShapes can be drawn or resized
- Object attributes
  - Are known as data members in C++
    - RectangleShapes have width, height, position, color

Member Functions

- Provide a controlled interface to data members and object access and manipulation
  - Create objects of the class
  - Inspect, mutate, and manipulate object of the class
  - Can be used to keep data members in a correct *state*
    - SetSize()
    - SetColor()
    - Draw()

Member Functions

- *Constructors*
  - Member functions that initialize an object during its definition

```
RectangleShape R(W, x, y, c, w, h);
```

  - Factoid
    - Constructors do not have a type
      - Considered superfluous

Member Functions

- *Inspectors*
  - Member functions that act as a messenger that returns the value of an attribute
  - Example
    - RectangleShapes have an inspector GetColor()

```
color CurrColor = R.GetColor();
```

Member Functions

- *Mutators*
  - Changes the value of an attribute
  - Example
    - RectangleShapes have a mutator SetColor()

```
R.SetColor(Black);
```

Member Functions

- *Facilitators*
  - Causes an object to perform some action or service
  - Example
    - RectangleShapes have a facilitator Draw()

```
R.Draw();
```

A Simple RectangleShape Class

- Consider a simpler version of the RectangleShape than what is defined in rect.h
- Giving the class definition *not* the implementation
- The definition in rect.h uses inheritance and member functions with default parameters
  - If you are wondering what is missing
    - Default constructor parameters
    - Member function
      - Erase()
    - Inherited member functions
      - HasBorder(), SetBorder(), and ClearBorder()

Simple RectangleShape Header File

```
#ifndef RECT_SHAPE_H
#define RECT_SHAPE_H
#include "ezwin.h"
class RectangleShape {
  public:
    // constructor
    RectangleShape(SimpleWindow &Window,
     float XCoord, float YCoord, const color &c,
     float Width, float Height);
    // facilitator
    void Draw();
```

Preprocessor directives

Passed by reference, do not want a copy of the window

Access right indicates no limitations on who can use these members

ezwin.h get us definitions of SimpleWindow and color

Simple RectangleShape

```
// inspectors
color GetColor() const;
float GetWidth() const;
float GetHeight() const;
void GetSize(float &Width, float &Height)
 const;
void GetPosition(float &XCoord, float &YCoord)
 const;
SimpleWindow& GetWindow() const;
```

Indicates the member functions won't change the object

Reference return, brings actual window (not a copy)

Simple RectangleShape

Lack of const indicate the member function might change the object

```
// mutators
void SetColor(const color &c);
void SetPosition(float XCoord, float YCoord);
void SetSize(float Width, float Height);
```

Simple RectangleShape

```
  private:
    // data members
    SimpleWindow &Window;
    float thisXCenter;
    float thisYCenter;
    color thisColor;
    float thisWidth;
    float thisHeight;
};

#endif
```

Access right

A client cannot directly access either private or protected data members

Close of #ifndef directive

Access Tests

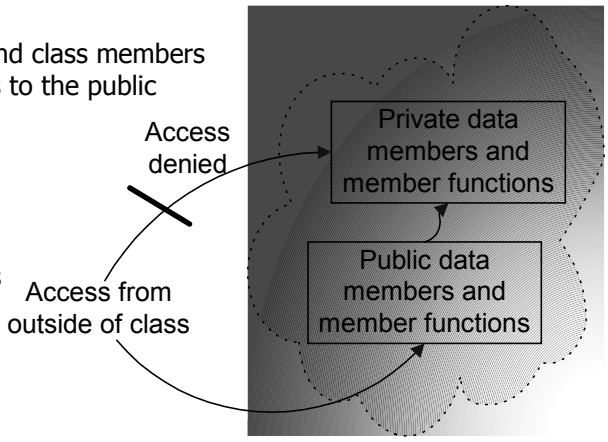
- Consider

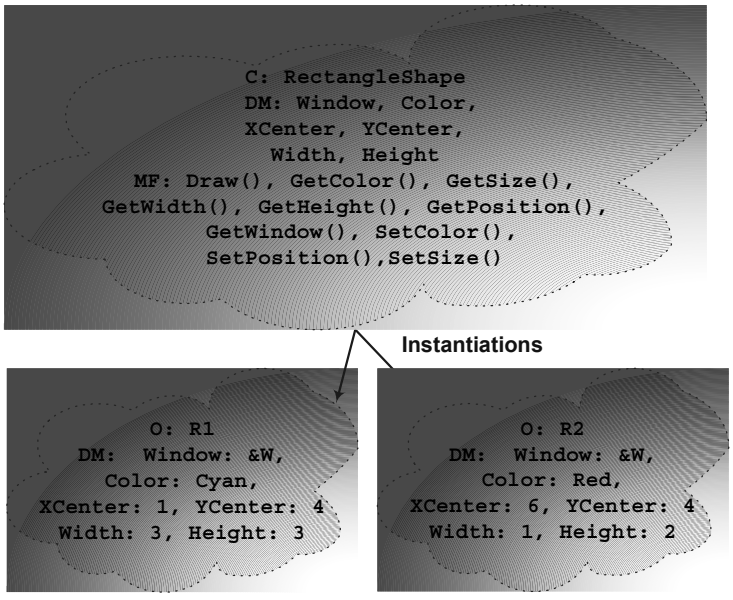
```
SimpleWindow W("Testing", 20, 10);
RectangleShape R(W, 2, 2, Blue, 4, 3);
const RectangleShape S(W, 15, 10, Red, 5, 6);
```

- Can we do the following?
  - `color c = R.GetColor();`
  - `color d = S.GetColor();`
  - `color d = R.thisColor;`
  - `R.DetColor(Yellow);`
  - `S.SetColor(Black);`

The RectangleShape Class

- Public access
  - All clients and class members have access to the public members
- Private access
  - Only class members have access to the private members





```
#include "rect.h"
SimpleWindow ColorWindow("Color Palette", 8.0, 8.0);
int ApiMain() {
  const int SideSize = 1;
  float XPosition = 1.5;
  const float YPosition = 4;
  ColorWindow.Open();
  RectangleShape ColorPatch(ColorWindow,
   XPosition, YPosition, White, SideSize, SideSize);
  for (int c = Red; c <= Magenta; c = color(c + 1)) {
      ColorPatch.SetColor(color(c));
      ColorPatch.SetPosition(XPosition, YPosition);
      ColorPatch.Draw();
      XPosition += SideSize;
  }
  return 0;
}
```