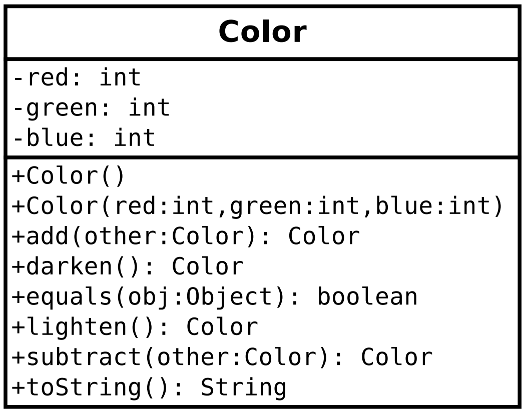
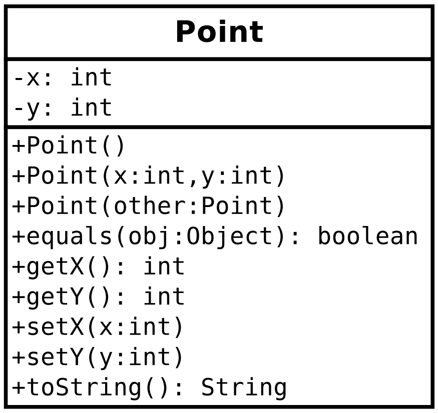
Model 1 Common Methods

Classes are often used to represent abstract data types, such as `Color` or `Point`:



```
Color
-red: int
-green: int
-blue: int
+Color()
+Color(red:int,green:int,blue:int)
+add(other:Color): Color
+darken(): Color
+equals(obj:Object): boolean
+lighten(): Color
+subtract(other:Color): Color
+toString(): String
```



```
Point
-x: int
-y: int
+Point()
+Point(x:int,y:int)
+Point(other:Point)
+equals(obj:Object): boolean
+getX(): int
+getY(): int
+setX(x:int)
+setY(y:int)
+toString(): String
```

As shown in the UML diagrams, classes generally include the following kinds of methods (in addition to others):

- **constructor** methods that initialize new objects
- **accessor** methods (getters) that return attributes
- **mutator** methods (setters) that modify attributes

Questions (15 min)

Start time:

1. Identify the constructors for the `Color` class. What is the difference between them?

There are two constructors: one that takes no parameters (the default constructor), and one that takes three integers for the RGB values.

2. What kind of constructor does the `Point` class have that the `Color` class does not?

The `Point` class also has a copy constructor: one that "copies" the values of another object.

3. Identify an accessor method in the `Point` class.

a) What is the name of the method? `getX` or `getY`

b) Which instance variable does it get? `this.x` or `this.y`

c) What arguments does the method take? none

d) What does the method return? The value of `x` or `y`

4. Identify a mutator method in the `Point` class.

a) What is the name of the method? `setX` or `setY`

b) Which instance variable does it set? `this.x` or `this.y`

c) What arguments does the method take? The value of `x` or `y`

d) What does the method return? nothing

5. How would you define accessor methods for each attribute of the `Color` class? Write your answer using UML syntax.

```
+getRed(): int
+getGreen(): int
+getBlue(): int
```

6. How would you define mutator methods for each attribute of the `Color` class? Write your answer using UML syntax.

```
+setRed(red:int)
+setGreen(green:int)
+setBlue(blue:int)
```

7. The `Color` class does not provide any accessors or mutators. Instead, it provides methods that return new `Color` objects. Why do you think the class was designed this way?

Other than the constructor, there are no methods that change the `red`, `green`, and `blue` values. This design makes the class immutable, which means that objects can be reused. The `String` class is also designed this way.