Computer Science 22: Object Oriented Programming

Lecture #4: Objects and Classes II

About This Lecture

- More on Classes
- What is in a “class”?
- Examples of classes in different programming languages

The Class

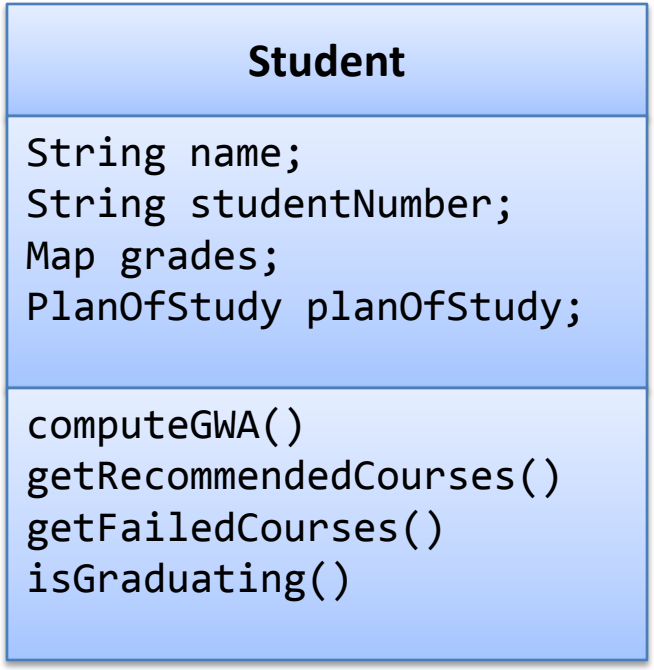
- A structure that contains data and the methods (operations) that manipulate those data
- A blueprint/template/specification for objects with similar structure and behavior

(Some) Types of Classes

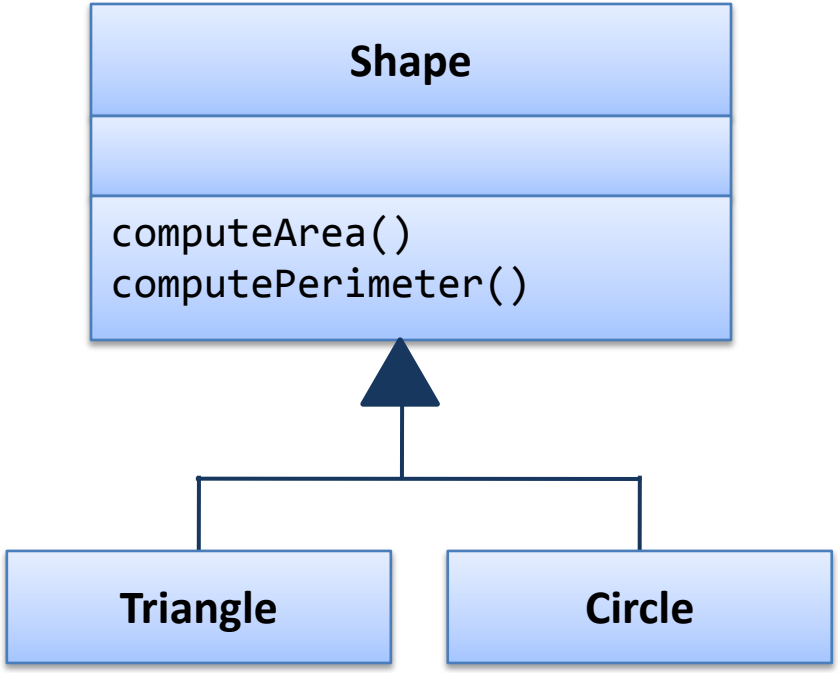
- **Concrete** class
  - Instantiatable classes (i.e., “concrete objects can be defined”)
  - Usually, a set of entities with well defined attributes and behavior
- **Abstract** class
  - Usually a generalized class of objects where some of its behaviors are left undefined
  - More concrete **subclasses** are expected to provide specific implementations

Examples

Concrete



Abstract



(Some other) Types of Classes

- **Entity Classes/Data Classes**
  - Classes whose objects encapsulate the data used in a system
- **Utility Classes**
  - Classes that maintain utility methods and usually are not instantiatable
- **Control Classes, View (Interface) Classes**
  - *Will be discussed later in the course*

What is inside a class?

- Inside a class definition:
  - Attributes of the objects that belong to the class
  - Methods (operations) representing the behaviors of the objects that belong to the class
    - List of “messages” that objects of the class respond to
  - Specialized methods
    - Constructors
    - Destructors

Additional Terms

Instance attribute/variable

- Attribute specific to an instance/object
- Each object of the class have the same set of properties BUT may have different values for each property

Class attribute/variable

- Attribute that is common to/reflected in all objects in the class

Additional Terms

Instance method/operation

- Method where the result of which is dependent on the current state of the object in question

Class method/operation

- Method that is common to all objects of the class and that the result of which is NOT tied to any particular instance of the class

Example

```
public class Bicycle(){

    private int cadence;
    private int gearNumber;
    private int speed;









}
```

Example

```
public class Bicycle(){

    private int cadence;
    private int gearNumber;
    private int speed;

    private int ID; // instance attribute or variable

    private static int numberOfBicycles = 0;  // class attribute or variable









}
```

Example

```
public class Bicycle(){

    private int cadence;
    private int gearNumber;
    private int speed;

    private int ID; // instance attribute or variable

    private static int numberOfBicycles = 0;  // class attribute or variable

    public int getID() {
          return ID;
    }

    public static int getNumberOfBicycles() {
          return numberOfBicycles;
    }

}
```

Additional Terms

- **Attributes**
  - data contained in objects
  - Properties of objects
  - Represented by variables with associated type
- **Methods**
  - Procedures/functions/subroutines that manipulate properties of objects
- **Constructors**
  - Specialized methods for initializing objects
- **Destructors**
  - Specialized methods for **destroying** objects

Code Feature: Simula

```
Class Rectangle (Width, Height); Real Width, Height;
                              ! Class with two parameters;
 Begin
    Real Area, Perimeter;  ! Attributes;
    Procedure Update;      ! Methods (Can be Virtual);
    Begin
      Area := Width * Height;
      Perimeter := 2*(Width + Height)
    End of Update;
    Boolean Procedure IsSquare;
      IsSquare := Width=Height;

    Update;                ! Life of rectangle started at creation;
    OutText("Rectangle created: "); OutFix(Width,2,6);
    OutFix(Height,2,6); OutImage
 End of Rectangle;
```

Code Feature: Smalltalk

```
Object subclass: #MessagePublisher
    instanceVariableNames: ''
    classVariableNames: ''
    poolDictionaries: ''
    category: 'Smalltalk Examples'

Object subclass: #Person
    instanceVariableNames: 'firstName lastName'
    category: 'OnSmalltalk'
```

Code Feature: C++

```
class Rectangle {
   private:
      float width, height;
   public:
      Rectangle();
      float computeArea();
      float computePerimeter();
}
Rectangle::Rectangle( ) {      }
float Rectangle::computeArea() {
      return width * height; }
```

Code Feature: Python

```
class Bag:
    def __init__(self):
        self.data = []
    def add(self, x):
        self.data.append(x)
    def addtwice(self, x):
        self.add(x)
        self.add(x)
```

Code Feature: PHP

```
class dummy {
        var $variable;
        var $variable2 = 1;
        var $variable3 = 2;
        function sum($one, $two) {
            $val = $one + $two;
            return $val;
        }
        function sum2() {
            $val = $this->sum($this->variable2, $this->variable3);
            return $val;
        }
    }
```

Code Feature: Delphi

```
type
   TStringy = Class
   private
       stText          : String;
       stWordCount     : Integer;
       stFindString    : String;
       stFindPosition  : Integer;

       procedure GetWordCount;
       procedure SetText(const Value: String);
     published
       constructor Create(Text : String);
       function Replace(fromStr, toStr : String) : Integer;
       function FindFirst(search : String) : Integer;
       function FindNext : Integer;
end;
```

Assignment

- Learn:
  - How to instantiate objects
  - How to access instance and class variables
  - How to call instance and class methods