CS 172 - Lab 10
Inheritance

Programs that do not compile will receive a grade of zero.

Style: Use indentation, spacing, and naming conventions as demonstrated by the instructor.

Include header comments and inline comments in all programs.

Read Chapter 9 before starting this lab

Review of Classes and Objects

Class – software model of a real-world thing, a blueprint, a template, a pattern

Object – a specific instance of a class, a specific person, place, thing, or event

Attributes – Data or properties of a class. In Java, the variables that contain the data of an object are called "instance variables".

Behavior – Actions that can be performed on or by objects of a class. In Java, the procedures that represent the behaviors of a class are called "methods".

Static methods – methods that act at the **class** level. Static methods are called using the name of the class. Example: **Math**.random( ), **Math**.sqrt( ).

Non-static method – methods that act on objects. Non-static methods are called using the name of an object. Example:

```
String sentence = "This is an easy assignment.";   // create a String object
System.out.println ( sentence.length( ) );
```

Special Method Types:

- Constructor – called when an object is created (instantiated). The purpose of a constructor is to initialize the instance variables of the object.
- Accessor – used to retrieve (get / access) the value of an instance variable.
- Mutator – used to change (set / mutate) the value of an instance variable.

Inheritance – the process of creating a new class from an existing one. We say that the new class is "derived from" the existing class.

The existing class is called the ***parent class***, ***base class*** or ***superclass***.

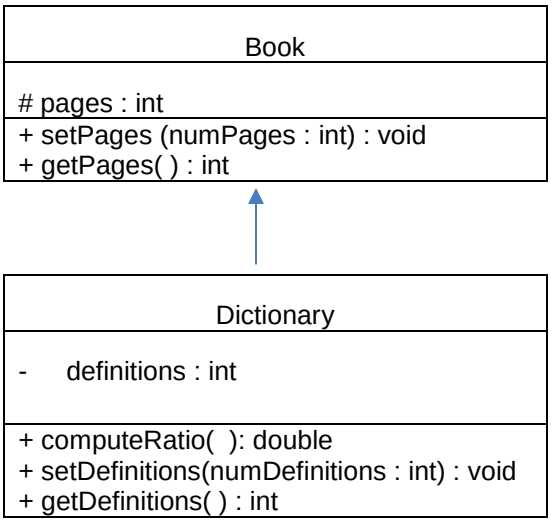
The new class is called the ***child class, derived class*** or ***subclass.***

The child class inherits data and behaviors from the parent class.

Benefits of Using Inheritance

- New classes can be created faster
- Software (methods, etc) can be reused
- Reduces costs of creating new software

Example from Chapter 9



Protected

The # symbol indicates that the instance variable is protected. The protected visibility modifier allows a variable or method to be accessed from any class in the same package or by any derived class.

Remember that private variables and method can only be accessed by name in the class where they are declared. Private variables and methods are still inherited, but they can't be accessed directly from the child class.

The Dictionary class is derived from the Book class. Every Dictionary object will have the instance variables ***pages*** (inherited from Book) and ***definitions***. Every Dictionary object will be able to call the methods ***setPages*** (inherited), ***getPages*** (inherited),

Part A:

1. Create class Box.

Box
- length : float - width : float - height : float
<<constructor>> Box ( ) <<constructor>> Box (l : float, w : float, h : float) + setLength(l : float) + setWidth(l : float) + setHeight(l : float) + getLength( ) : float + getWidth( ) : float + getHeight( ) : float + volume ( ) : float

- The volume method should return length * width * height.
- Mutators should ensure that the values stored in the instance variables are positive.
- Constructors should call mutators to set data values.
- Default Box is 1 x 1 x 1.

2. Compile and debug syntax errors in Box.java.

3. Create class PackedBox. This class inherits from class Box.

PackedBox
- contents : String - density : float
<<constructor>> PackedBox ( ) <<constructor>> PackedBox (c : String, d : float, l : float, w : float, h : float) + setDensity(d : float) + setContents(c : String) + getDensity( ) : float + getContents( ) : String + weight ( ) : float

- The weight method should return length * width * height * density.
- Mutators should ensure the density is positive and the contents are not the empty string.
- PackedBox constructors should invoke the Box constructor using this format:

  super (parameters);

  then call the mutators for density and contents.

- Default Box is 1 x 1 x 1, contents “paper”, density 20.

4. Compile and debug syntax errors in Box.java.

Part B: Create Lab10.java as a driver program.

Write a java program called Lab10.java. In the main method, thoroughly test all methods in the Box class and in the PackedBox class.

Remember that a PackedBox object will have all of the instance variables and all of the methods from the Box class as well as the instance variables and methods defined in PackedBox. Be sure to test methods setLength, setWidth, setHeight, getLength, getWidth, getHeight, and volume on PackedBox objects.

Compile, debug, test…repeat.

Submit Box.java, PackedBox.java, and Lab10.java.