CS2030 Lab

Lab 3 - Shapes?

Please talk on the Tele group!

- It will help your participation score :)
- Give your peers a chance to learn

Mistakes

Indentation

- Use spaces instead of tabs, or set your tab to auto-expand into 2 spaces

- Leave a line between each method

- Leave a line between imports and class declaration

- Between all operators, there should be a space ie "x == 0" instead of "x==0", "for (i = 0; i < x; i++)" instead of "for(i=0;i<x;i++)"

Configuring vimrc for indentation

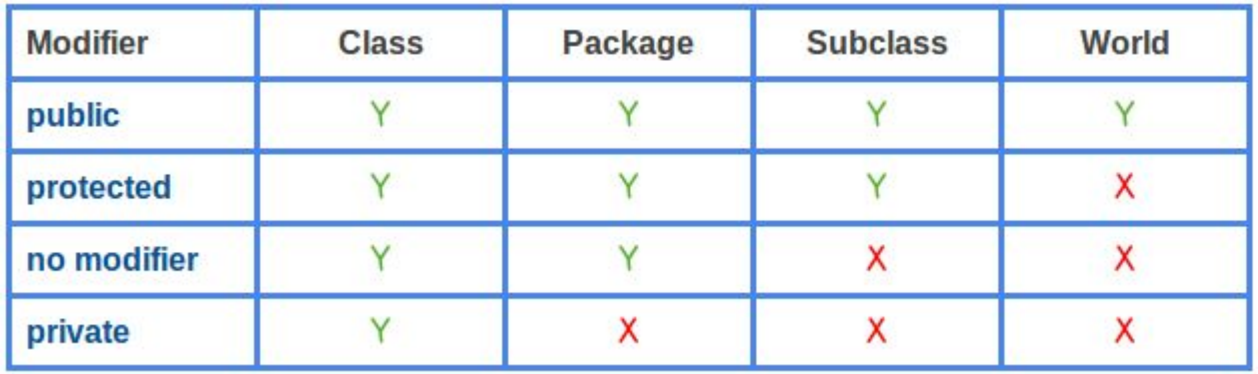
Put the following in your ~/.vimrc file to set indentations and

tab behavior

```
set tabstop=2

set shiftwidth=2

set autoindent

set smartindent
```

Access Modifiers

Public, private, protected? Remember to define access modifiers for your variables and methods! Make sure you are consistent.



Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	X
no modifier	Y	Y	X	X
private	Y	X	X	X

Static

- Static methods/variables are common to all objects of a class!
- You do not have to instantiate an object to use them, you can just call it through the class

Final

- Make variables final to ensure immutability

Take note of the special case where there is a non-primitive class data type:

```
final Point p;
```

Although you will not be able to assign p to a new point after the object has been created, you will still be able to modify it through methods.

```
final int i;
```

Whereas for int, you will not be able to change it at all.

Constants

If a value isn’t going to change at all (I.e. numofloaders in BigCruise), you should make it a constant variable that can be accessed by the class.

Although you can simply write 40 into your constructor, Prof would prefer if you defined these numbers properly as constants.

You can do this by simply doing:

```
private final static int numOfLoaders = 40;
```

Immutability!

Any questions?

Lab3

What is good design?

Subclassing

- Subclasses/children will have all the properties/methods of the superclass!
- Don’t make extra variables unless they’re special to the subclass.
- If you need specific functionality, override!

Level 1 - Cuboid

- Basic level to define an immutable Cuboid class
  - Height, width and length
  - Surface area and volume methods
  - toString()

Level 2 - SolidCuboid

- Height, width, length and density
- Density and mass methods

Level 3 - Sphere

- Is a sphere a cuboid?
- What does a sphere share with a cuboid?

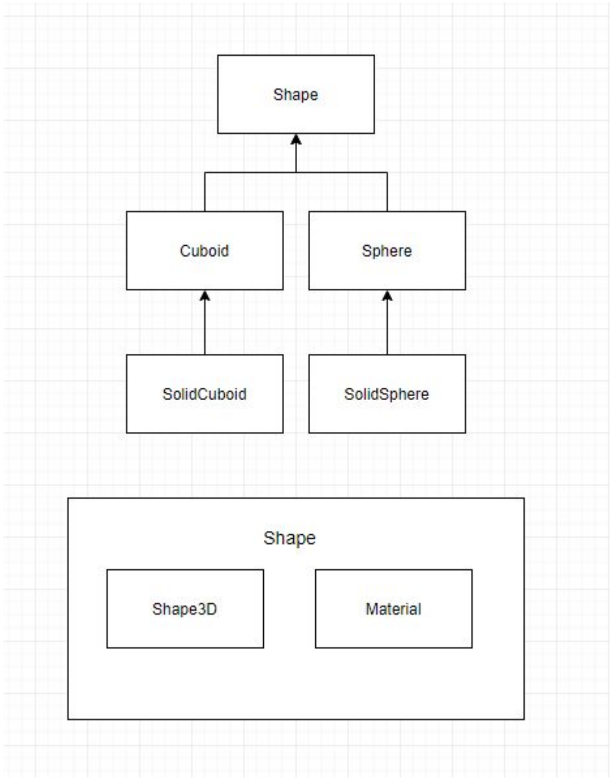
Level 4 - SolidSphere

Similar to SolidCuboid?

Level 5 - A new way

New Material abstraction

Structure and composition


Shape
Cuboid
Sphere
SolidCuboid
SolidSphere
Shape
Shape3D
Material