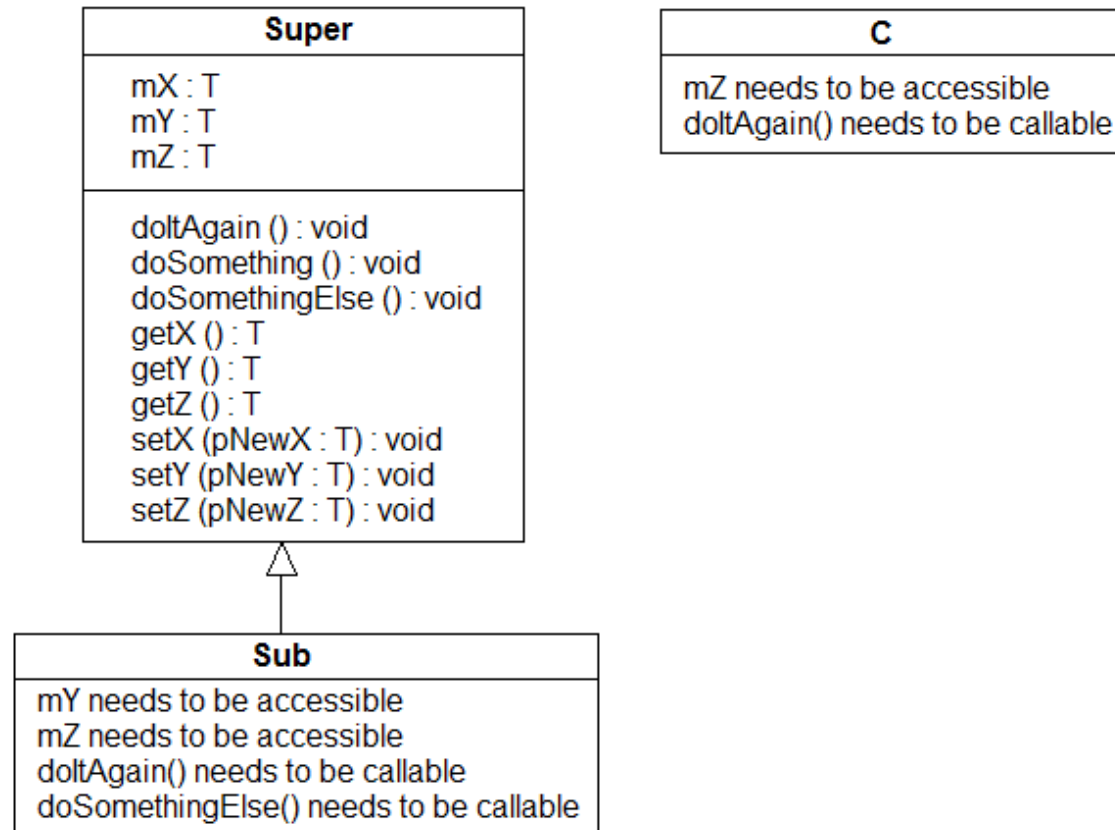


## 8. Inheritance :: Accessibility Specifiers Recommendations

1. Declare all instance variables in every class as **private**.
2. If an instance variable must not be accessible in other classes—whether subclasses or not—do not provide accessor and/or mutator methods (or provide **private** accessor/mutator methods for use within the class).
3. If an instance variable must be accessible in other classes—whether subclasses or not—declare **public** accessor and/or mutator methods to provide read/write access.
4. If an instance variable must be accessible in subclasses, but not in other nonsubclasses—declare **protected** accessor and/or mutator methods to provide read/write access to objects of subclasses.
5. If a method is only intended to be called by other methods of the same class declare the method as **private**.
6. If a method must be callable from the methods of subclass objects declare the method as **protected**.
7. If a method must be callable from the methods of any object—whether subclass objects or not—declare the method as **public**.

## 8. Inheritance :: Accessibility Specifiers Example

Example: Three classes are being designed, *Super*, *Sub*, and *C* with *Sub* inheriting from *Super* and *C* is unrelated to *Super* and *Sub*. *Super* has three instance variable, *mX*, *mY*, and *mZ* of data type *T*. *mX* is to only be accessible within the methods of *Super*. *mY* is to be accessible within the methods of *Super* and *Sub* but not in methods of *C*. *mZ* is to be accessible in the methods of *Super*, *Sub*, and *C*. *Super* has three methods *doSomething()*, *doSomethingElse()*, and *doItAgain()* where *doSomething()* is to be callable only within the methods of *Super*, *doSomethingElse()* is to be callable only within the methods of *Super* or *Sub*, and *doItAgain()* is to be callable from within the methods of any class.



## 8. Inheritance :: Accessibility Specifiers Example (continued)

```
public class Super {  
    // All instance variables are private.  
    private T mX;  
    private T mY;  
    private T mZ;  
  
    // Public methods are callable from the methods of any object of any class.  
    public void doItAgain() { ... }  
    public T getZ() { return mZ; }  
    public void setZ(T pNewZ) { mZ = pNewZ; }  
  
    // Protected methods are callable from Super and Sub methods.  
    protected void doSomethingElse() { ... }  
    protected T getY() { return mY; }  
    protected void setY(T pNewY) { mY = pNewY; }  
  
    // Private methods are callable only from methods of Super.  
    private void doSomething() { ... }  
    private T getX() { return mX; }  
    private void setX(T pNewX) { mX = pNewX; }  
}
```