**Exercise 11.7**

1. An abstract class is a class that cannot be instantiated, meaning you cannot create new instances of an abstract class. An example of this is the videoGame class; this class contains a field name and a method called go, that must be implemented by a subclass, therefore it is an abstract class.

abstract class Instrument {  
 protected String name;  
  
 abstract public void play();  
}

1. We use abstract methods to implement interfaces, thus making it more convenient than explicitly implementing subclasses.
2. Final methods are methods that cannot be overridden by a subclass in a class hierarchy. An example of this is:

public class XYZ{

public final void demo(){

System.out.println("XYZ Class Method");

}   
}   
   
public class ABC extends XYZ{

public void demo(){  
 System.out.println("ABC Class Method");  
 }   
   
 public static void main(String args[]){   
 ABC obj= new ABC();   
 obj.demo();   
 }   
}

}

1. Protected variables are variables are variables that are accessible by the class itself, the subclasses, and classes inside the same package. An example of this is defining two doubles for x y coordinates. Then passing them to a coordinate processing subclass, thus demonstrating the capabilities of protected variables.

**Exercise 11.8**

1. When locating the correct method to perform the JVM searches the class of the receiver object. If the method is non existent in that class, it proceeds to the superclass. The JVM then repeats this process until a method is located.
2. I order a package from amazon and in order for the USPS to deliver it to me amazon must provide the package to the USPS, therefore it is a dependant relationship. My phone contains many app objects, therefore it aggregates. The card class is a subclass of the deck class, therefore it inherits its characteristics.