Book 3

Chap 5

p299

Using Abstract Classes and Interfaces

- **Using Abstract Classes**

```
package Chap05_AbstractClasses_Interfaces;

/* Abstract classes and interfaces both let you declare signatures of methods and fields
   that a class implements from class itself;
 * Abstract classes through inheritance, interfaces without inheritance;
 * Default methods are designed to make interfaces easier to work with;
 * One can declare that a method or an entire class is abstract – method has no body;
 * An abstract class can't be instantiated;
 * Private method can't be abstract
 *
 */

//Page300
public abstract class Ball {                                    //class considered to be abstract as it contains
                                                                //at least one abstract method

    public abstract int hit(int batSpeed);                      //hit declared an abstract method,
                                                                //returns int value and accepts int parameter
}
```

```
package Chap05_AbstractClasses_Interfaces;

/* Class is a subclass of Ball
 * When subclassing an abstract class, subclass is to provide implementation for each abstract method in the abstract class;
 */
public class Baseball extends Ball {                            //subclass

    @Override
    public int hit(int batSpeed) {                              //Override each abstract method with
                                                                //a non abstract method
        return(batSpeed);
        // code that implements the hit method goes here
    }
}
```