Die Class Design (2.0)
Prepared by Dr. C. S. Tritt
Last revised 4/10/05

This class represents a multisided die (the singular of dice).

Its constructor requires a single integer arguments the represents the number of sides the die is to have.

It has a *void* method *roll* that simulates a role of the die sitting its valued to an integer between 1 and its number of sides. A die do not have valid values until they are rolled. This state is presented by the *public int* class constant NO_VALUE.

It has an *int* method *getValue* that returns the current value of the die. This value can only change when the *roll* method gets called.

The *die.java* file also includes the definition of the *public DieTest* class which includes a *public static void main()* method for testing of the die class. I tried several other approaches to class testing classes including a *main()* method in the class itself and a separate public test class in its own *.java* file with a *main()* method. Putting *main()* in the class made private class members too accessible. Putting the test code in a separate file seemed to divide stuff up to much and result in an "extra" file.

Internal (*private*) Details

This class uses the *Math.random()* pseudo-random number generator and does not provide any way to explicitly seed this generator.

It contains a private int data member *value* in which the current value is stored. Prior to a die's first role, the value of its *value* is NO_VALUE (I love that sentence).

The value of the NO_VALUE constant is -1.

Source code follows…

```java
// File: Die.java

import javax.swing.*; // For JOptionPane methods.

public class Die {
/*
 *  Created by C. S. Tritt. See DieClassDesign.doc for documentation
 *  (for now).
 *  Version 1.0 (last revised 3/31/05).
 */
    final static public int NO_VALUE = -1; // Indicates invalid value.
    private int sides = 6; // Number of sides, 6 is default.
    private int value = NO_VALUE; // 1 to sides

    public Die(int num_Sides) {
        // Constuctor. Sets number of sides.
        // Initial value of value set by initialization.
        sides = num_Sides;
    }

    public Die() {
        // Alternative constructor takes no argument and
        // creates 6 sided die. The empty constructor must be provide
        // because no default constructor is generated when any
        // constructor is provided.
    }

    public void roll() {
        // Simulate a role of the die.
        value = (int) (Math.random()* sides + 1);
    }

    public int getValue() {
        // Return the last rolled value.
        return value;
    }
}
```

```
class DieTest { // Test code for die class. No access modifier allowed.

    public static void main(String[] args) {
        // This method demonstrates and tests the Die class.
        int sides;
        sides = Integer.parseInt(JOptionPane.showInputDialog(
           "Enter number of sides: "));
        Die die = new Die(sides);
        int v0 = die.getValue(); // Before a roll.
        die.roll();
        int v1 = die.getValue();
        die.roll();
        int v2 = die.getValue();
        Die newDie = new Die(); // Use default constructor.
        newDie.roll();
        int v3 = newDie.getValue();
        Die thirdDie = newDie; // thirdDie now refers to newDie.
        int v4 = thirdDie.getValue(); // Use a roll twice.
        JOptionPane.showMessageDialog(null, "Values rolled: " + v0
                + ", " +  v1 + ", " +  v2 +  ", " + v3 +  ", " + v4
                + " (" + Die.NO_VALUE + " = Invalid)");
    }
}
```