Die

1. Problem Description

Your client is developing a new Java class library to support group games. As lead developer, you have been tasked to design and implement one of the most fundamental classes in this library: Die (singular of *Dice*). Typically, the Dice we are familiar with is a cube with the values 1 through 6 on different faces. When rolled, the Die lands randomly showing one of the faces up, i.e. its "value". Your client wants a more general Die that can represent a shape with any number of sides (> 1). Further, a Die can be "frozen" which will prevent its value from changing, even if someone tries to set its value or roll it.

In addition to writing the Class Die, you will need to write a test driver program to prove that the Die Class operates correctly. See the Required Test cases below, and the sample output.

2. Class Definition

Constructors (all public)

Die(int numSides): Creates a new, unfrozen Die with the given number of sides and initial value of 1. Ensure that the number of sides > 1. If not, print a message and assume 6 sides.

Die(): Creates a new Die with 6 sides by using the constructor above.

Accessors/Mutators

public void setValue(int v): Set the value of the Die, if the status is not frozen. If the Die is frozen, ignore this request, without a message. Ensure that the value is compatible with the number of sides (0 < v <= numSides). If not, print a message and set the value to 1.

public int getValue(): Return the value of the Die.

Functions

public void roll(): "Roll the Die" and set the value to the resulting value. If the Die is frozen, ignore this request, without a message. The resulting value should be between 1 and number of Sides, inclusive.

public void print(): Print a simple graphic for the Die: "[*n*:*x*]", where *n* is the value, *x* is the number of sides.

public void draw(): If the number of sides <= 6, print a more realistic graphic for the Die. If the number of sides > 6, default to print(). The example for 5 is shown below:

```
. . . . . . .
.           .
.   *   *   .
.     *     .
.   *   *   .
.           .
. . . . . . .
```

public void freeze(): Set the status of the die so that it will not change value (Set setValue and roll).

public void unfreeze(): Set the status of the die so that it is allowed to change value (See setValue and roll).

3. Notes

- I would suggest two classes: Die, and DieDriver (required). DieDriver will test Die by implementing the tests required below in “Required Test Cases”.
- Use plenty of print statements in your output so the grader can understand the test you are running.
- The Test Cases marked with an asterisk (*) should produce an error and keep going.
- Turn in only your source files.
- Because this program includes the verification built in, your verification table should match the tests and identify the expected output for each.

4. Required Main Class

DieDriver

5. Required Input

None

6. Required Test Cases

1. Test the creation of Dice. Create and print each (d1, d2, d3) of the following Dice:
   a. d1: Create a default Die, print it
   b. d2: Create a die with 4 side, print it
   c. d3: Create a die with 10 sides, print it
2. Test setting the value of Dice. Set the value and print the following Dice:
   a. d1: set to 3, print it
   b. *d2: set to 5, print it
   c. *d3: set to 0, print it
3. Test freezing a Die. Set the value and print the following Dice:
   a. d1: freeze it, set the value to 5, print it
   b. d1: unfreeze it, set the value to 5, print it
4. Test drawing a Die. Draw the following Dice:
   a. d1: Draw it
   b. d2: Draw it
   c. d3: Draw it
5. Test rolling a Die.

a. d1: Roll and print it 10 times.
b. d2: Roll and print it 3 times.
c. d3: Roll and print it 10 times.

7. Required Output

Your output should look something like the following example. It should include your name and a well-documented set of tests, as required above.

Note that your numbers rolled will likely be different, since this is based on random numbers.

```
Die Driver - E. Eckert

1. Creation of Die

   a) [1:6]
   b) [1:4]
   c) [1:10]

2. Setting Value of Die

*** Attempted to set value to 5, assumed 1.
*** Attempted to set value to 0, assumed 1.
   a) [3:6]
   b) [1:4]
   c) [1:10]

3. Freezing a Die

   a) [3:6]
   b) [5:6]

4. Drawing a Die

   a)
.......
.     .
. * * .
.  *  .
. * * .
.     .
.......

   b)
.......
.     .
.     .
.  *  .
.     .
.     .
.......

   c)
[1:10]

5. Rolling a Die

   a) [6:6] [5:6] [5:6] [6:6] [2:6] [2:6] [5:6] [3:6] [2:6] [2:6]
   b) [4:4] [1:4] [3:4]
   c) [9:10] [7:10] [3:10] [6:10] [10:10] [6:10] [5:10] [2:10] [5:10] [2:10]
```