

Craps



You are to finish the simulated game of craps shown below. In the game of craps a player rolls 2 dice. If their sum is 7 or 11 they win at once. If their sum is 2, 3, or 12 they lose at once. Otherwise, they continue rolling. If they repeat their initial number they "make their point" and win. If they roll a 7 while trying to make their point they "crap out" and lose.

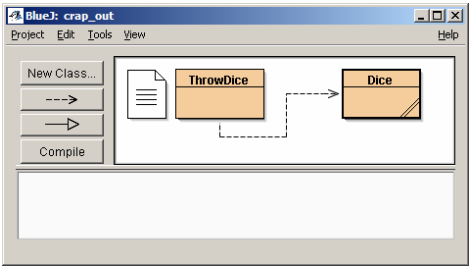
Here is the class that “runs” the simulation. It uses a loop to play three separate games of craps.

```
import java.util.*;
/**
 * Tester class
 */
public class ThrowDice {
    public static void main(String[] args){
        final int TRIES = 3;
        Dice d = new Dice(6);
        int die_1, die_2, point;
        for (int i = 1; i <= TRIES; i++) {
            System.out.println("Start Crap Game ... you throw");
            die_1 = d.cast( );
            System.out.print("[ " + die_1 + " ] ");
            die_2 = d.cast( );
            System.out.println("[ " + die_2 + " ]\n");
            point = die_1 + die_2;
            d.play( point );
        }
        System.out.println();
    }
}
```

Pass in the # of faces on the die when instantiating the class.

Add the faces of the two dice to obtain the point value.

A second class called `Dice` is used to generate a die and to implement two methods – `cast` which “rolls” a die and returns a random value from 1-6 – and `play(int)` which is passed the integer point to be made. The `play` method plays the game until the user either makes their point or craps out. The class relationships are shown below.



A partially implemented version of the `Dice` class follows.

```

import java.util.Random;
/**
 * This class models a die that, when cast, lands on a random
 * face.
 */
public class Dice {
    private Random generator;           // for random #'s
    private int sides;                 // number of sides
    /**
     * Constructs a die with a given number of sides.
     * @param s the number of sides, e.g., 6 for a normal die
     */

    public Dice(int s) {
        sides = s;                     // assign # of sides
        generator = new Random( );     // instantiate random # generator
    }

    /**
     * Simulates a throw of the die
     * @return the face of the die: 1 - number of sides
     */
    public int cast( ) {
        return 1 + generator.nextInt(sides);
    }

    /**
     * Plays the game of craps
     * @param point is the game point to be made
     */

    public void play( int point ){

        System.out.println("Play Game here - student writes the code!");

    }
}

```

A sample 3 game output sequence:

The screenshot shows a terminal window titled "BlueJ: Terminal Window - crap_out" with the following output and annotations:

```

Options

Start Crap Game ... you throw
[4] [2]

Trying to make the point 6
[5] [3]
[6] [6]
[4] [6]
[5] [1]
You made your point - you win!

Start Crap Game ... you throw
[6] [2]

Trying to make the point 8
[2] [4]
[1] [1]
[4] [6]
[4] [1]
[2] [5]
You have crapped out!

Start Crap Game ... you throw
[2] [5]

You have won!
|

```

Annotations with arrows pointing to the terminal output:

- An arrow points from the text "If not an immediate win, game announces the point to be made." to the line "Trying to make the point 6".
- An arrow points from the text "If a seven is rolled while trying to make the point you lose." to the line "You have crapped out!".
- An arrow points from the text "If you throw a 7 or 11 at the start you win." to the line "You have won!".