```
// William Kelley
// CrapsSimulationClass.cs
// ITE365-Lab04

using System;

namespace CrapsSimulationApp
{
    class MainClass
    {
        public static void Main(string[] args)
        {
            int sumOfDice = 0; // sum of the dice
            int myPoint = 0; // point if no win or loss on first roll
            Status gameStatus; // can contain CONTINUE, WON or LOST
            int roll; // number of rolls for the current game
            wins = new int[22]; // frequency of wins
            losses = new int[22]; // frequency of losses
            for (int i = 1; i <= 1000; i++)
            {
                sumOfDice = RollDice(); // first roll of the dice
                roll = 1;
                // determine game status and point based on sumOfDice
                switch ((DiceNames)sumOfDice)
                {
                    case DiceNames.SEVEN: // win with 7 on first roll
                    case DiceNames.YO_LEVEN: // win with 11 on first roll
                        gameStatus = Status.WON;
                        break;
                    case DiceNames.SNAKE_EYES: // lose with 2 on first roll
                    case DiceNames.TREY: // lose with 3 on first roll
                    case DiceNames.BOX_CARS: // lose with 12 on first roll
                        gameStatus = Status.LOST;
                        break;
                    default: // did not win or lose, so remember point
                        gameStatus = Status.CONTINUE; // game is not over
                        myPoint = sumOfDice; // remember the point
                        Console.WriteLine("Point is {0}", myPoint);
                        break;
                } // end switch
                while (gameStatus == Status.CONTINUE)
                {
                    sumOfDice = RollDice(); // roll dice again
                    ++roll;
                    // determine game status
                    if (sumOfDice == myPoint) // win by making point
                        gameStatus = Status.WON;
                    else if (sumOfDice == 7) // lose by rolling 7
                        gameStatus = Status.LOST;
                } // end while
                  // all roll results after 20th roll placed in last element
                if (roll > 21)
                    roll = 21;
                // increment number of wins in that roll
                if (gameStatus == Status.WON)
                {
                    ++wins[roll];
                    ++winSum;
```

```
        } // end if
        else // increment number of losses in that roll
        {
            ++losses[roll];
            ++loseSum;
        } // end else
    }
    PrintStats();
}

public static void PrintStats()
{
    int totalGames = winSum + loseSum; // total number of games
    int length = 0; // total length of the games
                    // display number of wins and losses on all rolls
    for (int i = 1; i <= 21; i++)
    {
        if (i == 21)
            Console.WriteLine("{0} {1} {2} {3}",
               wins[i], "games won and", losses[i],
               "games lost on rolls after the 20th roll");
        else
            Console.WriteLine("{0} {1} {2} {3}{4}",
               wins[i], "games won and", losses[i],
               "games lost on roll #", i);
        // for calculating length of game
        // number of wins/losses on that roll multiplied
        // by the roll number, then add them to length
        length += wins[i] * i + losses[i] * i;
    } // end for
          // calculate chances of winning
    Console.WriteLine("\n{0} {1} / {2} = {3:F}%",
       "The chances of winning are", winSum, totalGames,
       (100.0 * winSum / totalGames));
    Console.WriteLine("The average game length is {0:F} rolls.",
       ((double)length / totalGames));
}

public static int RollDice()
{
    // pick random die values
    int die1 = randomNumbers.Next(1, 7);
    int die2 = randomNumbers.Next(1, 7);
    int sum = die1 + die2; // sum die values
    return sum; // return sum of dice
} // end method RollDice

private static Random randomNumbers = new Random();
private enum Status { CONTINUE, WON, LOST } ;
private enum DiceNames
{
    SNAKE_EYES = 2,
    TREY = 3,
    SEVEN = 7,
    YO_LEVEN = 11,
    BOX_CARS = 12
}
static int[] wins; // number of wins, by rolls
```

```
        static int[] losses; // number of losses, by rolls
        static int winSum = 0; // total number of wins
        static int loseSum = 0; // total number of losses
    }
}
```

```
Point is 10
Point is 6
Point is 6
Point is 4
207 games won and 121 games lost on roll #1
74 games won and 129 games lost on roll #2
54 games won and 73 games lost on roll #3
40 games won and 50 games lost on roll #4
27 games won and 31 games lost on roll #5
20 games won and 31 games lost on roll #6
13 games won and 29 games lost on roll #7
9 games won and 24 games lost on roll #8
10 games won and 6 games lost on roll #9
7 games won and 6 games lost on roll #10
2 games won and 5 games lost on roll #11
4 games won and 6 games lost on roll #12
4 games won and 4 games lost on roll #13
1 games won and 2 games lost on roll #14
0 games won and 1 games lost on roll #15
4 games won and 2 games lost on roll #16
0 games won and 4 games lost on roll #17
0 games won and 1 games lost on roll #18
0 games won and 2 games lost on roll #19
0 games won and 3 games lost on roll #20
0 games won and 1 games lost on rolls after the 20th roll

The chances of winning are 476 / 1000 = 47.60%
The average game length is 3.44 rolls.

Press any key to continue...
```