

Programming Assignment #5: Let's Play Craps!

Due Date: April 6, 1998

1 The Problem

Craps is a game played with a pair of dice. In the game of craps, the shooter (the player with the dice) rolls a pair of dice and the number of spots showing on the two upward faces are added up. If the opening roll (called the 'come out roll') is a 7 or 11, the shooter wins the game. If the opening roll results in a 2 (snake eyes), 3 or 12 (box cars), the shooter loses, otherwise known as 'crapping out'. If the shooter rolls a 4, 5, 6, 8, 9 or 10 on the opening roll, then he or she must roll the same number before rolling a 7 to win the game. For example, if the shooter rolls a 6 on the come out roll, a 10 on the second roll and a 7 on the third roll, the shooter loses since he rolled a 7 before rolling another 6. If, however, he rolled a 6 on the third roll, he wins the game.

2 The Program

In this assignment, you will write a program to play a certain number of craps games and then print out statistics on the games played. You will use two classes to implement the craps games, class `di ce` and class `crapsGame`.

2.1 The class `di ce`

The purpose of this class is to establish the structure of a pair of dice and specify the operations that can be performed on a pair of dice. A partial

declaration of the class dice is given below.

```
class dice {
private:

    int getRandNum () {           //returns a random number
                                   //between 1 and 6
        return rand() % 6 + 1;
    }

    int
        face1,                    //number on face 1
        face2;                    //number on face 2

public:
                                   *****
};
```

The private section has a function called getRandNum which is coded for you. This function returns a 'random' number in the range from 1 to 6. That is, it returns a number in the speci...ed range with equal probability, namely, each integer between 1 and 6 has probability 1/6 of being selected. Note that the structure of a pair of dice can be speci...ed by a pair of integers, face1 and face2.

To implement the public section, complete the coding of the following constructors and member functions.

² dice (); //a constructor that creates a (uninitialized) pair of dice

² int getSumOfFaces (); //a member function that returns the sum of the two faces

² void roll (); //a member function that performs one roll of the dice

The member function roll should call the private function getRandNum.

2.2 The class crapsGame

This class uses the class `di ce` to describe the structure of a game of craps as well as the operations that can be performed on a game of craps. A partial declaration appears below.

```
class crapsGame {
private:
    di ce
        pai rOfDi ce; //a pai r of di ce

    i nt
        wi n,          //equals 1 i f game a win, 0 otherwise
        numRol ls;    //number of rolls in the craps game

publ i c:
        *****
};
```

To implement the public section, complete the coding of the following constructors and member functions.

- `2 crapsGame (); //a constructor that creates a craps game and sets both win and numRol ls to 0`
- `2 reset (); //sets win and numRol ls back to 0`
- `2 play (); //plays a game of craps`
- `2 getWi n (); //returns the value of win`
- `2 getNumRol ls (); //returns the value of numRol ls`

Write a test program that prompts the user to input a positive integer representing the number of craps games to be played. The program then plays the required number of games and prints out the following statistics.

- `2 number of games played`
- `2 number of wins`

- 2 estimated probability of winning at craps
- 2 total number of rolls
- 2 average number of rolls per game expressed as a decimal
- 2 number of wins that occurred on the coming out roll
- 2 estimated probability of winning a game on the coming out roll
- 2 number of losses that occurred on the coming out roll
- 2 estimated probability of losing a game on the coming out roll

The statistics should be printed out in a readable and attractive manner! To get fairly good estimates of the probabilities of winning, etc., you should run a large number of games, say 100,000 or so.

You do not have to put the class declarations and implementations in separate ...les for this program. Your test program should start as follows.

```
#include <stdlib.h>
#include <iomanip.h>
#include <iostream.h>

//Place class declarations here

void main () {

    crapsGame game;           //a craps game
    unsigned int seed;       //random number seed
    int n;                   //number of games

    cout << "Enter a seed for the ";
        << "random number generator: ";
    cin >> seed;
    srand(seed);             //start random numbers
```

The variable seed is needed to initialize the random number generator. When asked to enter a seed at the beginning of a run of your program, just type in any positive integer. Only one seed needs to be entered during the course

of your program. Random number generators are needed in game playing programs. Without randomness built into these programs, the computer would play the same game over and over again. Boring!

Your program may include other variable declarations as needed. In addition, you may decide to implement other nonmember functions in your program. For example, you might choose to call a function to print out all the statistics.

3 What To Turn In

You should turn in two files on a 3" floppy disk. The first one should be the source code for your program. This file will contain the C++ program that you write. This file must be named prog5.cpp. Always use the extension .cpp to indicate C++ source files. The other file on your disk should be the executable file produced by the compiler. This file contains the machine level instructions that the computer can actually run. (The computer cannot run the source file directly.) Depending on the compiler, the executable file may have different names. In Visual C++ (the official programming environment for the course), the executable file has the extension .exe. For example, if the source file is prog5.cpp, the executable file will be called prog5.exe. In addition to the disk, you should turn in a printout (called a "hard copy" in computerese) of your program, that is, a printout of the file prog5.cpp. The disk and printout should be placed in a two-pocket folder. On the front of the folder, the following information must appear.

Your Name Comp 170 Fall Semester 1998 Section 611 Instructor: Dr. John Del Greco

4 Late Program Policy

A program will lose 10% of its value each day it is late (excluding weekends). Starting early on your programs will maximize your chances of earning full credit on your work!