CS1021C Spring 2016 - Prof. Annexstein
Lab #3 Due Friday February 5
Title: Simulation of Game of Chance

Topics covered: Program Design, Modular Programming, Introduction to Functions, Code Reuse

Description: Your program will run a simulation of a game of chance. Consider the following simple dice game. **Game Rules:** You pick a number from 1 to 6, and call this number your point. You then roll a die 3 times. If there is any match with your point, then you win $1; otherwise you pay the house $1.

Task 1. Design a program to simulate this game. You will need a simple function rollDie() that returns a random integer in range 1-6. This was discussed in class and in the textbook. Design a loop control structure that keeps repeating the game, continuing until the user enters in an appropriate sentinel value. At the end of the game you should print out the value of the resulting winnings or losings. Verify that the program runs properly, and as expected, before continuing.

Task 2. Modify the code so that you encapsulate the logic of the game into a single function. The function should have a prototype

```
int  playOneGame  (void)
```

The function should return the int value +1 if the game is won by the user, and returns value -1 if game is lost. Note that you do not need to change the basic logic of the original program to design your function. Test your function with a driver program that prints the results of the function call.

Task 3. Modify the program so that the user is able to maintain a bank account (which is initialized with $100.00) and wager on each play. The program should prompt the user to enter a wager. Do an error check (using a while loop), so that the wager entered is no more than the current bank balance. If the wager is valid, then inform the user that they will be playing a single game. If the wager is invalid, print an error message and repeat the process.

Task 4. A play of a single game is made by calling the function you created in Task 2. A win (return value 1 by function playOneGame()) will increase the bank balance by the wager, and likewise a lose (return value of -1 by function playOneGame()) will decrease the balance by same. After the game is played, print the current balance by calling a new function void Display(float Balance) that you will write to display the current balance in an appropriate format. If the balance is greater than zero, prompt the user by asking whether to play again. Continue another round of the game until the user responds that the user is finished playing, or if the user's balance is zero.

Good Luck!