COP 3330 Suggested Exercises for Week 2

Loops

1) Write a program to take in a positive integer n > 1 from the user and print out whether or not the number the number is a perfect number, an abundant number, or a deficient number. A perfect number is one where the sum of its proper divisors (the numbers that divide into it evenly not including itself) equals the number. An abundant number is one where this sum exceeds the number itself and a deficient number is one where this sum is less than the number itself. For example, 28 is perfect since 1 + 2 + 4 + 7 + 14 = 28, 12 is abundant because 1 + 2 + 3 + 4 + 6 = 16 and 16 is deficient because 1 + 2 + 4 + 8 = 15.

2) Write a program that prompts the user to enter a single positive integer n and prints out a star design of n rows, where the first row has n-1 spaces followed by one stars, the second row has n-2 spaces followed by two stars, ..., and the last row has n stars. Here is the design that should be printed for n = 5:

```
    *
   **
  ***
 ****
*****
```

3) Write a program to play a game of marbles. The game starts with 32 marbles and two players. Each player must take 1, 2 or 3 marbles on their turn. Turns go back and forth between the two players. The winner is the person who takes the last marble. Your program should prompt each player with a message that states the current number of marbles and asks them how many they'd like to take. Continue until there is a winner. Then your program should print out the winner (either player #1 or player #2.) (Incidentally, if both players play optimally, who always wins? What is their strategy?)

Math Class

4) Research the formula for calculating the standard deviation of a set of numbers **WITHOUT** storing them in an array. Ask the user to enter how many numbers they are going to enter, and then, without using an array or a list, read in those numbers. As you are reading in those numbers you will need to maintain two running sums. After reading in all of the numbers and updating the two running sums, you'll perform one final calculation on those running sums to output the standard deviation of the data set. Use methods from the Math class as is appropriate.

5) Think of a function to integrate (probably the easiest is $f(x) = e^x$). Ask the user to enter a lower bound (x_{low}) and an upper bound (x_{high}) for x. Have your program output the area under the curve between the $x = x_{low}$ and $x = x_{high}$.

6) Write a program that asks the user to enter the three side lengths of a triangle. If the three lengths entered don't form a valid triangle, output an error message and end the program. If they do, then output the three angles of the triangle. (Hint: Look up the Law of Cosines!)

Random Class

7) Write a program to simulate the user playing blackjack. To generate a random card, generate a number from 1 to 13. Let 1 represent an Ace, 11 a Jack, 12 a Queen and 13 a King. To make life easier, force Aces to be worth 11, Jack, Queen and King to be worth 10 and each number card from 2 to 10 to be worth the number.

8) Write a guessing game where the user is guessing a number in between 1 and 100. If they guess it correct on the first guess, then tell them so and end the program. If they don't guess it correct, just say so, but don't say if their guess is too low or too high. Instead, have them enter a second different guess. Whenever the user guesses correct, tell them and end the program. For all incorrect guesses starting with the second guess, respond with "hotter" if their last guess is strictly closer to the correct number than their previous guess and respond with "not hotter" if their last guess was at least as far away from the correct number as the previous guess. Keep going until the user guesses the correct number and output how many guesses it took the user to get it.

9) Run a simulation of 10,000 coin tosses. Determine the maximum streak of heads in a row and tails in a row. Before you run your simulation, make a guess as to what ball park the answers will be in. Adjust your code to run different numbers of coin tosses. Do you see a pattern between the lengths of the longest runs and the number of coin tosses?

10) Many playoff series in sports are best of 7 games with one team playing upto four home games and the other team playing upto 3 home games. (The distribution of which team is the home team across the 7 potential games varies per sport. The NBA does 2, 2, 1, 1, 1 currently, for example.) It's presumably the case that there are two different probabilities of a team winning a single game: their probability as the home team and their probability as the away team. Ask the user to enter the probability of their team winning home games and the probability of their team winning away games, and then run a million simulations of the series, printing out what percentage of the series ran their team won.