

COP 3330 Suggested Exercises for Static Methods

Static Methods

1) Write a static method that takes in a String `str` and a character `c`, and returns the number of times `c` appears in `s`. For example, `count("hello", "l")` should return 2. Write the method and call it with several test cases you make on your own. Here's the method prototype:

```
public static int count(String str, char c);
```

2) Write a static method that takes in a positive integer `n` and returns the sum of the first `n` positive perfect squares. (Note: the first few positive perfect squares are 1, 4, 9, 16,...) To get some practice with loops, please use a loop in your method instead of the formula for this sum. Test your method by calling it with several test cases you make up. Here's the method prototype:

```
public static int sumSquares(int n);
```

3) Edit the posted program `PrintPatterns` to add a menu of choices with patterns to print out where the user is presented the menu and then chooses which design to print. Once they choose the design, the user is asked appropriate questions to specify the exact design to be printed. After your program prints the requested design, it should reprompt the user with the menu. This should continue until the user chooses to quit.

4) A nice game for students to practice their arithmetic is a game where students get practice questions that are randomly generated (either addition, subtraction or multiplication). Design such a program utilizing static methods. Feel free to add any bells and whistles you'd like to!

5) One algorithm to find an approximation to the square root of a positive real number `x` is to run a binary search of the answer for 1 fixed number of iterations. The way this works is by setting initial low and high bounds for the search. If $x > 1$, then the low bound is 1 and the high bound is `x`. If $x < 1$, then the low bound is `x` and the high bound is 1. For each iteration of the search, guess that the square root is halfway in between low and high. Then see if the guess is too low or too high. For example, if $x = 4.0$ initially, the low and high start off as 1 and 4, respectively. Our first guess is $(1+4)/2.0 = 2.5$. Notice that $2.5^2 > 4$. This means that the highest possible value for the square root is 2.5. (Whatever the square root is, it's less than 2.5.) If you repeat this process 100 times, each time either changing the low bound or the high bound to the previous guess, then the low and high bounds will be extremely close to each other and either can be used as the approximation to the answer. The method prototype is given below. Write the method and then call it with various test cases to see how well it works:

```
public static double mysqrt(double x);
```