Assignment 8/1000_2/10_8/8_{16}

Binary

Ones Comp

Twos Comp

Method Practice

In this assignment, you will develop some methods that give you practice with various aspects of logic. These are designed to help you get comfortable with calling methods, passing arguments and handling return values. Each method should have a post condition written in a comment above the method. If a precondition is warranted, please have a precondition for the parameter specified.

See the example below

```java
public class MethodPractice {
/**
  * Problem a: Area of a triangle
  * @param length length of triangle > 0
  * @param height height of triangle >0
  * @return area of triangle
  */
 public double areaTri(double length, double height)
 {
   return .5 * length * height;
 }
}
public class MethodPractTest {
        public static void main(String [] args) {
        //Use a menu in here to allow the user to choose from the different options!!!!
                Scanner input = new Scanner(System.in);
                MethodPractice mp = new MethodPractice();
                double b, h, area;
                System.out.println("Please enter a base:");
                b = input.nextDouble();
                System.out.println("Please enter a height:");
                h = input.nextDouble();
                area = mp.areaTriangle(b,h);
                System.out.println("Area = " + area);
        }
}
```

NOTE: Parameter and return documentation is worth 5 points for each method!
When do you need to specify limitations on the parameters???????????????
NOTE: If a value must be in a specific range, use 0 < variable <= 50 to show that this
variable must be positive and can be less than or equal to 50

Problem	Parameter thoughts
a	Keep 'em positive
b	Ditto
c	Make sure one of the Strings has .length() = 1
d	No restrictions, if there are no vowels or length, it's not our problem!!
e	You can't scramble AAAAAA, make sure there are at least 2 unique characters or just say you are going to "attempt" to scramble the input!
f	List the possible moods in the param, either throw an exception or just

	return a default CD if they don't hit the mood
g	The angle needs to be in degrees and needs to be restricted low and restricted high!!!!!!!!!!!!!! No triangles can have 0 or negative length sides
h	Make sure they pass the start and end in the correct order. Return either an ArrayList or an array, don't return a String
i	Duh!
j	Is your time input in milliseconds or seconds (Sleep uses milliseconds)
k	Can't really think of any limitations on inputs, use double or float so you can have decimal answeres
l	You will need to specify somehow the limit on the day as each month has a different number of days
m	Convert positive and negative numbers, how will you return a negative binary # for a string? LIMITS: No limits are really needed, if you go int, then they can only give you the max size for and int and if you go long it is likewise!!!!
n	?: How will you accept postitive and negative binary numbers? LIMITS are important here!!! If you are returning an int, ints are stored in Java using 32 bits (1 bit for the sign) so 31 1's is the most you can handle!!! But 0 is included and takes away 1 # from the positive side, so be conservative and only allow 30 bits for the number. long: a long uses 8 bytes, use the above reasoning to figure out how many digits of magnitude you can use BigInt: you're golden, these allow for arbitrarily large numbers!!!!
o	If you have a size 1 or if all Colors are the same, then can you really scramble the Array? Either soften your return tag or enforce the actual scrambling. You better use .equals for Color
p	Make sure you get 12 digits of length

1. Develop a class called MethodPractice and a separate class called MethodTester. Also, you will need a third class called MusicDisc as described below.

   a. Write a method that finds the area of a rectangle. Use real number types for the parameters and return types. (doubles or floats)

b. Write a method to find the perimeter of a rectangle.   Use real number types for the parameters and return types.

c. Write a method that takes a String and a single letter String and returns true if the original String contains that character.

d. Write a method that strips the vowels from a String and returns the resulting string. For example, stripes returns strps. **DO NOT USE .replace or .replaceAll for this method.**

e. Write a method that scrambles the letters of a String randomly. (Use Random to do this, create a Random number generator inside your method)

f. public class MusicDisc {
       public String title, artist;
       public int year;
    };
    Write a method that takes in a mood variable (use a String) and recommends a music CD to play based on that mood. IE, Moods could include; somber, party, political, last day before Xmas break, dance, hyper, etc.) You must include at least 5 moods.
    You will need to declare an instance of the MusicDisc class in your method that you can return.

    THIS SHOULD BE YOUR METHOD SIGNATURE
    public MusicDisc recommend(String mood) {
       MusicDisc md = new MusicDisc();
       //more commands to decide which disc to recommend

       md.title = "All that you can't leave behind";
       md.artist = "U2";
       md.year = 2000;
       return md;   //returns a reference to the object
    }
    This will return a MusicDisc object back to the caller of the method

g. Write a method that finds the length of the third side of a triangle, given the lengths of the other two sides and the included angle (Beware, the Math methods probably use radians, I want all inputs in degrees!)

h. Write a method that returns all of the odd numbers (inclusive) between num1 and num2 where num1 and num2 are the boundaries of a range of numbers with num1 <=num2. (ArrayList or Array are possible return types)
   NOTE: 1 9 should return 1 3 5 7 9
          2 9 should return 3 5 7 9

i. Write a method that returns the absolute value of a real number.

j. Write a method that displays a String one character at a time, with a user specifiable delay between each character void return type, must use console for this program unless you know how to use a timer.

k. Write a method that will find the distance between 2 points.

l. Write a method that takes the month and day of the year and whether or not it is a leap year and returns the number of days left in the year, ie 12/31 should return 0 (must have 3 PARAMETERS)

m. Write a method that will take in an integer(positive or negative) and convert it to binary. Use the following approach if you want. RETURN A STRING!!!!!!!!!!!!

   Divide the number by 2
   34/2 = 17 rem 0
   If you get an 1 remainder, put in a 1 for the binary number, if 0, put in a 0.
   Divide by 2 again
   17/2 = 8 rem 1
   8/2 = 4 rem 0
   4/2 = 2 rem 0
   2/2 = 1 rem 0
   1/2 = 0 rem 1

Stop here because you have reached 1 and cannot divide any further
Now read the remainders from bottom to top to figure out your binary number.
100010 = 34 in decimal

n. Write a method that will convert a number from binary to decimal. (positive or negative!) The binary number will be passed in as a String.?: How to pass in a negative number in a String?????

o. Write a method that will accept an array of Colors and will return the array shuffled up
Color[] colors = {Color.green, Color.red, Color.orange, Color.black}; will work to create a test array
NOTE: You cannot print out an array with a SINGLE LINE OF CODE, YOU NEED A LOOP
NOTE: You cannot use Collects.scramble here!!

p. Write a method that will accept a UPC bar code and return true if the check digit is correct and false if the check digit is incorrect.

The following excerpt was taken from a description of bar codes from how stuffworks.com
The last digit of the UPC code is called a **check digit**. This digit lets the scanner determine if it scanned the number correctly or not. Here is how the check digit is calculated for the other 11 digits, using the code 639382000393 from "The Teenager's Guide to the Real World" example shown above: (the last 3 is known as the check digit, it is compared to the result of the calculation below. If it matches, the bar id scanner is considered correct!!)

1. Add together the value of all of the digits in odd positions (digits 1, 3, 5, 7, 9 and 11).
   6 + 9 + 8 + 0 + 0 + 9 = 32
2. Multiply that number by 3.
   32 * 3 = 96
3. Add together the value of all of the digits in even positions (digits 2, 4, 6, 8 and 10).
   3 + 3 + 2 + 0 + 3 = 11
4. Add this sum to the value in step 2.
   96 + 11 = 107
5. Take the number in Step 4. To create the check digit, determine the number that, when added to the number in step 4, is a multiple of 10.
   107 + 3 = 110

The check digit is therefore **3**.
Each time the scanner scans an item, it performs this calculation. If the check digit it calculates is different from the check digit it reads, the scanner knows that something went wrong and the item needs to be rescanned.

Project Name	**Assign 7 – Method Practice**
Class 1 Name	MethodPractClient
	MethodPractice
	MusicDisc (contents listed above)

SEE THE TEST PLAN ON MY WEB SITE TO TEST YOUR CODE

Printing this project,
For convenience, this project will be graded in 4 separate chunks.

To print the first 4 methods and the tester for them, do the following;
   1. **At the top of your MethodPractice.java file, make the following header**
      /*_____

      Name: Mike Yee

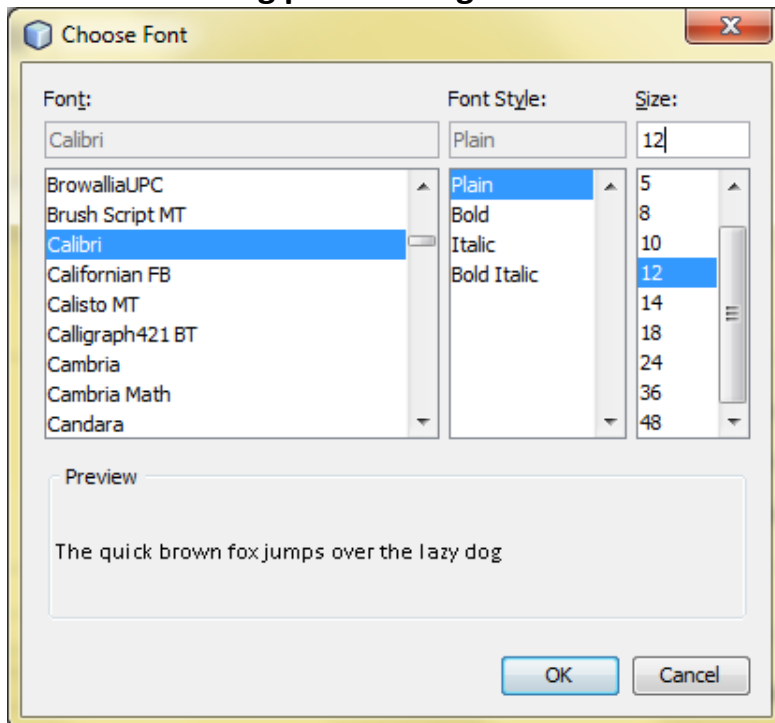   Assignment: MethodPractice, parts 1-4(a-d)

      Date: 12/15/2011
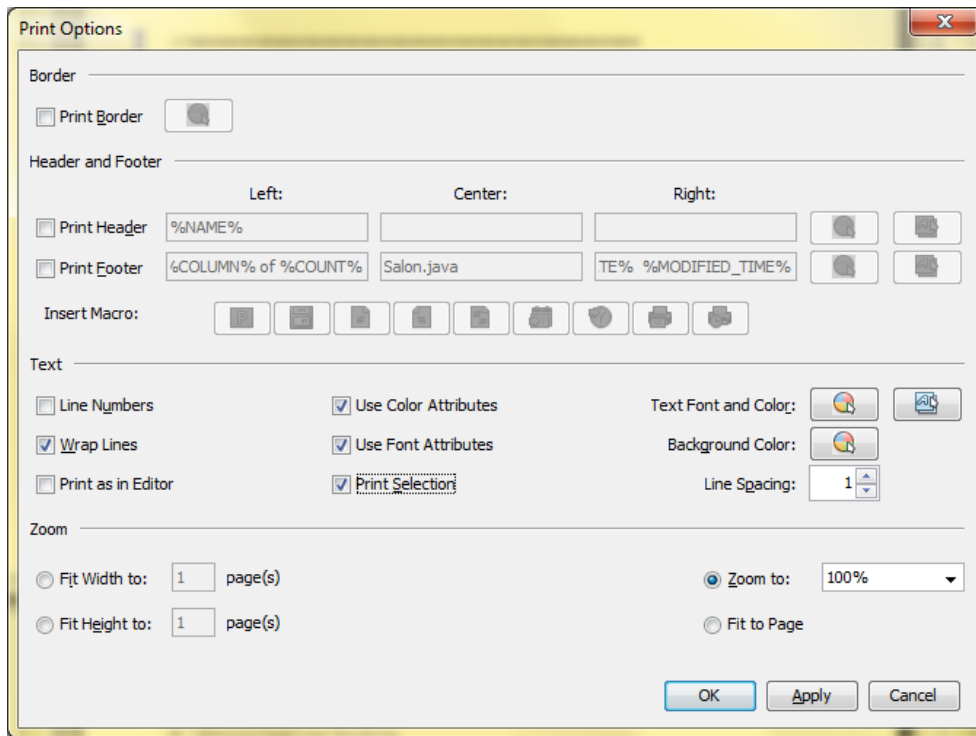
      Course: AP Computer Science

      _____*/
   2. **Use the following print settings in NetBeans:**

Calibri Plain Size 12

NO Line Numbers, header or footer!!!!!!!!!!, Wrap the lines, Print Selection

3. **Only print the methods to be graded!!!!!!!!!!!!! -5 for each method printed that is NOT part of this submission. Highlight the lines of code you want printed before going to the print command.**

4. **NO Methods may cross over a page boundary, USE PRINT PREVIEW and enters as necessary!!!!!!!!!!!!!**

5. **Use the two sided printing option for the 3410 in room 134**

6. **Staple your tester on bottom and submit to instructor. Use the print preview to make sure you don't have gross amounts of wasted paper!!!!!!!!!!!!!**

Project Pitfalls!!!!!

Historically, the following are problems encountered with this project;

1. **Students do not take advantage of the test plan!!!! Run through the test plan with your commands**
2. **For g, the third side, be sure to set your preconditions on the angle and sides**
3. **For f, students do not return a MusicDisc from a method**
4. **For n, students don't know that you can return an array of Colors from a method, ie public Color[] scramble (Color[] arr)**
5. **For a postcondition for e, HOW CAN YOU GUARANTEE TO SCRAMBLE LETTERS, ESPECIALLY IF THE STRING IS SOMETHING LIKE eeeee?**

Rubric	
Test Class (must be able to input interactively)	20
Methods a-k (10 points each) 11 methods	110
Pre and Post conditions for each (a-p) 16 methods (only need a precondition if warranted (Strings generally do NOT require preconditions!)	80
Method l (Days Left)	20
Method m (Int to Binary)	25
Method n (Binary to Int)	25
Method o (Color Scramble)	15

Method p (UPC Check Digit)	40
Comments(Each method must have comments to describe the process)	20
TOTAL	355

*Recursion*Linear Search*Binary Search*Grid World Case Study*File Processing *nlogn*Hangman*