

CS 1301 - Homework 06

Due before October 3, 1:43 am on Vista (WebCT). You can resubmit your work up until the deadline.

1. Scores2.java –

- a. Write a method, *writeGradeReport* that accepts the number of A's, B's, C's D's, and F's and displays a table like this:

Grade	Number	Percentage
A	2	16.6%
B	3	25.0%
C	4	33.3%
D	2	16.6%
F	1	8.3%

- b. Write some code to test your method. i.e. try some lines of code like:

```
// Test method
writeGradeReport(2,3,4,2,1);
writeGradeReport(1,0,0,0,0);
etc.
```

- c. **Comment out the test code.** Solve the following problem using your method. A user will enter the test scores one-at-a-time. The program will compute and display a breakdown showing the number and percentage of A, B, C, D, F's. Use the usual 10 point scale to assign grades. Display the formatted results.

2. Formatter –

- a. Write a method, *displayVals* that accepts a method that accepts two decimal numbers, the number of decimals to display, and then displays the numbers with the proper number of decimals. For example, If the method is called with: `displayVals(22.876652, 58.252356, 3)` Then the output will be:

```
22.877 58.252
```

Hint: A format string is a *String*. You can build one using concatenation.

- b. Test your method.
- c. **Comment out the test code.** Solve the following problem using your method. Display a table showing the square roots of the numbers: 10.0, 10.1, ..., 20. Prompt the user for the number of decimals to display. For instance, if the user enters 3:

Val	Square Root
10.000	3.162
10.100	3.178
...	...
20.000	4.472

3. **Reverse2.java** –

- a. Write a method, *numDigits* that accepts an integer and returns the number of digits in the number.
- b. Test your method.
- c. Write a method, *reverseNumber* that accepts an integer and returns an integer with the digits reversed. This method should use the *numDigits* method.
- d. Test your method.
- e. **Comment out the test code.** Solve the following problem using your method. Prompt the user for an integer and display the reversed number, multiplied by 2. For instance, if the user enters, 61, then 32 will be displayed. If the user enters 654321 then 246912 will be displayed.

4. **Primes2.java** –

- a. Write a method, *isPrime* that accepts an integer and determines whether it is prime or not. For instance, *isPrime(384)* will return *false* and *isPrime(17)* will return *true*. Hint: Modify your *Primes.java* code from the previous assignment.
- b. Test your method.
- c. **Comment out the test code.** Solve the following problem using your method. Prompt the user for three integers: *beg*, *end*, *numPerLine*. Your program will then find all primes between *beg* and *end* (inclusive). For instance, if *numPerLine* is 4, then display these primes 4 per line.

5. **Piecewise.java** –

- a. Write a method, *f* that accepts a decimal number and evaluates this function:

$$f(x) = \begin{cases} x^2 + 1 & -1 \leq x < 1 \\ \sqrt{x} & 1 \leq x < 100 \\ 0 & \text{otherwise} \end{cases}$$

In other words, $f(0)=1$, $f(0.5)=1.25$, $f(16)=4.0$, $f(-83.2)=0$, $f(212.3342)=0$.

- b. Test your method.
- c. **Comment out the test code.** Solve the following problem using your method. Evaluate the function for these values of *x*: -2, -1, -0.5, 0, 0.5, 1, 4, 100, 101. Just hard code the method calls.