**Scope**

This assignment covers chapter 3 and chapter 4.

**Instructions**

* Answer these questions and submit your solutions as a neatly typeset PDF by the due date reported by blackboard.
* Be sure to restate the problem including the problem number, and to answer each question in the order listed here.
* If you confuse the grader by sloppy writeup you will not recieve points for the work.
* To see the due date or submit an answer, click on the assignment name in blackboard.
* Do not forget to submit (don't just save a draft or the grader won't see it!)

**Exercises:**

**3.3**: Write a declaration for a String variable called change and initialize it to the characters stored in another String object called original with all 'e' characters changed to 'j'.

**3.7f**: Write code to declare and instantiate an object of the Random class (call the object reference variable rand). Then write a list of expressions using the nextInt method that generates random numbers in the following specified ranges, including the endpoints. Use the version of the nextInt method that accepts a single integer parameter.

**f.** –10 to 15

**3.11**: Write code statements that prompt for and read a double value from the user, and then print the result of raising that value to the fourth power. Output the results to three decimal places.

**3.12**: Write a declaration for an enumerated type that represents the days of the week.

**4.3**: What is wrong with the following code fragment? Will this code compile if it is part of an otherwise valid program? Explain.

if (length = MIN\_LENGTH)

System.out.println ("The length is minimal.");

**4.6**: Put the following list of strings in lexicographic order as if determined by the compareTomethod of the String class. Consult the Unicode chart in Appendix C in the book.

"fred"

"Ethel"

"?-?-?-?"

"{([])}"

"Lucy"

"ricky"

"book"

"\*\*\*\*\*\*"

"12345"

" "

"HEPHALUMP"

"bookkeeper"

"6789"

";+<?"

"

^^^^^^^^^^

"

"hephalump"

**4.12**: Transform the while loop given below into an equivalent for loop. (Make sure it produces the same output.)

int num = 1;

while (num < 20)

{

num++;

System.out.println (num);

}

**Programming Problems:**

**3.3**: Write an application that creates and prints a random phone number of the form XXX-XXX-XXXX. Include the dashes in the output. Do not let the first three digits contain an 8 or 9 (but don’t be more restrictive than that), and make sure that the second set of three digits is not greater than 742. Hint:Think through the easiest way to construct the phone number. Each digit does not have to be determined separately.

**3.4**: Write an application that reads the (x,y) coordinates for two points. Compute the distance between the two points using the following formula:

**3.6**: Write an application that reads the lengths of the sides of a triangle from the user. Compute the area of the triangle using Heron’s formula (below), in which s represents half of the perimeter of the triangle, and a,b, and c represent the lengths of the three sides.

Print the area to three decimal places.

**4.1**: Design and implement an application that reads an integer value representing a year from the user. The purpose of the program is to determine if the year is a leap year (and therefore has 29 days in February) in the Gregorian calendar. A year is a leap year if it is divisible by 4, unless it is also divisible by 100 but not 400. For example, the year 2003 is not a leap year, but 2004 is. The year 1900 is not a leap year because it is divisible by 100, but the year 2000 is a leap year because even though it is divisible by 100, it is also divisible by 400. Produce an error message for any input value less than 1582 (the year the Gregorian calendar was adopted).

**4.2**: Modify the solution to programming project 4.1 so that the user can evaluate multiple years. Allow the user to terminate the program using an appropriate sentinel value. Validate each input value to ensure it is greater than or equal to 1582.

**4.8**: Design and implement an application that plays the Hi-Lo guessing game with numbers. The program should pick a random number between 1 and 100 (inclusive), then repeatedly prompt the user to guess the number. On each guess, report to the user that he or she is correct or that the guess is high or low. Continue accepting guesses until the user guesses correctly or chooses to quit. Use a sentinel value to determine whether the user wants to quit. Count the number of guesses and report that value when the user guesses correctly. At the end of each game (by quitting or a correct guess), prompt to determine whether the user wants to play again. Continue playing games until the user chooses to stop

**4.10 d**: Create modified versions of the Stars program to print the following pattern.

