COMP 170 Week 3 Homework

These are Chapter 3, Programming Projects 1, 3, and 5, plus 4 for extra credit

1. (*giving change*) Repeat Programming Project 5 of Chapter 2, but include input checking. Chapter 2 PP 5 was: Write a program that determines the change to be dispensed from a vending machine. An item in the machine can cost between 25 cents and a dollar, in 5-cent increments (25, 30, 35, . . . , 90, 95, or 100), and the machine accepts only a single dollar bill to pay for the item. In this Chapter 3 version, display the change only if a valid price is entered (no less than 25 cents, no more than 100 cents, and an integer multiple of 5 cents). Otherwise, display separate error messages for any of the following invalid inputs: a cost under 25 cents, a cost that is not an integer multiple of 5, and a cost that is more than a dollar. Name your file VendingChangeImproved.java

3. (*"profane" words*) Suppose that we are working for an online service that provides a bulletin board for its users. We would like to give our users the option of filtering out profanity. Suppose that we consider the words **cat**, **dog**, and **llama** to be profane. Write a program that reads a string from the keyboard and tests whether the string contains one of our profane words. Your program should find words like **cAt** that differ only in case. Option: As an extra challenge, have your program reject only lines that contain a profane word exactly. For example, **Dogmatic concatenation is a small category** should not be considered profane. Name your file ProfaneFilter.java.

5. (*BMR extended*) Repeat the calorie-counting program described in Programming Project 8 from Chapter 2. This time ask the user to input the string “M” if the user is a man and “W” if the user is a woman. Use only the male formula to calculate calories if “M” is entered and use only the female formula to calculate calories if “W” is entered. Output the number of chocolate bars to consume as before. Name your file CandyCalculator2.java

4. (*date validation; **extra credit problem***) Write a program that reads a string from the keyboard and tests whether it contains a valid date. Display the date and a message that indicates whether it is valid. If it is not valid, also display a message explaining why it is not valid.

   The input date will have the format *mm* / *dd* / *yyyy*. A valid month value mm must be from 1 to 12 (January is 1). The day value *dd* must be from 1 to a value that is appropriate for the given month. September, April, June, and November each have 30 days. February has 28 days except for leap years when it has 29. The remaining months all have 31 days each. A leap year is any year that is divisible by 4 but not divisible by 100 unless it is also divisible by 400.
   Name your file CheckDate.java.