

## COMP 170 Week 4 Homework

These are Chapter 4, Programming Projects 3, 4, and 9, plus 10 for extra credit.

3. (*bank account value in 10 years*) Write a program that reads a bank account balance and an interest rate and displays the value of the account in ten years. The output should show the value of the account for three different methods of compounding interest: annually, monthly, and daily. When compounded annually, the interest is added once per year at the end of the year. When compounded monthly, the interest is added 12 times per year. When computed daily, the interest is added 365 times per year. *Save as [BankAccount.java](#)*

You do not have to worry about leap years; assume that all years have 365 days. For annual interest, you can assume that the interest is posted exactly one year from the date of deposit. In other words, you do not have to worry about interest being posted on a specific day of the year, such as December 31. Similarly, you can assume that monthly interest is posted exactly one month after it is deposited. Since the account earns interest on the interest, it should have a higher balance when interest is posted more frequently.

Be sure to convert the interest rate to a *double* percentage by **dividing by 100**. Also be sure to adjust the interest rate for the time period of the interest. If the rate is 5 percent, you use **5/12** percent when posting **monthly** interest and **5/365** percent when posting **daily** interest. Perform this calculation using a loop that adds in the interest for each time period, that is, do not use some sort of algebraic formula. Your program should have an outer loop that allows the user to repeat this calculation for a new balance and interest rate. The calculation is repeated until the user asks to end the program. **Hint: This difficult exercise uses nested loops.**

4. (*giving change*) Modify Programming Project 5 from Chapter 2 again to check the validity of input data in a loop. Valid input is no less than 25 cents, no more than 100 cents, and an integer multiple of 5 cents. Compute the change only if a valid price is entered. Otherwise, print separate error messages for any of the following invalid inputs: a price under 25 cents, a price that is not an integer multiple of 5, and a price that is more than a dollar, and then loop until valid data is input. *Save as [VendingChangeImproved.java](#).*

9. (*chocolate bar coupons*) Suppose we can buy a chocolate bar from the vending machine for \$1 each. Inside every chocolate bar is a coupon. We can redeem six coupons for one chocolate bar from the machine. This means that once you have started buying chocolate bars from the machine, you always have some coupons. We would like to know how many chocolate bars can be eaten if we start with  $N$  dollars and always redeem coupons if we have enough for an additional chocolate bar.

For example, with 6 dollars we could consume 7 chocolate bars after purchasing 6 bars giving us 6 coupons and then redeeming the 6 coupons for one bar. This would leave us with one extra coupon. For 11 dollars, we could have consumed 13 chocolate bars and still have one coupon left. For 12 dollars, we could have consumed 14 chocolate bars and have two coupons left.

Write a program that inputs a value for  $N$  and outputs how many chocolate bars we can eat and how many coupons we would have left over. Use a loop that continues to redeem coupons as long as there are enough to get at least one chocolate bar.

*Save as ChocolateCoupons.java*

(*Riddler's riddle; extra credit problem*) Holy digits Batman! The Riddler is planning his next caper somewhere on Pennsylvania Avenue. In his usual sporting fashion, he has left the address in the form of a puzzle. The address on Pennsylvania is a **four-digit number** where:

- All four digits are different
- The digit in the thousands place is three times the digit in the tens place
- The number is odd
- The sum of the digits is 27

Write a program that uses a loop (or loops) to find the address where the Riddler plans to strike. *Save as Riddler.java*