Tutorial 4 and 5

(Java concepts and Test where you stand?)

**Part I (Internet access permissible)**

***The first thing needed of a programmer is patient listening from client, preparing SRS, Identifying classes, methods and variables, Drawing class diagrams, then comes implementation...See how comfortable you sail with these questions given below.***

**1. Project Question I**

**PAYING OFF CREDIT CARD DEBT**

**STORYLINE**

Each month, a credit card statement will come with the option for you to pay a minimum amount of your charge, usually 2% of the balance due. However, the credit card company earns money by charging interest on the balance that you don't pay. So even if you pay credit card payments on time, interest is still accruing on the outstanding balance.

Say you've made a $5,000 purchase on a credit card with an 18% annual interest rate and a 2% minimum monthly payment rate. If you only pay the minimum monthly amount for a year, how much is the remaining balance?

You can think about this in the following way.

At the beginning of month 0 (when the credit card statement arrives), assume you owe an amount we will call *b*0 (*b* for *balance*; subscript *0* to indicate this is the balance at month 0).

Any payment you make during that month is deducted from the balance. Let's call the payment you make in month 0, *p*0. Thus, your **unpaid balance** for month 0, *u**b*0, is equal to *b*0−*p*0.

*u**b*0=*b*0−*p*0

At the beginning of month 1, the credit card company will charge you interest on your unpaid balance. So if your annual interest rate is *r*, then at the beginning of month 1, your new balance is your previous unpaid balance *u**b*0, **plus** the interest on this unpaid balance for the month. In algebra, this new balance would be

*b*1=*u**b*0+*r*12.0⋅*u**b*0

In month 1, we will make another payment, *p*1. That payment has to cover some of the interest costs, so it does not completely go towards paying off the original charge. The balance at the beginning of month 2, *b*2, can be calculated by first calculating the unpaid balance after paying *p*1, then by adding the interest accrued:

*u**b*1=*b*1−*p*1

*b*2=*u**b*1+*r*12.0⋅*u**b*1

If you choose just to pay off the minimum monthly payment each month, you will see that the compound interest will dramatically reduce your ability to lower your debt.

Let's look at an example. If you've got a $5,000 balance on a credit card with 18% annual interest rate, and the minimum monthly payment is 2% of the current balance, we would have the following repayment schedule if you only pay the minimum payment each month:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Month | Balance | Minimum Payment | Unpaid Balance | Interest |
| 0 | 5000.00 | 100 (= 5000 \* 0.02) | 4900 (= 5000 - 100) | 73.50 (= 0.18/12.0 \* 4900) |
| 1 | 4973.50 (= 4900 + 73.50) | 99.47 (= 4973.50 \* 0.02) | 4874.03 (= 4973.50 - 99.47) | 73.11 (= 0.18/12.0 \* 4874.03) |
| 2 | 4947.14 (= 4874.03 + 73.11) | 98.94 (= 4947.14 \* 0.02) | 4848.20 (= 4947.14 - 98.94) | 72.72 (= 0.18/12.0 \* 4848.20) |

You can see that a lot of your payment is going to cover interest, and if you work this through month 12, you will see that after a year, you will have paid $1165.63 and yet you will still owe $4691.11 on what was originally a $5000.00 debt. Pretty depressing!

**Questions which you need to work out based on above storyline**

## PROBLEM 1: PAYING THE MINIMUM

Write a program to calculate the credit card balance after one year if a person only pays the minimum monthly payment required by the credit card company each month.

The following variables contain values as described below:

1. balance - the outstanding balance on the credit card
2. annualInterestRate - annual interest rate as a decimal
3. monthlyPaymentRate - minimum monthly payment rate as a decimal

For each month, calculate statements on the monthly payment and remaining balance, and print to screen something of the format:

Month: 1

Minimum monthly payment: 96.0

Remaining balance: 4784.0

Be sure to print out no more than two decimal digits of accuracy - so print

Remaining balance: 813.41

instead of

Remaining balance: 813.4141998135

Finally, print out the total amount paid that year and the remaining balance at the end of the year in the format:

Total paid: 96.0

Remaining balance: 4784.0

A summary of the required math is found below:

**Monthly interest rate**= (Annual interest rate) / 12.0  
**Minimum monthly payment** = (Minimum monthly payment rate) x (Previous balance)  
**Monthly unpaid balance** = (Previous balance) - (Minimum monthly payment)  
**Updated balance each month** = (Monthly unpaid balance) + (Monthly interest rate x Monthly unpaid balance)

**Hints**

Use the [round](http://docs.python.org/library/functions.html#_blank) function!

[How to think about this problem?](https://courses.edx.org/courses/MITx/6.00.1x/3T2013/courseware/d5d822451677476fbfb0a0f9a14e0501/58b13e8f74a9407583d416d0be5ec907/)

To help you get started, here is a rough outline of the stages you should probably follow in writing your code:

* For each month:
  + Compute the monthly payment, based on the previous month’s balance.
  + Update the outstanding balance by removing the payment, then charging interest on the result.
  + Output the month, the minimum monthly payment and the remaining balance.
  + Keep track of the total amount of paid over all the past months so far.
* Print out the result statement with the total amount paid and the remaining balance.

Use these ideas to guide the creation of your code.

## 2. PROBLEM 2: PAYING DEBT OFF IN A YEAR

Now write a program that calculates the minimum **fixed** monthly payment needed in order pay off a credit card balance within 12 months. By a fixed monthly payment, we mean a single number which does not change each month, but instead is a constant amount that will be paid each month.

In this problem, we will *not* be dealing with a minimum monthly payment rate.

The following variables contain values as described below:

1. balance - the outstanding balance on the credit card
2. annualInterestRate - annual interest rate as a decimal

The program should print out one line: the lowest monthly payment that will pay off all debt in under 1 year, for example:

Lowest Payment: 180

Assume that the interest is compounded monthly according to the balance at the end of the month (after the payment for that month is made). The monthly payment must be a multiple of $10 and is the same for all months. Notice that it is possible for the balance to become negative using this payment scheme, which is okay. A summary of the required math is found below:

**Monthly interest rate** = (Annual interest rate) / 12.0  
**Monthly unpaid balance** = (Previous balance) - (Minimum monthly payment)  
**Updated balance each month** = (Monthly unpaid balance) + (Monthly interest rate x Monthly unpaid balance)

### Hints

[Hint: How to think about this problem?](https://courses.edx.org/courses/MITx/6.00.1x/3T2013/courseware/d5d822451677476fbfb0a0f9a14e0501/58b13e8f74a9407583d416d0be5ec907/)

* Start with $10 payments per month and calculate whether the balance will be paid off in a year this way (be sure to take into account the interest accrued each month).
* If $10 monthly payments are insufficient to pay off the debt within a year, increase the monthly payment by $10 and repeat.

[Hint: A way of structuring your code](https://courses.edx.org/courses/MITx/6.00.1x/3T2013/courseware/d5d822451677476fbfb0a0f9a14e0501/58b13e8f74a9407583d416d0be5ec907/)

* If you are struggling with how to structure your code, think about the following:
  + Given an initial balance, what code would compute the balance at the end of the year?
  + Now imagine that we try our initial balance with a monthly payment of $10. If there is a balance remaining at the end of the year, how could we write code that would reset the balance to the initial balance, increase the payment by $10, and try again (using the same code!) to compute the balance at the end of the year, to see if this new payment value is large enough.

* + [I'm still confused!](https://courses.edx.org/courses/MITx/6.00.1x/3T2013/courseware/d5d822451677476fbfb0a0f9a14e0501/58b13e8f74a9407583d416d0be5ec907/) **while** loops. Think hard about how the program will know when it has found a good minimum monthly payment value - when a good value is found, the loop can terminate.
* Be careful - you don't want to overwrite the original value of balance. You'll need to save that value somehow for later reference!