

Java – Credit Card Debt Part B

Purpose

This lab was designed to teach you how to use computational thinking to solve a real-world problem.

Description

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:

balance - the outstanding balance on the credit card
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 fixed monthly payment)
Updated balance each month = (Monthly unpaid balance) + (Monthly interest rate x
Monthly unpaid balance)

Hint

- 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.

Test Case 1

```
balance = 3329; annualInterestRate = 0.2
```

Output:

```
310
```

Test Case 2

```
balance = 4773; annualInterestRate = 0.2
```

Output:

440

Test Case 3

```
balance = 3926; annualInterestRate = 0.2
```

Output:

360

Test Case 4

```
balance = 70; annualInterestRate = 0.25
```

Output:

10

Test Case 5

```
balance = 938; annualInterestRate = 0.2
```

Output:

90

Test Case 6

```
balance = 793; annualInterestRate = 0.2
```

Output:

80

Test Case 7

```
balance = 835; annualInterestRate = 0.18
```

Output:

80

Test Case 8

```
balance = 4778; annualInterestRate = 0.15
```

Output:

430

Test Case 9

balance = 4510; annualInterestRate = 0.04

Output:

390

Test Case 10

balance = 4919; annualInterestRate = 0.04

Output:

420

Test Case 11

balance = 3603; annualInterestRate = 0.18

Output:

330

Test Case 12

balance = 4800; annualInterestRate = 0.2

Output:

440

Test Case 13

balance = 4053; annualInterestRate = 0.15

Output:

370

Test Case 14

balance = 4019; annualInterestRate = 0.15

Output:

360

Test Case 15

```
balance = 3618; annualInterestRate = 0.15
```

Output:

330