# Homework 01 - Coin Changing & Leetspeak Translator

Hello, and welcome! This assignment has two parts, so be sure to complete both.

## Part 1: Coin Changing

JacobCorp runs several coin changing machines (you insert coins into the machine and it returns bills). For the next part of your interview, you are to simulate the software powering these machines. Your program should prompt for the number of pennies, nickels, dimes and quarters, and output the total value of the change. The program should output the dollars and cents separately, for example, "3 dollars and 25 cents". Call this program `CoinChanger.java`.

It might take a bit of thinking, but it's possible to separate the dollar and cents without any complex math and using just the built-in arithmetic operators (hint: how many cents are in a dollar?).

Print out the number of dollars and cents as whole numbers; i.e. "4 dollars and 2 cents" as opposed to "4.0 dollars and 2.0 cents". (hint: would those decimals appear if the values for bills and cents were stored as ints?)

Example of what it might look like (user input in bold):

```
Pennies: 2
Nickels: 4
Dimes: 6
Quarters: 8

Total is 2 dollars and 82 cents.
```

You may assume the user will only enter valid, positive integer numbers.

## Part 2: Leetspeak Translator

From Wikipedia: Leet or Eleet (sometimes rendered l33t, 1337 or 31337), also known as Leetspeak or Leetzorz (1337Z0l2Z), is an alphabet used primarily on the Internet, which uses various combinations of ASCII characters to replace Latinate letters (see http://en.wikipedia.org/wiki/Leet_speak for more details).

Write a program that translates English into simple leetspeak. Call this program `LeetTranslator.java`. The user will enter a sentence in English. Convert the sentence to lower case and then translate it into leetspeak given the following set of rules:

```
"@" is equal to "a"
"3" is equal to "e"
"1" (one) is equal to "i"
"$" is equal to "s"
"0" (zero) is equal to "o"
```

You do not need to handle any other translations apart from those five.

Hint: Conditionals are not necessary for this homework. In fact, it'll be much easier if you don't use them. Instead, check out the methods in this handy Java String API.

Example (user input in bold):

```
Enter string to translate: Emma thinks she is so COOL, but she's not
3mm@ th1nk$ $h3 1$ $0 c00l, but $h3'$ n0t
```

## Closing Comments

Remember that to compile your code from the command line, navigate to the proper directory and use the command `javac FileName.java`. Then run your program with the command `java FileName`.

---

## Allowed Imports

To prevent trivialization of the assignment, you may only import `java.util.Scanner`.

## Feature Restrictions

There are a few features and methods in Java that overly simplify the concepts we are trying to teach or break our auto grader. For that reason, do not use any of the following in your final submission:

- var (the reserved keyword)
- System.exit

## Collaboration

### Collaboration Statement

To ensure that you acknowledge collaboration and give credit where credit is due, **we require that you place a collaboration statement as a comment at the top of at least one java file that you submit**. That collaboration statement should say either:

*I worked on the homework assignment alone, using only course materials.*

or

*In order to help learn course concepts, I worked on the homework with [give the names of the people you worked with], discussed homework topics and issues with [provide names of people], and/or consulted related material that can be found at [cite any other materials not provided as course materials for CS 1331 that assisted your learning].*

Recall that comments are special lines in Java that begin with `//`.

## Turn-In Procedure

### Submission

To submit, upload the files listed below to the corresponding assignment on Gradescope:

- CoinChanger.java
- LeetTranslator.java

Make sure you see the message stating "HW01 submitted successfully". From this point, Gradescope will run a basic autograder on your submission as discussed in the next section.

You can submit as many times as you want before the deadline, so feel free to resubmit as you make substantial progress on the homework. We will only grade your last submission: be sure to **submit every file each time you resubmit**.

### Gradescope Autograder

For each submission, you will be able to see the results of a few basic test cases on your code. Each test typically corresponds to a rubric item, and the score returned represents the performance of your code on those rubric items only. If you fail a test, you can look at the output to determine what went wrong and resubmit once you have fixed the issue.

The Gradescope tests serve two main purposes:

1. Prevent upload mistakes (e.g. forgetting checkstyle, non-compiling code)
2. Provide basic formatting and usage validation

In other words, the test cases on Gradescope are by no means comprehensive. Be sure to thoroughly test your code by considering edge cases and writing your own test files. You also should avoid using Gradescope to compile, run, or checkstyle your code; you can do that locally on your machine.

Other portions of your assignment can also be graded by a TA once the submission deadline has passed, so the output on Gradescope may not necessarily reflect your grade for the assignment.

### Important Notes (Don't Skip)

- Non-compiling files will receive a 0 for all associated rubric items
- Do not submit `.class` files.
- Test your code in addition to the basic checks on Gradescope
- Submit every file each time you resubmit
- Read the "Allowed Imports" and "Restricted Features" to avoid losing points
- Check on Piazza for all official clarifications

---