SAMPLE REFLECTION #1: WORTHY OF FULL MARKS

Name:	Full name here
Student Number:	#########
Email:	userID@myseneca.ca
Section:	XXX
Workshop:	2 (at-home) Reflection

1. During this workshop I learned how to do simple calculations, how to convert from double to int and how to round fractional numbers to the nearest whole number. I also learned how to use division to find how many loonies are in an amount and then how to use modulus to find out how much money is left after removing that number of loonies. I found it interesting to see how the remainder after division could be used to find out how much of something was left after some of it had been removed. I also discovered that I could reuse old variables by storing new values in them without the need to declare new variables.
2. I had never thought of using modulus in this way. In fact, I could not see any use for modulus at the start of the workshop. I found it interesting to see that modulus has uses and that when a calculate an amount modulus n, it gives numbers from 0 to one less than n. I also thought that I needed a new variable for every amount and the amount remaining and it was tough to realize that I could reuse the one I already had.
3. Some quantities, like money, have limited precision. Money has nothing smaller than the penny and calculating more precision can cause the accumulation of fractional cents which, in some cases, can add up to a large fraction of a cent. When these large fractional cents are rounded they can throw the answer off by a cent. To avoid this problem, it is better to convert to cents as soon as possible, storing them as ints so that there will be no fractions to store fractional cents. While it might seem like you are losing a fraction of a cent all the time, this is how it is done with real money. No one ever cuts a penny in half to give you half a cent. They give you a whole cent or none at all.
4. Modulus is a single operation for the computer while dividing and subtracting is at least two operations. This means that using modulus will be faster for the computer. Once you get the expression with modulus written, it can be reused, saving time for the programmer. In fact, using modulus results in shorter expressions which are faster to write and less prone to error.

SAMPLE REFLECTION #2: <u>MINIMUM REQUIRMENTS</u>

```
Name:           Full name here
Student Number: #########
Email:          userID@myseneca.ca
Section:        XXX
Workshop:       2 (at-home) Reflection
```

1. During this workshop I learned how to do simple calculations, how to convert from double to int and how to round fractional numbers to the nearest whole number. I also learned how to use division to find how many loonies are in an amount and then how to use modulus to find out how much money is left after removing that number of loonies.
2. I had never thought of using modulus in this way. In fact, I could not see any use for modulus at the start of the workshop. I found it interesting to see that modulus has uses and that when a calculate an amount modulus n, it gives numbers from 0 to one less than n.
3. Some quantities, like money, have limited precision. Money has nothing smaller than the penny and calculating more precision can cause the accumulation of fractional cents which, in some cases, can add up to a large fraction of a cent. When these large fractional cents are rounded they can throw the answer off by a cent. To avoid this problem, it is better to convert to cents as soon as possible, storing them as ints so that there will be no fractions to store fractional cents.
4. Modulus is a single operation for the computer while dividing and subtracting is at least two operations. This means that using modulus will be faster for the computer. Once you get the expression with modulus written, it can be reused, saving time for the programmer.