# Programming Project 2 – Money on Mars

*Note: When you turn in an assignment to be graded in this class, you are making the claim that you neither gave nor received assistance on the work you turned in (except, of course, assistance from the instructor or teaching assistants).*

The Martians on the planet Mars have a different money system than we do on Earth. Write a program called **MartianMoney.java** that will help new settlers on Mars to better understand the money denominations on Mars. The conversions are as follows:

* 80 dollars = 1 tintina
* 1 tintina = 5 sutton
* 1 sutton = 2 knorr
* 1 knorr = 8 wernicke

The program will first ask the user if they would like to randomly generate a dollar amount or enter it. If the user picks the randomly generated dollar amount, the program will use **Math.random()** to randomly generate a dollar amount between 80 and 10,000,000 inclusively and will display the number generated to the user. If the user wants to enter the number, then the program will ask the user for a dollar amount between $80 and $10,000,000.

The program will then convert the dollar amount into the correct number of tintina, sutton, knorr, and wernicke, just as we would do with our dollars, quarters, dimes, nickels and pennies. It will only display the currencies needed as we do, so if you have 3.75, you would only display 3 dollars 3 quarters. Please use Listing 2.10 on page 63 of the PDF version of our textbook as your guide.

The program should be able to handle incorrect numeric inputs by the user. If the user does not enter a number that is expected the program will state, “Incorrect input”.

Here are some sample runs:

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***2***

**Enter a dollar amount between $80 and $10,000,000.**

***9592906***

**9592906 is 119911 tintina 1 sutton 1 knorr 2 wernicke**

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***2***

**Enter a dollar amount between $80 and $10,000,000.**

***100***

**100 is 1 tintina 1 sutton 4 wernicke**

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***1***

**1046896**

**1046896 is 13086 tintina 1 sutton**

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***25***

**Incorrect input**

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***2***

**Enter a dollar amount between $80 and $10,000,000.**

***74***

**Incorrect input**

**Would you like to randomly enter a dollar amount (1) or enter it yourself (2)?** *(Note this is a single line of output)*

***2***

**Enter a dollar amount between $80 and $10,000,000.**

***345632899***

**Incorrect input**

Note for Gradescope Testing:

* You **cannot use System.exit()** in your program. This is not the way to end the program or exit from decisions and will prevent Gradescope from working.

This and all program files in this course must include a comment block at the beginning (top) of the source code file that contains:

* the Java program name
* project description
* your name
* the date created
* the course number and section

The comment lines should look like this:

/\*

\* Java program name

\*

\* Project description

\* Your name

\* The version date

\* The course number and section

\*/

Before beginning this project, you will document your algorithm as a list of steps to take you from your inputs to your outputs. This algorithm will be due a week before the project submitted in Canvas. This will be graded mostly for completion and returned to you. It will be your responsibility to understand and correct any errors you might have with your algorithm.

Each step of your algorithm will be added as a comment block within your code. You will have the comment block right above the code that performs the actions specified. For example, before your lines of code that ask the user for inputs, you would have a comment block that states what inputs you are requesting from the user.

You will test your program when you run it at the command line / terminal. You need to have a test case for each path through your program. I have given you some samples of tests to run in the examples above.

You will submit the following:

* Java source code file (MartianMoney.java) to Gradescope
* A screenshot showing the running of at least four test cases at the command line / terminal that will test all the major paths through the program to the assignment link in Canvas.

Ask questions about any part of the programming project that is not clear!

# Rubric for Programming Project 2

|  |  |
| --- | --- |
| **Item** | **Points** |
| Algorithm submitted in class on time | 20 |
| Comment block at top of file with specified project information | 5 |
| Comment blocks stating the algorithm step above the code as specified including comment header block | 10 |
| Appropriate choice of variable names | 5 |
| Decisions are properly structured | 10 |
| Program layout and appearance (Coding style is clear and easily understood) | 5 |
| User prompts are correct | 5 |
| Handles incorrect integer input | 10 |
| Output is correct | 20 |
| Screenshot of test cases submitted in Canvas | 10 |
| **Total** | **100** |