# Programming Project 3 – Living 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).*

Mars is not a friendly place to live. There are strong winds that settlers have to deal with. With this in mind, settlers need to build “round” houses, or houses that are in the shape of an octagon. There is no precipitation, so the roof and floors are flat. In order to plan for building their houses, the settlers want a program to calculate the rough estimate of the cost of their house based on the house wall size they pick. The program will be called **MartianHouses.java**. The user will enter their name and then the length of one side, **s**, in terms of feet of the house they wish to build and the current price per square foot. The program will calculate the base cost of the house using the formulas provided below, using the length of the side, s.

,

The program will display the name of the settler and then the house’s total surface area as well as the cost per square foot all on one line. The base cost per square foot is $14.50. Each calculation will be rounded to two decimal places using print formatting, **printf()**.

After the first planet’s calculations are performed and printed, the program should ask the user if they would like to continue. They will type in the word “no” if they wish to end the program. They can enter anything else to continue. If they continue, they will then be prompted to enter a new settler’s name and length of the house side.

Here is a sample run:

**Enter the settler’s name:**

***Caroline Budwell***

**Enter the length of a side of the house:**

***10.0***

**Caroline Budwell has a house surface area of 1,925.69 and cost of $27,922.44**

**Would you like to enter another house? Enter no to exit.**

***yes***

**Enter the settler’s name:**

***Zach Whitten***

**Enter the length of a side of the house:**

***15.5***

**Zach Whitten has a house surface area of 3,808.06 and cost of $55,216.86**

**Would you like to enter another house? Enter no to exit.**

***no***

Note for Gradescope Testing:

* Place your scanner declaration above your loop.
* Use nextLine() for all String inputs.
* Remember that combining .nextLine() and .nextInt() can cause unexpected behavior (Discussed on slide 43 of Chapter 4 powerpoint)
* You should use nextLine().trim() for the last String input of if you would like to continue or not.
* You will also need to use printf() for printing and rounding your values. Note that you will need commas in your numeric output.
* You **cannot use System.exit()** to end your loop. You must use your loop control variable containing the value “no” to end the loop.

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. 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 (MartianHouses.java) to Gradescope
* A screenshot showing the running of at least three 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 3

|  |  |
| --- | --- |
| **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 and use of constants | 5 |
| Program layout and appearance (Coding style is clear and easily understood) | 5 |
| Appropriate use of looping and no System.exit() | 10 |
| Output is correct | 30 |
| Output is properly rounded using printf to two significant digits | 5 |
| Screenshot of test cases submitted in Canvas | 10 |
| **Total** | **100** |