Assignment 1
Due at 5:00 PM on Tuesday, October 6

For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch. The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources. Please read the course Web page for more information on assignment policies and how to organize and submit your work.

Be sure to download the interface file from the course Web page and to follow all the instructions listed in the Style Guide (on the Web page and in the printed package of handouts).

For full marks, it is not sufficient to have a correct program. Be sure to follow all the steps of the design recipe, including the definition of constants and helper functions where appropriate.

Language level: Beginning Student.

Coverage: Module 2.

1. Create a function *bulk-price* that determines the amount of savings possible by buying soil delivered by the cubic yard instead of in bags by the cubic foot. Your function will consume the cost per cubic yard, the cost per cubic foot, the delivery charge, and the total number of cubic yards purchased. The delivery charge is applied to the bulk order only, a flat fee regardless of volume purchased. For example, (*bulk-price* 54 20 10 2) will produce 962 since the cost of the bulk purchase will be $2 \times 54 + 10 = 118$ and the cost of the bag purchase will be $2 \times 20 \times 27 = 1080$. If the bulk cost is higher than the cost of buying soil in bags, the number produced will be negative. Note: there are three feet in one yard.

2. Create a function *drop-units* that consumes an integer 10 or greater and produces the number obtained by removing the units. For example, (*drop-units* 143) will produce 140, (*drop-units* 299) will produce 290, and (*drop-units* 350) will produce 350.

3. Create a function *monogram* that consumes two strings (first and last name), and produces a monogram formed by putting a period after each initial and a space after the first period. For example, (*monogram* "Nelson" "Mandela") will produce the string "N. M.". Do not use the built-in function *string-ref*.

4. Create a function *choc-points* that determines the number of points awarded for selling chocolate bars of three different types. Your function will consume the numbers of bars sold of each type and will produce the number of points you receive (the sum of basic points and bonus points). The basic points are calculated as follows: 1 point for each bar of type 1, 2 points for each bar of type 2, and 5 points for each bar of type 3. The bonus points are based on the total number of bars sold (the sum of the numbers of bars of each type). Two bonus points are awarded for each multiple of ten bars sold *in excess of* the first 100. For example, (*choc-points* 75 50 200) will produce 1219, the sum of $75 * 1 + 50 * 2 + 200 * 5 = 1175$ basic points and $2 * 22$ bonus points, since there are 22 groups of ten bars in the 225 bars in excess of the first 100.