Assignment 1

Due at 9:00 AM on Wednesday, October 3

For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch, including helper functions. Do not copy the purpose directly from the assignment description. The purpose should be written in your own words and include reference to parameter names of your functions.

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.

Do not send any code files by e-mail to your instructors or tutors. They will be not accepted by course staff as assignment submissions.

Test data for all questions will always meet the stated assumptions for consumed values.

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). Specifically, your solutions should be placed in files `a01qY.rkt`, where `Y` is a value from 1 to 4.

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.

Contracts: You are expected to distinguish among the types *num*, *int*, *nat*, *any*, and *string*. See Section 1.6 of the Style and Submission Guide for details. Read each questions carefully for restrictions.

Language level: Beginning Student.

Coverage: Module 2. Use only material from Modules 1–2 in your solutions.

1. A movie theatre gives you a choice of paying for a single ticket or buying a pass that lets you see as many movies as you wish. Create a function *break-even* that consumes the cost *single-ticket* of a single ticket and the cost *pass* of a pass and produces the number of movies you need to see before the cost of a pass is at least as cheap as buying tickets at the single-ticket price. For example, (*break-even* 2.99 12.50) should produce 5, since the cost of four or fewer single tickets is less than 12.50.

2. You are responsible for placing an order for pizzas, where you need to feed both vegetarians and meat-eaters. Each pizza has 8 slices; a hungry person will eat 3 slices but a full person only 1. Create a function *number-pizzas* that consumes the numbers *veg-full* of full vegetarians, *veg-hungry* of hungry vegetarians, *meat-full* of full meat-eaters, and *meat-hungry* of hungry meat-eaters, and produces the smallest number of pizzas you need to order in order to feed everyone. You can assume that vegetarians will only eat vegetarian pizza and meat-eaters will only eat meat pizza. For example, (*number-pizzas* 0 1 1 5) should produce 3, as you will need to order one vegetarian pizza for the one hungry vegetarian (even though 5 of the slices will go to waste) and two meat pizzas for the meat-eaters (one slice for the full eater and three slices for each of the hungry meat-eaters for a total of 16 slices or two pizzas).

3. Create a function *moving-cost* that consumes the number *distance* of kilometres to move and the number *gas-price* of dollars per litre paid for gas (both positive) and produces the total cost of moving. The charges for a move consist of a base cost of 20 dollars per move, plus a charge of 2 dollars per kilometre moved, plus the cost of gas (based on the distance and the cost of gas, assuming that your van can move .5 km using one litre of gas). For example, (*moving-cost* 1 1) will produce 24, computed by adding 20 (the base cost), 2 (the charge for moving one kilometre), and 2 (the cost required to move 1 km at a cost of 1 dollar per litre and gas mileage of .5 km per litre).

4. Create a function *room-cost* that consumes the dimensions *width* and *length* (in metres) of a room and produces the total cost (in dollars) of installing carpet in the room. You should assume that each dimension is a positive number. Carpet costs 10 dollars per square metre. In addition, there is the cost of nailing down the carpet along each of the four edges. The nails cost 1 cent each and are placed along each edge at a distance of 2 cm apart. You should calculate the number of nails on each edge without worrying about the placement of nails on adjoining edges; in particular, it is acceptable to have two nails in each corner. For example, (*room-cost* 2 3) will produce 65.04. The cost of the carpet will be 60 dollars. The two edges of length 2 will require 101 nails each and the two edges of length 3 will require 151 nails each, for a total of 504 nails at a cost of 5.04 dollars.