Comp160
Lab 1
Spring 2019

The goal of the first lab is to get you acclimated to Dr. Racket by having you type and run some BSL code from the Prologue of HTDP2e. Along the way you'll be directed to take a closer look at the code you're interacting with. When you're done, you'll submit print and turn in some annotated BSL code. Read through this document and be sure you know what is expected of you before diving into the work.

Code Comments

Not everything you type in a program document[1] is meant for the computer to read. We'll increasingly put comments for humans alongside the code. To signify that a line is not meant to be read by the machine we place a semicolon before the content of that line. You'll often see me using multiple semi-colons to denote different types of comments. By the end of the Prologue you'll see comments used to help delineate where different parts of the program sit within the document.

One of the most important uses for comments is documenting who wrote the code and when. At the top of every program you write should be a *file header*. This is simply a set of comments listing your name, the date, and a basic description of the file contents. In Figure 1 you can see a basic example of a file header.

```
;; Logan Mayfield
;; 1/15/2018
;; Lab 1
```

Figure 1: Example BSL Racket File Header Comments

Be sure to put a header at the top of the file you'll be turning in for this lab!

Lab 1

You should, eventually, go through the whole prologue carefully. It really lays out where things are going over the coming weeks, the core mechanics of the BSL Racket language, and, most importantly, the basic principles of program design and refinement. However, for this lab you'll focus your attention on the subsection of the prologue titled *Inputs and Output* and the sections that follow it. This portion of the prologue works through the development and refinement of a

[1] The definitions window

program to animate a rocket. When you are directed to write or run some code, do so. It is tempting to copy and paste code from the web browser over to Dr. Racket but I recommend you do not do this or do it sparingly. Typing out the code as you see it in the book gives you a chance to practice typing BSL Racket and seeing how the Dr. Racket environments assists you in doing so. As you work through the lab and the prologue in general, be certain to keep an eye on the following:

- Using the interaction and definitions window
- Running Code
- Proper code formatting (use of spaces, parenthesis, etc.) and the ways in which Dr.Racket helps you maintain that formatting.
- Proper code style (indentation, newlines, etc.) and the ways in which Dr.Racket helps you maintain that style.
- Finding and using the Documentation system

You can keep or remove whatever you'd like from your definitions window but by the end of lab you need to have a definitions window that contains the sixth version of the rocket program just as it appears in Figure 9 of the text. This means only the code in the figure along with the file header as described above. Once you have this, you're ready continue with some analysis and annotation of this program. These annotations will be done by hand so print out your definitions window before proceeding on to the next step.

Identifying Expression Types

The final version of the rocket program is built up from several key types of expressions[2]:

1. function/operator application expressions - expressions instructing the computer to calculate with a function or primitive operator
2. conditional expressions - expressions instructing the computer to calculate different things under different conditions
3. constant definition expressions - expression assigning a name to a fixed value
4. function definition expressions - expression assigning a calculation to an a non-primitive function application expression

Label each expression in the program with the number (as given above) corresponding to its expression type. Record the number of occurrences of each expression type.

[2] things surrounded by parenthesis

Design and Versioning Dissection

There are six versions of the rocket program. Each version either adds or refines a feature of the rocket program. Let's write a concise version history of the rocket program. On the back of your code create a table like the one outlined in Figure 2. Be sure to leave lots of space. Feel free to use another sheet of paper.

version	what	why	how
1			
2			
3			
4			
5			
6			

Figure 2: Version Analysis Table

In the *what* column for the first version, give a brief description of *what* the program does[3]. For versions two through five, in the what column, fill in a description of any newly added or refined features. The why and how columns for version one will be left blank. For versions two through five describe the nature of the improvement in the *why* column. Is the new program better from the user or programmer's perspective? Why? In what way? Finally, in the how column list the programming techniques, ideas, and expressions that enabled the change. What expressions are in the code that weren't there before or perhaps what was removed?

What To Submit

I should receive your printed and annotated version of the sixth version of the rocket program along with your tabular analysis of the code before you leave the lab. If you find that the end of lab is approaching and you're not going to complete everything that was asked of you that's fine. Just wrap up what you're working on and turn it in. It's highly recommended that you complete the lab on your own time before the next class period. It is not required that you do so.

[3] Not how it does what it does, but what the program does. Do not start describing calculations. Focus on a more user/human perspective.