CS 320 Principles of Programming Languages
Homework 1
Due Wednesday 1/17 before class

1.

Comment briefly on the role of "syntax" and "semantics" in each of the following scenarios. [NOTE: We are not looking for very deep answers here, just for evidence that you have a good understanding of the basic concepts. A couple of sentences for each scenario should be sufficient.]

a) The tenant in a recently constructed house uses a voice activated assistant to turn on the lights and to adjust the thermostat temperature in their home. (Think of systems like Amazon Alexa, Apple Siri, Google Assistant, or Microsoft Cortana.)

In this scenario, the syntax of these systems would be that the person using this assistant would have to speak a certain way into the system for it to recognize
certain commands. The voice activated assistant will make sure that the command (syntax) of the user was recognizable and in the proper order. The semantics of the
assistant will be what the assistant will choose to do with the command given. Example of syntax would be "Alexa, set a 10 minute timer" and in response, Alexa would
set the timer which would be the semantics.

b) A person types the address of the Portland State University home page into a browser on a computer at their local public library.

In this particular example, the syntax would be whether or not the person entered the address in correctly. The PSU home page is "www.pdx.edu", so the
syntax would be the key elements that need to be in a link and the order of what is entered. It would need to be entered in the correct format such as "www.pdx.edu",
not something like "edu.pdx.www.", this would be incorrect syntax. The semantics would be when the user hits enter, the website would then open up, or do whatever is
required on that computer to open the website.

c) The IRS allows people to submit the information for their tax returns via an online system. An advantage for taxpayers is that the system gives them a prompt notification if it finds any errors in their return, and then provides an opportunity for them to submit a corrected version.

The syntax of the scenario would be the user entering the correct information. If it says "Enter Name" the correct syntax would be to use alphabetical letters, instead of
numbers or symbols. After the user has entered the correct information, the semantics would be the process of the system checking to make sure the syntax was correct, and
deciding what to do with the results the user entered.

d) An innovative start up is using artificial intelligence to generate two sentence summaries of news articles that are published on major web sites.

The syntax is how the AI will retrieve the data and formalize it into proper readable grammer that people will understand. The semantics would be the process of the AI figuring out what is important to retrieve and generate into the major websites.

2.

For each of the following items, explain what the term refers to and why it would be important and/or relevant in the design of a practical programming language:

a) concrete syntax

Concrete syntax is syntax that can be represented in readable text strings that people understand. So one example of this would be the parenthesized prefix syntax

b) abstraction

Abstraction is a process which makes something easier to understand by ignoring the details that may not be important to the current situation. Abstraction is quite
important in many programming languages because it creates efficiency and reduces complexity, so the program won't have to search through unnecessary data to get
to whatever it needs.

c) static semantics

Static semantics are checked at compile time and are checked to make sure that all of the data types are included. Will check to see if the functions are called in
the proper order. Basically to make sure that everything is declared, assigned, and called correctly. Static semantics are very important in any programming languages
because it will go through and check to see if what was coded even makes sense. After making sure, it will tell you what needs to be corrected and what will work. Without
static semantics, programming would be exponentially harder.

d) dynamic semantics

Dynamic semantics are checked at run time during execution. Dynamic semantics determine when and how the program expressions should behave. Where as regular semantics are
the strategy of how the expressions can be evaluated. This is quite important in a program because this is the fundamental process of how programs are executed and run.

Order and determination of execution is important because in programming code has order and discipline, so it is important for the system to know when and how to execute.

3.

Recall the Prop language of propositional/digital logic whose abstract syntax is described in Haskell by the following definition:

```
data Prop = FALSE | TRUE | IN String | NOT Prop | OR Prop Prop | AND Prop Prop
```

For each of the following Boolean-valued expressions (written C/C++ syntax with capital letters denoting parameters/inputs to the circuit being represented), write the associated Haskell expression of type Prop and draw the corresponding AST:

```
a) !(A || A)
        NOT (OR (IN "A") (IN "A"))
b) !(!A && !!B)
        NOT (AND (NOT (IN "A")) (NOT (NOT (IN "B"))))
c) A && !B || C
        AND (IN "A") (OR (NOT (IN "B")) (IN "C"))
```

4.

Build a circuit using the abstract syntax for Prop to test if two inputs are equal. Justify that your circuit is correct.

Answer is AND (IN "A") (IN "B")

Justify:

if [("A", TRUE), ("B", TRUE)]

```
      AND (IN "A") (IN "B")
=     AND (TRUE) (IN "B")
=     AND TRUE (TRUE)
=     AND TRUE TRUE
=     TRUE
```

if [("A", FALSE), ("B", FALSE)]

```
      AND (IN "A") (IN "B")
=     AND (FALSE) (IN "B")
=     AND FALSE (FALSE)
=     AND FALSE FALSE
=     TRUE
```

if [("A", TRUE), ("B", FALSE)]

```
      AND (IN "A") (IN "B")
=     AND (TRUE) (IN "B")
=     AND TRUE (FALSE)
=     AND TRUE FALSE
```

```
      =     FALSE

      if [("A", FALSE), ("B", TRUE)]

            AND (IN "A") (IN "B")
      =     AND (FALSE) (IN "B")
      =     AND FALSE (TRUE)
      =     AND FALSE TRUE
      =     FALSE
```

5.

Is it possible to construct an expression in the Prop language that could produce an infinite sequence of steps in the normalization procedure described in the lectures? Justify your answer.
It is not possible because if the expressions can be displayed as a binary tree, and the more simplifcations you do, the simpler the solution will come. It's impossible to make it infinite.