Homework 0

Due 7 February

Handout 3
CSCI 334: Spring 2018

Turn-In Instructions

For the first assignment, please use the LaTeX template supplied on the course webpage, print it out, and turn it in by the due date. Note that this assignment is due by Wednesday, February 7. You must either hand it to me in person or slip it under my door in TCL 307.

Subsequent assignments will be turned in electronically using Github. Stay tuned.

Reading

1. **(Required)** Mitchell, Chapters 1–2.
2. **(Recommended)** “Why Undergraduates Should Learn the Principles of Programming Languages”, 2011.

   An overvew of why PL is worth studying and what the course objectives are.

Problems

Q1. (0 points) .... Github Username

If you have a Github username you would like to use for class, please supply it. Otherwise, please follow the instructions at `http://www.cs.williams.edu/~dbarowy/cs334s18/assets/setup-github.html` to create a new account. Do not share your Github password!

Q2. (10 points) .... Partial and Total Functions

For each of the following function definitions, (i) give the graph of the function. Say whether this is a (ii) partial function or a total function on the integers. If the function is partial, say where the function is (iii) defined and where it is (iv) undefined.

For example, take the function $\texttt{f(x)} = \texttt{if x} > \texttt{0 then x} + \texttt{2 else x/0}$

The graph of this function is the set of ordered pairs $\{\langle x, x+2\rangle \,|\, x > 0\}$. The function is partial. It is defined on all integers greater than 0 and undefined on integers less than or equal to 0.

Functions:

(a) $\texttt{f(x)} = \texttt{if x} < \texttt{10 then 0 else f(x} - \texttt{2)}$

(b) $\texttt{f(x)} = \texttt{if x} + \texttt{3} > \texttt{3 then x} + \texttt{4 else x/0}$

(c) $\texttt{f(x)} = \texttt{if sin(x)} > \texttt{0 then 1 else f(x} + \pi\texttt{)}$

Q3. (10 points) .... Deciding Simple Properties of Programs

Suppose I give you a library containing the Java function

```
boolean Halt∅(String f)
```

where calling the function with the source code for a function `f` has the following behavior:

$\texttt{Halt}_\emptyset\texttt{(f)}$	returns true if program `f` will halt when executed.
$\texttt{Halt}_\emptyset\texttt{(f)}$	returns false if program `f` will not halt when executed.

(a) Write a Java function `Halt` that takes a program text `f` (a `String`) and an `int n` as input, and then decides whether `f` halts when it reads `n` as input. The return type of `Halt` should be `void`. `Halt` should print "yes" if `f` halts when run on input `n` and "no" if `f` does not halt when run on input `n`.

You may assume that any `f` read by your program has the following form:

```
void MyFunction(int n) {
    ...
}
```

In other words, all valid forms of `f` read an integer input and returns nothing.

Note that you may use any function available in the Java Class Library that you think will help you. For convenience, you may assume that all valid `f` are named "`MyFunction`". You may also assume that any given `f` is a syntactically correct Java function.

(b) Does your `Halt` function imply that $\texttt{Halt}_\emptyset$ can be used to solve the halting problem? Be sure to review the definition of the halting problem (pp. 14–15, Mitchell).

(c) Why or why not? Explain your answer by describing how your `Halt` function works.

Q4. (2 points) .... Challenge Problem

Recall the definition of a total function:

A function `f(i)` is total if and only if `f` is defined for every input `i`.

Suppose I give you a library containing the Java function

```
boolean Tot(String f)
```

where calling the function with the source code for function `f` has the following behavior:

```
Tot(f)   returns true if f is a total function.
Tot(f)   returns false if f is a partial function.
```

You may assume that the form of `f` is:

```
SomeFunctionName(Input j) {
    ...
}
```

(a) Is `Tot` computable?

(b) Why or why not? For full credit, demonstrate using Java code as in the previous problem. Recall, as in the previous problem, that we know that the halting problem is not computable.

Hint: You may find it convenient to be able to define functions inside of other functions; feel free to abuse Java syntax to achieve this, being sure to explain the meaning of your syntax (or use a Java "lambda expression," if you are feeling extra adventurous).