Steven Meyers

Professor Buffenbarger

CS 354

10/2/2020

1.1

The imperative language being used will be Java.

a. int varName;$
   The $ at the end messes up the token and produces a lexical error.
b. itn varName;
   “int” is misspelled, causing a syntax error
c. undeclaredVar = 4;
   undeclaredVar is never declared before this statement, so at compile time it picks up this semantic error.
d. int newNumber;
   method(newNumber) {
   …
   }
   The method calls newNumber with no definition to it, being detected at run time.
e. public int method() {
   ```
   int value = 3;
   return (value/0);
   ```
   }
   This is completely fine to do but will throw an undefined behavior error in java as dividing by 0 is impossible.

1.8

I would imagine that this form of dependence management would be pretty accurate as there is little harm to be done from it and if something major changes in file B, recompiling A would definitely change how the compiled target code looks like. This may still lead to unnecessary work when things such as simple variable name changes occur or comments are added or removed within B, causing a recompile of A that changes little of how it functions. If your make file is not set up correctly with dependencies, it may not update a file that needs to be recompiled when you change one of its files as it does not deem it dependent.

2.1

A.

“(* non-backslashcharacter*)*”

B. nonopeningChar => all characters but { and “(*”

nonendingChar => all characters but } and “*)”

(* nonopeningChar* *) | { nonopeningChar } | (* nonendingChar *) | { nonendingChar }

C. ((0 (0 | 1 | 2 | 3 | 4 | 5 | 6 | 7) *) | (0x | 0X (digits | (a | b | c | d | e | f) | (A | B | C | D | E | F))*) | (digit* . digit* | digit* e digit* | digit* E digit*)) (U | l | ll | LL | nothing)

I have no clue how to write a RE in this class. I will have to ask the instructor for a lot of clarification next week.