Homework 2

Programming Language Concepts

Due October 30th, 2020

Name: _____

1. (20 points) Given the following Grammar and the right sentential form determine why the string does not go in the language.

   $S \rightarrow AB \mid BC \mid cB$

   $A \rightarrow Ac \mid aBB \mid b$

   $B \rightarrow Cb \mid cBb \mid a$

   $C \rightarrow Ac \mid bCA \mid c$

   a) cbcaaab

   b) cBbaBBccc

   c) aBbcbcBb

2. (20 points) Given the following Grammar and the right sentential form draw a parse tree and show the phrases, simple phrases and handle.

   $S \rightarrow AB \mid BC \mid cB$

   $A \rightarrow Ac \mid aBB \mid b$

   $B \rightarrow Cb \mid cBb \mid a$

   $C \rightarrow Ac \mid bCA \mid c$

   a) aaBcbcba

   b) cBbAbcccc

   c) cbCaBBcb

3. (10 points) Describe the different types of parsers. Give an example of each. What are the benefits and limitations of each. What role do they play in the compilation process.

4. (10 points) What are attribute grammars, operational, axiomatic semantics, denotational semantics. What are they used for in the compilation process.

5. (10 points) Correct the EBNF to force at least one statement for each case or default and then convert the following EBNF into a BNF.

```
<switch_stmt> -->
   switch"("(<expr>|<identifier>) ")""{" <body> "}"
<bod> -->
   case<literal>:{<stmt>;}{case<literal>:{<stmt>;}}[default:{<stmt>;}]
```

6. (15 points) Design a state diagram to recognize one floating point literals in C.

```
3.14159       /* Legal */
314159E-5L    /* Legal */
510E          /* Illegal: incomplete exponent */
210f          /* Illegal: no decimal or exponent */
.e55          /* Illegal: missing integer or fraction */
```

7. (15 points) Design a state diagram to recognize one floating point literals in Go-Lang.

```
0.
72.40
072.40       // == 72.40
2.71828
1.e+0
6.67428e-11
1E6
.25
.12345E+5
1_5.         // == 15.0
0.15e+0_2    // == 15.0

0x1p-2       // == 0.25
0x2.p10      // == 2048.0
0x1.Fp+0     // == 1.9375
0X.8p-0      // == 0.5
0X_1FFFP-16  // == 0.1249847412109375
0x15e-2      // == 0x15e - 2 (integer subtraction)

0x.p1        // invalid: mantissa has no digits
1p-2         // invalid: p exponent requires hexadecimal mantissa
0x1.5e-2     // invalid: hexadecimal mantissa requires p exponent
1_.5         // invalid: _ must separate successive digits
1._5         // invalid: _ must separate successive digits
1.5_e1       // invalid: _ must separate successive digits
1.5e_1       // invalid: _ must separate successive digits
1.5e1_       // invalid: _ must separate successive digits
```


_____ 30 points