Tutorial Questions for CSC 309

1) Explain how a variable can be allocated and de-allocated in C++
2) Three extensions are commonly included in various versions of EBNF; state and explain them.
3) In most contemporary languages, parameter communication takes place through the run-time stack. State and explain the five implementation models of parameter passing.
4) Multiple selection construct allows the selection of one of any number of statements or statement groups. Write and discuss the general switch-case statement in C, C#, java and javascript. How does switch statement in C# differ from C-based language?
5) In the following grammar:

$$
\begin{aligned}
\langle assign \rangle &\longrightarrow \langle id \rangle = \langle expr \rangle \\
\langle id \rangle &\longrightarrow A/B/C \\
\langle expr \rangle &\longrightarrow \langle expr \rangle + \langle term \rangle \\
&\quad \mid \langle term \rangle \\
\langle term \rangle &\longrightarrow \langle term \rangle * \langle factor \rangle \\
&\quad \mid \langle factor \rangle \\
\langle factor \rangle &\longrightarrow (\langle expr \rangle) \\
&\quad \mid \langle id \rangle
\end{aligned}
$$

Show a parse tree and leftmost derivation for each of the following statements:

I. A = B + C * A
II. B = A*C+B
III. A = A*B+C

6) Differentiate between static type binding and dynamic type binding
7) Write a Common Lisp program that clearly shows the difference between static and dynamic scoping.

8) Scope and lifetime are sometimes related but are different in concepts, explain the scope and lifetime of variable sum in function printheader in the following C++ program.

```
Void printheader() {
……
}  /* end of printheader */
Void compute () {
int sum;
.   .   .
Printheader ();
}   /*  end of compute */
```

9) Consider the following grammar over the terminals +,−and id:

$$
\begin{aligned}
S &\longrightarrow E \\
E &\longrightarrow E{+}E/{-}E/id
\end{aligned}
$$

Draw all the parse trees for the string id+-id + id. Is this grammar ambiguous? Why or why not?

10) What makes the for statement in java and C# different from C++?

11) Compute the weakest pre-condition and post conditions for each of the following assignment statements:

I. if (x>0) then
   y = y – 1;
   else
   y = y +1;
   {y>0}

II. a = 2 *(b-1) -1 (a>0)

12) In the following grammar :

$$<assign> \longrightarrow <var> = <expr>$$
$$<expr> \longrightarrow <var>[2] + <var>[3] \mid <var>$$
$$<var> \longrightarrow A \mid B \mid C$$

I. Compute attribute values using appropriate parse tree for A=A+B

II. Indicate the flow of attributes for the tree generated in I.

13) What is variable initialization? Write a scheme program that returns the value of the expression: (a+b)/(c-d)

14) Write test programs in C++ to determine the scope of a variable declared in a for statement. Specifically, the code must determine whether such a variable is visible after the body of the for statement.

15) If we write a code in C as

```
int a,b;
float c;
………
C = a/b;
```

In the above codes, explain how coercions are done during compilation stage.

16) Write short notes on the following terms:

I. Operational Semantics
II. Denotational Semantics
III. Axiomatic Semantics

17) How does C support relational and Boolean expressions?

18) Would it be a good idea to eliminate all operator precedence rules and require parentheses to show the desired precedence in expressions? Why or why not

19) Show the order of evaluation of the following expressions by parenthesizing all sub-expressions and placing a superscript on the right parenthesis to indicate order.

I. p + q * r + s
II. p* q - 1 + s

20) Consider the following C program:

```
int b = 11;
int fun1() {
b = 19;
   return 7;
 } /* end of fun1 */
 void main() {
  b = b + fun1();
} /* end of main */
```

The value computed for b in main depends on the order of evaluation of the operands in the expression b+ fun1(). What is the value of b?

i. if b is evaluated first
ii. if the function call is evaluated first.