Midterm Review (Chapters 3 – 6)  
Questions and Answers taken from open-quizzes:  
  
**Chapter 3**1. Given the following grammar and the following sentential form, what would the next sentential form be if you used the left-most derivation?  
  
<A> -> a<A> | b<A> | c<B>  
<B> -> <C><A> | b<C>  
<C> -> d  
  
Input: abcdacbd

Current: abc<C><A>  
Answer: abcd<A>

2. Given the following grammar and the corresponding input string, what would the parse tree look like?  
  
<A> -> a<B> | a<A>

<B> -> b<B> | b<C> | b<A>  
<C> ->  
  
Input: abbbc  
Answer:  
Chart, scatter chart

Description automatically generated

3. The key difference between a left-most and a right-most derivation is that:  
Answer: Given a sentential form that contains multiple non-terminals, a left-most derivation always picks the left-most non-terminal to reduce next, while a right-most derivation always picks the right-most non-terminal to reduce next.

4. A grammar is considered ambiguous if:

Answer: It is possible to generate a sentential form that has more than one distinct parse tree.

5. A Token:

Answer: is a category or name of a group of characters that comprise the smallest unit of a programming language.

6. A Lexeme   
Answer: is a group of characters that comprise the smallest unit of a programming language

7. A grammar can be written in some type of BNF notation. There are two types of symbols in BNF expressions. Non-terminal symbols are typically:

Answer: Abstract concepts that need to be further defined.

8. A grammar can be written in some type of BNF notation. There are two types of symbols in BNF expressions. Terminal symbols are typically:

Answer: The symbols that would appear in a source code file.

**Chapter 4**1. A bottom up parser:

Answer: Produced a parse tree that is in the order of a right-most derivation. It examines the input from left-to-right.

2. How would you resolve the pairwise disjointness problem in the grammar below?

Text

Description automatically generated with medium confidence

Answer:

Text

Description automatically generated

3. Given the following parser table, state of the stack and input string, if you were to read in the next symbol, what action would you take?

Calendar

Description automatically generated

Answer:

Add id to the stack

Add [1] to the stack

Advance the input string so that the . is after the first id symbol

Transition to state [1]

Reduce by rule [3]

And pop one pair of symbols off the stack

4. What is the purpose of a lexer?

Answer: A lexer scans an input file and identifies or associates tokens with groups of characters that match a particular token category

5. A top down parser:

Answer: Produces a parse tree that is in the order of a left-most derivation. It examines the input from left-to-right.

6. A bottom up parser:

Answer: Produces a parse tree that is in the order of a right-most derivation. It examines the input from left-to-right.

7. How would you remove the left-recursion from the following grammar?

<A> -> <A>a<B> | <A>bb<A> | <B>a | <C><B><A>

<B> -> b<C> | <C>

<C> -> c

Answer:

Text

Description automatically generated with low confidence

8. How would you remove the direct left-recursion from the following grammar?

<A> -> <A>a<B> | <B>a | <C><B><A>

<B> -> b<C> | <C>

<C> -> c

Answer:

A picture containing text

Description automatically generated

9. Give the following grammar and the closure for the starting state, what would the next closure or state be if the next input symbol was “ ( “ ?

A picture containing diagram

Description automatically generated

**Chapter 5**1. What does it mean to say that a **type** (data type) biding is dynamic?

Answer: The type is first bound during executing, and can change during execution of the program, usually via an assignment statement.

2. When and where are storage bindings created for Stack-Dynamic variables?

Answer:   
When: Run-time, when the declaration statement is elaborated

Where: On the stack

3. What does it mean to say that a variable attribute is statically bound?

Answer: The attribute is bound during compile time, and remains bound during the entire execution of the program.

4. What is the referencing environment for the **block of code** within the function **myf2()(highlighted below)** using statis scoping?

Graphical user interface, text, application

Description automatically generated

5. What are the **lifetimes** of the variables **x**, **y**, and **z**?

Graphical user interface, text, application

Description automatically generated

**Chapter 6**1. What is the key distinctions between a primitive data type and other data types?

Answer: Primitive data types are not defined in terms of other data types.

2. What is the key distinctions between Rectangular and Jagged Arrays?

Answer: With rectangular arrays every row has the same number of elements and every column has the same number of elements.

With jagged arrays there can be rows or columns with varying numbers of elements.

3. Given the following array index of a 2-based array implementation (indexes start at 2), representing an array that has the dimensions 10x10 (10 rows of 10 elements each), what is the linear (1D) address or index (offset from start address)?

*Assume element size (size of data type is 1 byte).*

**Array[4][5]**

Answer: 23

4. Given the following array index of a 3-based array implementation (indexes start at 3), representing an array that has the dimensions 20x20 (20 rows of 20 elements each), what is the linear (1D) address or index (offset from start address)?

*Assume element size (size of data type is 1 byte).*

**Array[4][5]**

Answer: 22

5. Given the following array index of a 5-based array implementation (indexes start at 5), representing an array that has the dimensions 20x20 (20 rows of 20 elements each), what is the linear (1D) address or index (offset from start address)?

*Assume element size (size of data type is 1 byte).*

**Array[6][7]**

Answer: 22

To do the above 3 questions, take the elements in the array and subtract the #-base.

So [6 – 5] = 1 and [7-5] = 2

Then multiply the left-most result by the dimension (20) and add the right-most result.

Work:  
[6-5]\*20 + [7-5]

1 \* 20 + 2   
20 + 2 = 22

**Chapters 7 - 12**

Graphical user interface, application

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email, Teams

Description automatically generatedGraphical user interface, text, application, email

Description automatically generated