

Homework #1

Due Date: Thursday, October 6, 2005, 2:00

Your Name: _____

Your Email: _____

1. What is the input to a compiler called? _____

2. What is the output called? _____

3. What are the phases in a typical compiler?

4. Lexical Analysis breaks the input into what? _____

5. What kind of grammars are we concerned with?

6. The symbol table generally has an entry for each ... what?

7. What phase is type-checking done in? _____

8. Is the code generated in the intermediate code generation phase target-machine-specific or target-machine-independent? _____

9. When the compiler detects an error in the source, should it halt after printing a message?

10. A compiler will not normally detect logical errors (i.e., program bugs). What 3 kinds of errors will a compiler catch?

11. What is a lexeme?

12. What are blanks, tabs, and newlines called? _____

13. Are comments passed to the parser? Why or why not?

14. For each token, the lexer passes 2 pieces of info to the parser. What are they?

15. For what types of token would additional information be needed?

16. Some tokens (like “if”, “while”, “class”) are built into the language. What are these tokens called? _____

17. Here is a regular expression describing identifier tokens:

Letter { Letter | Digit }*

Describe the tokens specified by this regular expression:

Digit { Letter }* Digit

18. How does a “string table” differ from a “symbol Table”?

19. Why would we enter keywords into a string table?

20. Would it be faster to use a hash-table or an array to implement a string table?

21. Can the same ID have more than one meaning in a program?

22. In a CFG, what does epsilon mean? Please do not say “nothing”.

23. What is the Greek letter for epsilon? _____

24. In a parse tree, the internal nodes correspond to what... Terminals or Non-terminals?

25. In a parse tree, the leaves correspond to what... Terminals or Non-terminals?

26. Here is a grammar

$A \rightarrow A b$

$\rightarrow c$

Draw a parse tree for the string “c b b b”

28. What is an ambiguous grammar?

29. Is it a good idea for programming languages to use ambiguous grammars?

30. The expression $x-y-z$ means either $(x-y)-z$ or $x-(y-z)$. This is a question of operator... what? _____

31. The expression $x+y-z$ means either $(x+y)-z$ or $x+(y-z)$. This is a question of operator... what? _____

32. Is this grammar rule left- or right-recursive?

$A \rightarrow A b$

33. What is the input to a parser (in the context of compiling)?

34. What is the input to a parser generator? _____

35. Can we construct a parser which will parse / recognize any CFG? _____

36. What are the two major approaches to parsing?

37. We can create a parser by constructing a routine for each non-terminal. This approach is called... what? _____

38. Which is a problem in this approach... left- or right-recursion in the grammar rules?

39. Compared to a parse tree, is an abstract syntax tree larger or smaller?

40. When we implement an AST in Java, we'll use a single ...what?... to represent each *kind of* node in the AST. _____

41. When we construct a specific AST, we'll use a single ...what?... to represent each node. _____