1. Why is it a bit of a misnomer to say that a language is an 'interpreted' rather than a 'compiled' language? Explain more precisely what is meant when someone says X is an interpreted language.

2. What is a virtual machine (in the sense of the Java Virtual Machine)? And what role does the just-in-time (JIT) compiler play in the virtual machine?

3. List six compilation phases that might be found in a typical optimizing compiler. Pick one and briefly describe what happens during that phase.

4. List three optimizations that can be applied during compilation. Briefly explain what one of them does.

5. * Consider a Java program formed by combining some arbitrary, but known, class B with the follow Java class A:

   ```java
   public class A {
       public static void main(String[] args) {
           try {
               B.f();
           } catch (Throwable e) { }
           System.out.println("The answer to the ultimate "+ "question of the life the universe and everything "+ "is 42");
       }
   }
   
   Suppose you want a method to determine whether the program will reveal that 'The answer to the ultimate question of the life the universe and everything is 42'. Is it possible to write a program to accomplish this task by examining the code in class B? If so, outline how you would do it. If not explain why this is not possible.

6. Order the following types of languages/grammars according to the Chomsky Hierarchy: context-free, context-sensitive, recursively enumerable, and regular. That is, rank them from most general (most difficult to parse) to least general (easiest to parse).

7. Languages from which of these classes can be described using a BNF grammar?

8. What term is given to a grammar that has more than one parse tree for some well-formed formula?

9. Consider the following BNF grammar for arithmetic expressions:

   ```
   <expr> ::= <expr> + <expr> | <expr> - <expr> | <expr> * <expr> 
   | ( <expr> ) | <num>
   <num> ::= 0 | 1 | 2 | 3 | 4 | ...
   ```

   For each of the arithmetic expressions:
   (a) 1 + 1  (b) 1 + 1 * 2  (c) (1 + 1) * 2  (d) 5 - 3 - 2
i. Give two distinct derivation showing how to produce that expression from the start symbol $<\text{expr}>$.

ii. Draw a parse tree showing how to produce that expression from the start symbol $<\text{expr}>$.

iii. Is there more than one distinct parse tree for that expression? If so draw a second parse, and indicate which—if any—of your parse trees are correct according to arithmetic’s standard rules of precedence (i.e., multiplication has higher precedence than addition) and associativity (i.e., subtraction is left associative).

10. What are lex and yacc? How are they different?

Note: If you collaborated with your classmates or used their notes, please note which classmates you collaborated with. If you use an external source, besides the text book, lectures, notes provided by the instructor, and your own intellect, please cite that source. Use quote marks if you are quoting material word-for-word from any source (including the text book).