

1. Ambiguity in Grammars

When is a context-free grammar ambiguous?

Is the following grammar (given in terms of its production rules) ambiguous? Justify your answer. (Note that $\langle E \rangle$ is a nonterminal, while $(,)$, and y are terminals.)

$$\langle E \rangle \quad :: = \quad \langle E \rangle \quad \langle E \rangle \quad | \quad (\quad \langle E \rangle \quad)$$
$$\langle E \rangle \quad :: = \quad y$$

What is the associativity of '<<' in C++? Justify that design decision.

2. PL Concepts

- A. What is name/operator overloading? Illustrate with an example.
- B. What is dynamic binding? What is its significance in an OOPL?
- C. What are the software engineering benefits of inheritance?
- D. What is information hiding?
- E. Automatic memory management / garbage collection is said to contribute to abstraction and reliability of a programming language. Justify this claim.
- F. What is closure?
- G. Why does FORTRAN ban recursion even though it is commonly used for defining mathematical/combinatorial functions?
- H. What are the benefits of type checking?
- I. Give an example of symbolic computation?

J. What are the advantages of using object-oriented programming paradigm over procedural paradigm?

3. Consider the following method.

```
/** @param x is an int and x >= 0 */  
public int mystery(int x) {  
    System.out.print(x % 10);  
    if ( (x / 10) != 0) mystery(x / 10);  
    System.out.print(x % 10);  
}  
}
```

Hand trace the code for the call `mystery(1234)` and then verify your results by running (a suitably modified version of) your code.