Question and Answers Document

1. What are error recovery techniques in top down parsing?  
  
Top-down parsing is a recursive descent parsing technique that is used to parse a string of characters into a parse tree. The parser starts at the top of the grammar and tries to match the next character in the string to the start symbol of the grammar. If the match succeeds, the parser recursively parses the rest of the string. If the match fails, the parser tries to recover from the error by backtracking and trying a different rule.  
  
There are several error recovery techniques that can be used in top-down parsing. One technique is to simply ignore the character that caused the error and continue parsing. Another technique is to try to find a rule that can match the character that caused the error. If no rule can match the character, the parser can report an error.  
  
2. What is Boot ‘Strapping?  
  
Boot ‘strapping is the process of initializing a computer system so that it can start running programs. The first step in boot ‘strapping is to load the BIOS (basic input/output system) into memory. The BIOS is a small program that is stored on a chip on the motherboard. The BIOS initializes the hardware and then loads the operating system into memory.  
  
The operating system is the software that controls the computer. It manages the hardware, provides a user interface, and runs programs. Once the operating system is loaded, it can start running programs.  
  
3. What is the difference between Top Down parsing and Bottom up parsing?  
  
Top-down parsing is a recursive descent parsing technique that is used to parse a string of characters into a parse tree. The parser starts at the top of the grammar and tries to match the next character in the string to the start symbol of the grammar. If the match succeeds, the parser recursively parses the rest of the string. If the match fails, the parser tries to recover from the error by backtracking and trying a different rule.  
  
Bottom-up parsing is a non-recursive parsing technique that is used to parse a string of characters into a parse tree. The parser starts at the bottom of the grammar and tries to match the next character in the string to the right-hand side of a production rule. If the match succeeds, the parser recursively parses the rest of the string. If the match fails, the parser tries to recover from the error by backtracking and trying a different rule.  
  
4. What are ambiguous grammars? Give an example.  
  
An ambiguous grammar is a grammar that can have more than one parse tree for a given string. For example, the following grammar is ambiguous:  
  
```  
S -> NP VP  
NP -> Det N  
VP -> V NP | V  
Det -> the | a  
N -> man | woman  
V -> saw | ate  
```  
  
The string "the man saw the woman" can be parsed as follows:  
  
```  
S -> NP VP  
NP -> Det N  
Det -> the  
N -> man  
VP -> V NP  
V -> saw  
NP -> Det N  
Det -> the  
N -> woman  
```  
  
or as follows:  
  
```  
S -> NP VP  
NP -> Det N  
Det -> the  
N -> woman  
VP -> V NP  
V -> saw  
NP -> Det N  
Det -> the  
N -> man  
```  
  
5. What is Yacc? Explain the Syntax.  
  
Yacc is a parser generator that is used to generate parsers from grammars. Yacc is a recursive descent parser generator, which means that it generates parsers that use a recursive descent parsing algorithm.  
  
The syntax of Yacc grammars is as follows:  
  
```  
%token <token-name>  
%left <production-name>  
%right <production-name>  
%nonassoc <production-name>  
  
<start-symbol> : <production>  
<production> : <non-terminal> | <non-terminal> <production>  
<non-terminal> : <terminal> | <non-terminal> <terminal>  
<terminal> : <string>  
```  
  
The `%token` directive defines a token. The `%left` and `%right` directives define the associativity of a production. The `%nonassoc` directive defines a production that is non-associative.  
  
The `<start-symbol>` is the name of the start symbol of the grammar. The `<production>` is a production rule. The `<non-terminal>` is a non-terminal symbol. The `<terminal>` is a terminal symbol.  
  
6. Define shift-reduce conflict and Reduce-reduce conflict.  
  
A shift-reduce conflict is a conflict that occurs when the parser can either shift a token onto the