Parse & Syntax Trees

Syntax & Semantic Errors

Mini-Lecture

**Parse Tree**
- Expression: \( a[index] = 4 + 2 \)
- Identify tokens:
  - `a` identifier
  - `[` left bracket
  - `index` identifier
  - `]` right bracket
  - `=` assignment operator
  - `4` number
  - `+` plus sign
  - `2` number

**Syntax Tree**
- Also called “Abstract Syntax Tree” or AST
- Expression: \( a[index] = 4 + 2 \)
- Condensed version of Parse Tree
- Excludes redundant information

**Syntax Errors**
- **Syntax error** - an error in the syntax (the rules of formation) of a sequence of characters or tokens that is intended to be written in a particular programming language.
- Example is: entering an invalid equation into a calculator, such as opening brackets without closing them, or several decimal points in a number.
Syntax Errors

• In Java, the following is a syntactically correct statement:
  ```java
  System.out.println("Hello World");
  ```

• while the following is not:
  ```java
  System.out.println("Hello World");
  ```

Semantic Errors

• Semantic error: Writing a valid programming structure with invalid logic.
• The compiler will generate instructions that the computer will execute, because it understands the syntax of the programming statements, but the output will not be correct.

Semantic Errors

• Non-initialized variable:
  ```java
  int i;
  i++;
  ```

• Type incompatibility:
  ```java
  int a = "hello";
  ```