



Saint Vincent College

Department of Computing
and Information Science



Java Tutorial III and Syntax Case Study: Parsing English

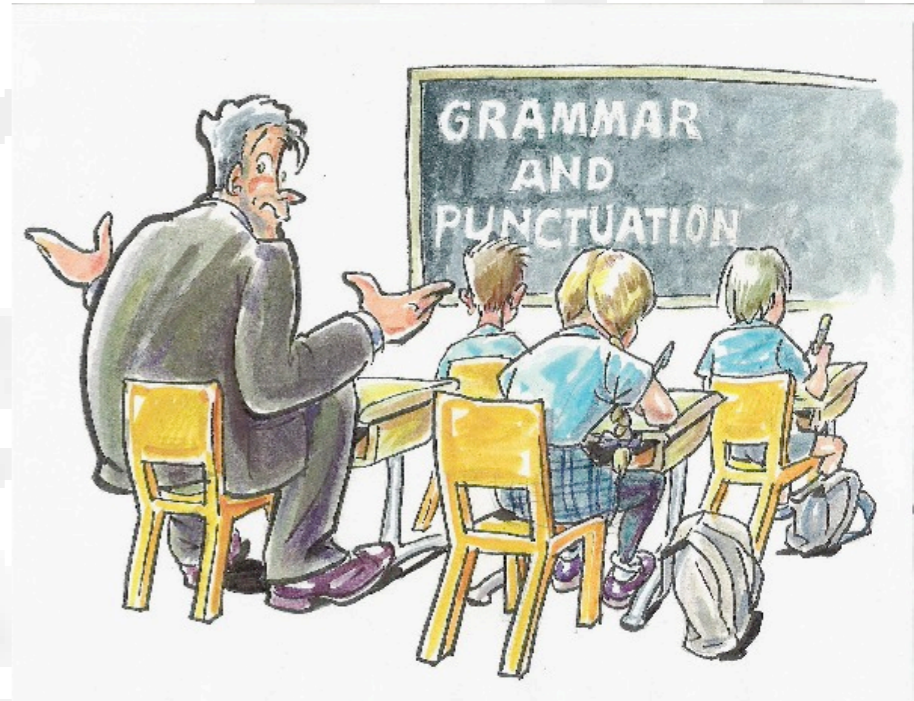
Fr. Boniface, OSB
Lecture 5

(some slides modified from Tucker/Noonan, copyright McGraw-Hill, 2006)

“Could you not stay awake with me one hour?” (Mt 26:40)

Parsing English Grammar

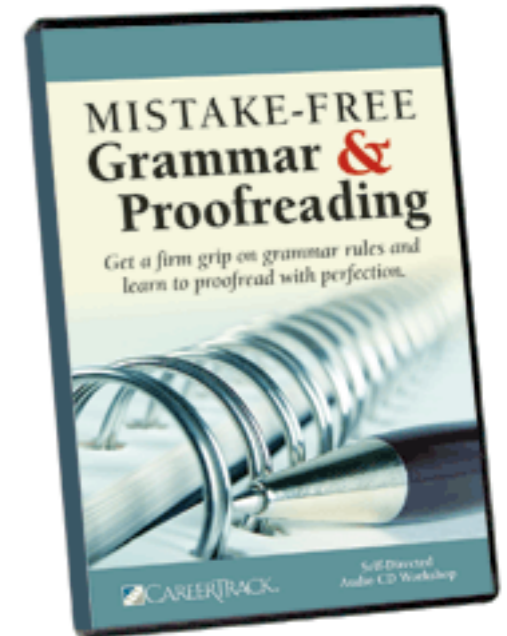
- What is the grammar for English?
- How do we define it formally?
- How do we store the information from parsing English (i.e. Abstract Syntax Tree or parse tree)
- How do we encode these things in Java?



Parsing English



- Sentence ::= Question | Statement | Imperative
- Statement ::= NounPart Verb [NounPart]
- Question ::= Verb NounPart [NounPart] |
WWord Verb NounPart
- Imperative ::= Verb NounPart
- NounPart ::= Noun | Article Noun
- Article ::= 'the' | 'a' | 'an'
- Noun ::= 'fish' | 'teacher' | 'saint'
- Verb ::= 'teaches' | 'swims' | 'serves'
- WWord ::= 'why' | 'when' | 'where' | 'which' | 'what'



Abstract Syntax



```
abstract class NounPart{
}
class Noun extends NounPart {
    Article art;
    String noun;

    public Noun(String noun) {
        this.art = null;
        this.noun = noun;
    }
    public Noun(Article art, String noun) {
        this.art = art;
        this.noun = noun;
    }
    /** This method turns the Noun into a String
     * @return The noun String with article if it has one or just the noun.
     */
    //...code goes here
}
```

Abstract Syntax

```
abstract class NounPart{
}
class Noun extends NounPart {
    Article art;
    String noun;

    public Noun(String noun) {
        this.art = null;
        this.noun = noun;
    }
    public Noun(Article art, String noun) {
        this.art = art;
        this.noun = noun;
    }
    /** This method turns the Noun into a String
     * @return The noun String with article if it has one or just the noun.
     */
    //...code goes here
}
```

```
public String toString() {
    String output = "";
    if (this.art != null) {
        output += this.art + " ";
    }
    output += this.noun;
    return output;
}
```

Abstract Syntax cont..



```
class Pronoun extends NounPart {
    String pronoun;

    public Pronoun(String pronoun) {
        this.pronoun = pronoun;
    }

    public String toString() {
        // what goes here??
    }

    /**
     * Checks whether the pronoun
     * is accusative
     * @return true if accusative,
     *         false otherwise
     */
    // ... method code goes here ...
}
```



Abstract Syntax cont..

```
class Pronoun extends NounPart {
    String pronoun;

    public Pronoun(String pronoun) {
        this.pronoun = pronoun;
    }

    public String toString() {
        // what goes here??
    }

    /**
     * Checks whether the pronoun
     * is accusative
     * @return true if accusative,
     *         false otherwise
     */
    // ... method code goes here ...
}
```

```
public boolean isAccusative() {
    if (this.pronoun.equals("him") ||
        this.pronoun.equals("her") ||
        this.pronoun.equals("me") ||
        this.pronoun.equals("it") ||
        this.pronoun.equals("us") ||
        this.pronoun.equals("them") ||
        this.pronoun.equals("you"))
        return true;
    else
        return false;
}
```



Abstract Syntax cont...

```
class Article {  
    // Article = a | an | the  
    final static Article THE = new Article("the");  
    final static Article A = new Article("a");  
    final static Article AN = new Article("an");  
  
    private String id;  
  
    private Article (String t) { id = t; }  
  
    public String toString ( ) { return id; }  
  
}
```

Note: constructor is private.

Question: How to make new articles?



Abstract Syntax cont...

```
class Article {  
    // Article = a | an | the  
    final static Article THE = new Article("the");  
    final static Article A = new Article("a");  
    final static Article AN = new Article("an");  
  
    private String id;  
  
    private Article (String t) { id = t; }  
  
    public String toString ( ) { return id; }  
  
}
```

Note: constructor is private.

Question: How to make new articles?

Answer: You can't -- you can only use the ones defined in the class:

```
Noun noun = new Noun(Article.THE, "man");
```

Superclass: Sentence

- Sentence is a superclass, intended to be extended for particular kinds of sentences.
- Its only member data is “verb”, because we know every sentence has a verb.

```
abstract class Sentence {  
    Verb verb;  
  
    public Sentence(Verb verb) {  
        this.verb = verb;  
    }  
}  
  
class Statement extends Sentence {  
    NounPart subject;  
    NounPart object;
```

A Story



- A Story is an ordered list of sentences
- What Collection to use?
- How to display a sentence?



A Story

```
class Story extends ArrayList<Sentence>{
```

```
}
```

- A Story is an ordered list of sentences
- What Collection to use?
- How to display a sentence?



A Story

```
class Story extends ArrayList<Sentence>{  
    void display() {  
        for (Sentence s : this)  
            System.out.println(s);  
    }  
}
```

- A Story is an ordered list of sentences
- What Collection to use?
- How to display a sentence?

