

# CHRONOS



# Team Intro



- ❧ Stephen Somuah – Project Manager
- ❧ Shannon Lee – Language & Tools Guru
- ❧ Lishan Amde – Tester & Validator
- ❧ Richard Nwaobasi – System Integrator
- ❧ Tikue Anazodo – System Architect

# Language Evolution



- ❧ Etude
- ❧ Chronos Version 1.0
- ❧ Chronos Version 2.0

# CHRONOS Intro



# Chronos Goals & Features



- ❧ Ease of Re-usability
- ❧ Ease of Course Schedule Generation
- ❧ Flexible
- ❧ I/O Based

# Chronos Tutorial

## Basic Program



```
schedule sched;
courselist CL = generateCourses(input_file);
foreach course c in CL {
    if (c.getStartTime() > 10:00) { // User doesn't want to
        //wake up early for class
        sched.add( c ); // Add whatever section of c
                        //that works
    } else{}
}
print(sched);
```

## Adding Constraints

---

```
int courseMax = 6; // maximum number of courses
int courseMin = 4; // minimum
int creditMax = 20; // maximum number of credit points
int courseMaxR = 2; // maximum number of courses on Thursday

foreach course c in CL
{
    if (sched.numCourses() < courseMax
        && sched.numCredits() + c.numCredits() <= creditMax)
    {
        sched.add( c );
    } else {}
}
```

## Adding Groups



```
courselist required;  
required.add( CL.getCourse("Math") );  
required.add( CL.getCourse("ArtHum") );  
required.add( CL.getCourse("PLT") );  
required.add( CL.getCourse("Artificial Intelligence") );
```

```
/* Create a list of elective classes */
```

```
courselist electives;  
electives.add( CL.getCourse("PE") );  
electives.add( CL.getCourse("Chinese") );  
electives.add( CL.getCourse("Stat") );  
electives.add( CL.getCourse("German") );
```



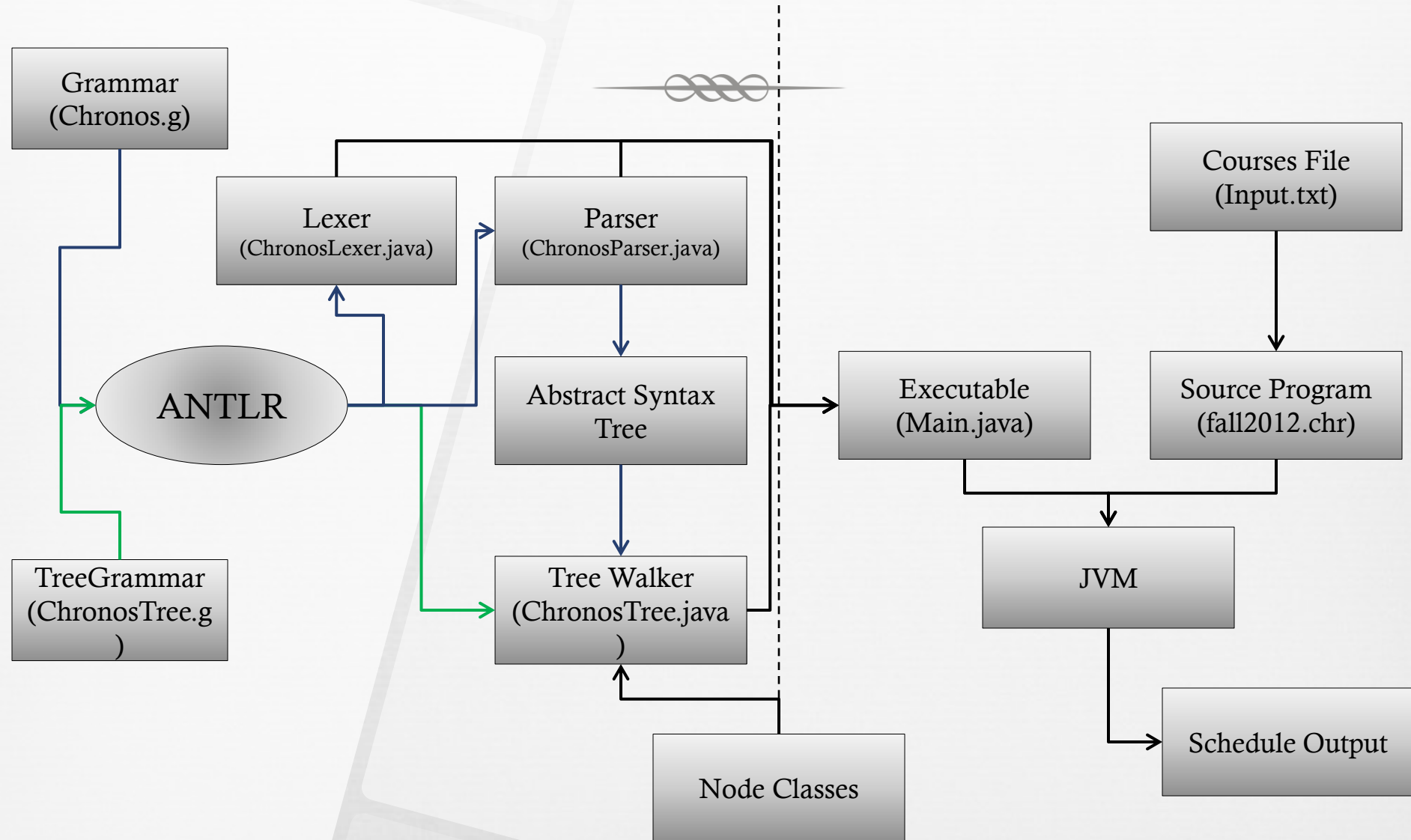
## Using Groups



```
foreach course c in required {
  if (sched.numCourses() <= 3)
    {
      sched.add( c );
    } else{}
}
```

```
foreach course c in electives {
  if (sched.numCredits() <= 20) {
    sched.add( c );
    print(c.name + " successfully added");
  }
} else{}
}
```

# Chronos Architecture

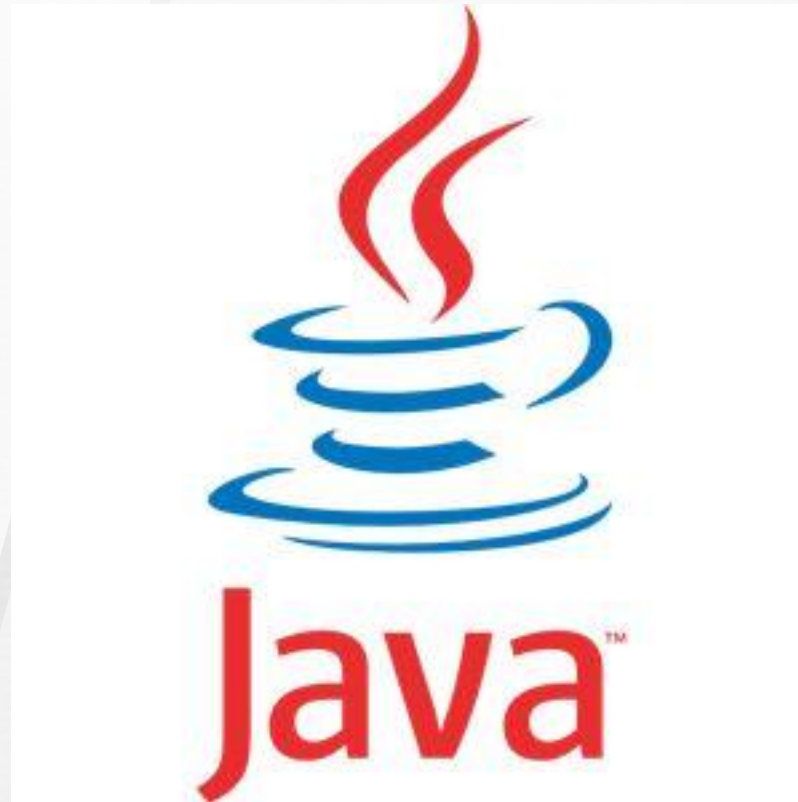


# Chronos Implementation



- ❧ Antlr – Takes Chronos.g & ChronosTree.g
  - ❧ Generates:
    - ❧ ChronosLexer.java
    - ❧ ChronosParser.java
    - ❧ ChronosTree.java
- ❧ ChronosParser.java Generates AST
- ❧ Semantic Checking in Tree Walker

# Code Generation



# Team Interaction



- ❧ Github
- ❧ Dropbox
- ❧ GoogleDocs
- ❧ Email
- ❧ Weekly Meetings

# Team Member Responsibilities



- ❧ Stephen
  - ❧ General overseer
  - ❧ Connecting the dots
  
- ❧ Shannon & Lishan
  - ❧ ANTLR, Lexer & Parser
  - ❧ Semantic Analysis
  
- ❧ Richard & Tikue
  - ❧ Linking with Semantic Analysis
  - ❧ Java Nodes, ANT Build Script

# Challenges



- ❧ Finalizing language
  - ❧ Went through 3 different language ideas
- ❧ Deciding on which language to compile to
- ❧ Deciding on compiler tools to use

# Advice To Future Teams



- ❧ Start early!
- ❧ Talk to class alumni
- ❧ Get to know your teammates
- ❧ Get help from the professor and TAs when needed
- ❧ Start small



Thank you!



Questions?