

# CS335: Milestone 1 Report

Chitwan Goel  
Roll No. 210295

Shrey Bansal  
Roll No. 210997

Talin Gupta  
Roll No. 211095

## Code Base Structure

## Requirements

To run the program, you will need `bison`, `flex` and `graphviz`, the latest versions.

## Usage

We have provided a script to run the program in the `scripts/` directory. The output `dot` file and `pdf` file will be generated in the `src/` directory. Run the following commands:

```
cd scripts
./run.sh --input=<input_file_path> [options]
```

The following options can be used

<code>--output=&lt;output_file_path&gt;</code>	: Output file specification (default is <code>ast.dot</code> and <code>ast.pdf</code> )
<code>--colorop=&lt;color&gt;</code>	: Color <code>for</code> operator nodes <code>in</code> the AST
<code>--colordelim=&lt;color&gt;</code>	: Color <code>for</code> delimiter nodes <code>in</code> the AST
<code>--colorkey=&lt;color&gt;</code>	: Color <code>for</code> keyword nodes <code>in</code> the AST
<code>--help</code>	: Instruction regarding usage instructions and options
<code>--verbose</code>	: Prints the <code>complete</code> stack trace of the parser execution

You can provide your own colors for operators, delimiters and keywords in the AST as described in the usage above. The default color coding in the generated AST is as follows:

**OPERATORS**      **KEYWORDS**      **DELIMITERS**      **OTHERS**

## Conventions Followed

We have considered assignment (`=`) and augmented assignment (eg : `+ =`) as operators for graph color coding.

## Optional Features Included

We have supported the following features:

- Dictionary
- `try / except`
- multi-dimensional list
- f-string

## Building AST from Parse Tree

- We removed non-terminal nodes which were single children of some other node.
- We have not included newline, indent and decent terminals in the AST to make it less cluttered.
- Operators (as well as assignment, and augmented assignment) have been shifted up so that the operands are their children.