D) Create symbol table for Student class and add entry in main table, connecting the two.
Main

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>Class</td>
<td></td>
</tr>
</tbody>
</table>

1. `jane = Student('Jane', 2024)`

   Constructor call

   * Create `__init__` function symbol table
   * Create object symbol table for `jane`

   ```python
   Name   | Type       | Value |
   ------ | ---------- |-------|
   __init__ | method |       |
   getname | | "" |
   getyear | | "" |
   setyear | | "" |
   Name   | 'Jane' |
   Year   | 2024 |
   ```

   As `Self` = connection to object symbol table that we are "in" referencing

   * in `__init__` : `Self = object we are in = jane`
   2) Line 3: `Self.name = name`
   3) Line 4: `Self.year = year`
Remember: dot notation means hop between symbol tables

We currently are doing like 3, so inside __init__:

```python
self.name = name
```

hop to 'self' table ('jane'), check variable name. Assign value name, the parameter passed into __init__ constructor — 'jane'.

Before 4) starts:

- __init__ table goes away
- Jane variable added to Main, connection to Jane object table
4) Jane, get Year()
3) Start in main table, hop to Jane → get Year

```
Main
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Student</td>
<td>&lt;Class&gt;</td>
<td></td>
</tr>
</tbody>
</table>
```

\[\begin{array}{|c|c|c|}
\hline
Name & Type & Value \\
\hline
jane & object & \\
\hline
\end{array}\]

```
< Student Class >
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>--init--</td>
<td>method</td>
<td></td>
</tr>
<tr>
<td>getNama</td>
<td></td>
<td></td>
</tr>
<tr>
<td>get Year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Year</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

```
< Student object > Jane
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>--init--</td>
<td>method</td>
<td></td>
</tr>
<tr>
<td>getNama</td>
<td></td>
<td></td>
</tr>
<tr>
<td>get Year</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Year</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| Name | Value |  \\
| Jane |       |
| Year | 2024   |
```

```
return Self.
Year
hop to "Self"
| table
| 2024 |
```
5) \texttt{jane.set\_year(2023)}

\texttt{\textbf{Main}}

\begin{tabular}{|c|c|c|}
\hline
\textbf{Name} & \textbf{Type} & \textbf{Value} \\
\hline
\texttt{Student\ Class} & & \\
\hline
\texttt{jane} & \texttt{Object} & \\
\hline
\end{tabular}

\begin{itemize}
\item \texttt{\textbf{< Student Class >}}
\item \texttt{\textbf{Name} \textbf{Type} \textbf{Value}}
\item \texttt{\texttt{--init--} method}
\item \texttt{\texttt{get\_name}}
\item \texttt{\texttt{get\_year}}
\item \texttt{\texttt{Set\ year}}
\end{itemize}

\texttt{\textbf{< Student object > jane}}

\begin{tabular}{|c|c|c|}
\hline
\textbf{Name} & \textbf{Type} & \textbf{Value} \\
\hline
\texttt{--init--} & \texttt{method} & \\
\texttt{get\_name} & & \\
\texttt{get\_year} & & \\
\texttt{Set\ year} & & \\
\hline
\texttt{\texttt{Name}} & \texttt{\texttt{Jane}} & \\
\texttt{\texttt{YEAR}} & \texttt{2024} & \texttt{2023}
\end{tabular}

\texttt{\textbf{Self.\ year = new\ year \rightarrow 2023}}

\texttt{\textbf{\_\_\_\_\_jane.\ set\_year(2023)}}
Lecture 27: L-systems

Koch Snowflake

- Start with initial shape in mind that turtle will draw:

- Represent initial shape using a string: \[ F++F++F \] (base string)

  Each character is interpreted as a command used to move turtle:
  - \( F \) means turtle move forward by a distance \( d \)
  - \( - \) means turtle turn right by an angle \( \theta \)
  - \( + \) means turtle turn left by an angle \( \theta \)

  Symbols represent L-system alphabet

- \( d \) can be anything for Koch Snowflake
- \( \theta \) should be \( 60^\circ \) for Koch Snowflake

- Different for different L-systems