

# Lab 7

Due Wednesday, April 9 by 11:59pm

Handout 17  
CSCI 334: Spring 2025

---

## Coding Guidelines

---

The usual guidelines apply for both programming and  $\text{\LaTeX}$  submissions: be sure to provide all of your source code and make sure that it compiles. You should submit both a  $\text{\LaTeX}$  source file *and* a compiled PDF for this assignment. **(20 points)**

This assignment asks you to submit a brainstorm in  $\text{\LaTeX}$ . Although  $\text{\LaTeX}$  is geared toward generating documents, the most important thing to remember is that  $\text{\LaTeX}$  is a programming language. Approach it as if you are learning a new programming language and don't assume that unfamiliar syntax works the way you expect. Write small amounts of code and compile often to isolate problems. You can comment out text using the `%` character, and this can help you when debugging.

Fortunately, for most uses,  $\text{\LaTeX}$  is a simple language. For example, if you type

```
This is some text.
```

then after you compile your document, `This is some text.` will appear in it. For formatting, we use macros. A macro (pronounced *MACK-row*) is a kind of text substitution function. For example, suppose I want to emphasize some text (i.e., make it italic). I would use the `\emph` macro, like so

```
This is \emph{some} text.
```

which renders `This is some text.`

Some commonly used macros are `\emph` for *emphasis*, `\textbf` for **boldface**, `\underline` for underline, and `\texttt` for **typewriter font**. Note that it is usually easier to use the `\verbatim` environment than `\texttt` for big blocks of code; see the  $\text{\LaTeX}$  reading for examples.

Lastly,  $\text{\LaTeX}$  uses an unusual mechanism for quoted text. Do not use the double quote character `"` otherwise your quotes will be rendered strangely. Instead, start quoted text with two backtick characters, ```, and end it with two single quote characters, `'`, like so

```
This is \emph{some} ``text.''
```

which renders `This is some "text."`

$\text{\LaTeX}$  has the ability to render virtually anything you've ever seen in a book, including fancy math symbols. Feel free to explore on your own.

---

## Turn-In Instructions

---

Turn in your work using your assigned git repository. The name of your repository will have the form `https://aslan.barowy.net/cs334-s25/cs334-lab07-<USERNAME>.git`. For example, if your CS username is `abc1`, the repository would be `https://aslan.barowy.net/cs334-s25/cs334-lab07-abc1.git`.

You should have received an invite to commit to the repository via email. If you did not receive an email, please contact me right away!

---

## Group Programming Assignment

---

This is a partner lab. You may work with another classmate if you wish, and you may co-develop solutions. Remember: although you can work on code together, you must each independently write up and submit your solution. No code copying is allowed. Tell me who your partner is by committing a `collaborators.txt` file to your repository. **Be sure to commit this file whether you worked with a partner or not.** If you worked by yourself, `collaborators.txt` should contain something like “I worked by myself.” (5 points)

This assignment is due on Wednesday, April 9 by 11:59pm.

---

## Reading

---

1. (Required) “Appendix C: A Short L<sup>A</sup>T<sub>E</sub>X Tutorial”
2. (As Needed) Previous student presentations at <http://www.cs.williams.edu/~dbarowy/protected/cs334-final-projects/index.html> (see email for password if off-campus).

---

## Problems

---

### Q1. (10 points) ..... Project Partner

For the final project, you may work with a partner or by yourself. Tell me about your project collaboration plans by filling out the following project partner sign-up sheet (<https://forms.gle/hibAoi3yatGdJpyf9>). Note that the form has an option **Help me find a partner** if you would like my help finding somebody to work with. I will try to find a match based on the interests you describe in your submitted brainstorm.

Please be sure to fill out the form whether you plan to work with a partner or not, as I need that information to create project repositories.

### Q2. (65 points) ..... Project Brainstorming

For your final project, you will design and implement a programming language. For this part of the assignment, you should think of three different possible final projects you might explore. One of those proposals should involve an artwork at WCMA’s Object Lab. The remaining two can be any project that you want. Feel free to let your imagination run wild.

If you find the notion of creating your own programming language daunting, don’t worry. For future project checkpoints, you are welcome to fall back to a “default” Object Lab project whose scope will be a little less open-ended. For now, throw caution to the wind with the assurance that at this stage any idea you dream up is non-binding.

Note that a programming language need not be a so-called general purpose programming language. A general purpose programming language is equivalent in expressive power to the lambda calculus or a Turing machine, meaning that is can be used to compute anything. Instead, I encourage you to explore domain specific programming languages, or DSLs. DSLs are usually tailored toward solving a specific problem.

You have probably used a DSL without even realizing it. Some example DSLs are:

- (a) `make`
- (b) GraphViz
- (c) Hypertext Markup Language (HTML)
- (d) Extensible Markup Language (XML)
- (e) Structured Query Language (SQL)

- (f) Markdown
- (g) L<sup>A</sup>T<sub>E</sub>X
- (h) Scalable Vector Graphics (SVG)
- (i) JavaScript Object Notation (JSON)

If you have a personal itch, scratch it. For example, I often have to grade student work, and the part I always mess up involves computing scores. Therefore, the grading rubrics I supply to my teaching assistants are actually written in a domain specific language I designed called `tabulator`.

Be creative! Do you love music, or art, or literature? Can you make a language that generates music? Could you make a language that creates art? Could you make a language that generates poetry? Some of your former classmates have designed languages that have done precisely these things and more.

For each potential language, describe

- (a) The purpose of the language. What problem does it solve? When I have trouble focusing on this part, it often helps me to think in terms of *inputs* and *outputs*.
  - i. *Programs*, written in the language you design, are the inputs.
  - ii. The output is what happens *when one of those programs is evaluated*.
- (b) Write two sample programs to demonstrate what the language might look like. Feel free not to be constrained by reality at this point. Just imagine that your programming language already exists.

For the Object Lab piece, you can propose anything computational involving that artwork. If you are struggling with this idea, as a “default project,” consider the following purpose: your language should generate artwork in the style of the artist. One way to do this would be to either make your program non-deterministic (it “interprets” the program differently every time), or perhaps it takes an input that tells the interpreter how to proceed with variations deterministically (e.g., by specifying a random seed).

For example, for the Josef Albers painting, “Homage to the Square: Warming,” one kind of “fantasy program” might be:

`5 squares, from gray to orange, stacked and 25% smaller, variation 1.`

With small tweaks to the same syntax, we can also get very different outputs.

`8 squares, from gray to blue, side-by-side and equal size, variation 2.`

Observe that for this language is it easy to ask the following questions: What’s the grammar? Does it take any additional inputs? What are the outputs? Think about those questions; we will answer them in a future lab. If the answers come easy, it means it’s probably an interesting and well-defined language. In the future you will say precisely how your language is parsed and interpreted, so although you don’t have to write anything down now, you might spend a few minutes thinking about it.

Be sure to submit your project brainstorm in a file called “`lab-7.tex`” along with a pre-built “`lab-7.pdf`” in a folder called “`project`”.

### Q3. ( $\frac{1}{10}$ th bonus point) ..... Optional: Feedback

I always appreciate hearing back about how easy or difficult an assignment is.

For  $\frac{1}{10}$ th of a bonus to your final grade, please fill out the following Google Form.