

# CMSC 221 - Math, Pseudocode, and L<sup>A</sup>T<sub>E</sub>X

Jory Denny

For the following sections, the final product is shown in the pdf and how to generate the item is found in the tex file. Single best L<sup>A</sup>T<sub>E</sub>X resource: <https://en.wikibooks.org/wiki/LaTeX>. It is typically the top result when Google search contains “latex”. Then, if you can’t find the answer, ask Piazza and I will help steer you. I will likely add to this list throughout the semester as needed.

## 1 Math and Pseudocode

### 1.1 Math

Use the following recommendations for mathematics:

- All variables should be specified in math mode. Example  $x$ . Single data elements (doesn’t matter the type) should be named with a lowercase letter. Example  $x$ , where  $x$  is an integer. Multiple data elements, i.e., sets (doesn’t matter the data structure) should be named with a capital letter. Example  $X$ , where  $X$  is a set of real numbers.
- Use  $\leftarrow$  (gets) for assignment. Example  $x \leftarrow 0$ . Use  $=$  (equals) for equality comparison of single elements. Example  $x = y$ . Use  $\equiv$  (equivalent) for equality comparison of sets. Example  $X \equiv Y$ .
- Use  $\in$  (in) for stating an-element-of. Example  $x \in X$  means that  $x$  is an element of (or in) the set  $X$ . Use  $\subset$  or  $\subseteq$  (subset or proper subset, respectively) for subset statements. Example  $X \subseteq Y$  means that the set  $X$  is a proper subset of the set  $Y$ .
- Avoid “type” as much as possible. I mean you should not say “int  $x = 0$ ;” but rather “ $x \leftarrow 0$ ” without the type specified.
- Almost anything is allowed that you might imagine in math mode. Common ones you might want: Fractions  $\frac{1}{2}$ , Subscripts  $x_i$ , Superscripts  $x^2$ , and Summations  $\sum_{i=0}^n$  or  $\sum_{a \in A}$ . Subscripts are used to name elements of series.
- Use propositional logic:  $\neg$  is negation (not !),  $\wedge$  is conjunction (and) (not &&), and  $\vee$  is disjunction (or) (not ||).
- Website for other symbols: <http://web.ift.uib.no/Teori/KURS/WRK/TeX/symALL.html>

### 1.2 Pseudocode

Use the following recommendations for pseudocode:

- Define all input and output using `\INPUT` and `\OUTPUT` prior to the steps of the algorithm
- Comments can be added either with a `\COMMENT` or `\STATE` or as an optional parameter to loops and conditionals.
- Function calls to other algorithms, member functions of ADTs, etc should be in math mode but as a different font using `\mathtt` for example.

- Use `\IF`, `\ELSIF`, and `\ELSE` for if, else-if, and else, respectively.
- Use `\FOR` for for-loops and `\FORALL` for for-each-loops.
- Use `\WHILE` for while-loops and `\REPEAT` and `\UNTIL` for do-while-loops.

Examples for all of the recommendations:

---

**Algorithm 1** Example

---

**Input:** Set  $X$ , Stack  $Y$ , positive integer  $n$

**Output:** Tree  $T$

```

Step 1 {Comment on step}
Y.pop() {Example function call of a variable}
if  $n = 0$  then {Comment on if}
  Step 2
else if  $n = 1$  then
  Step 3
else
  Step 4
for  $i \leftarrow 0..n$  do
  Step 5
for all  $x \in X$  do
  Step 6
while  $\neg Y.is\text{Empty}()$  do
  Step 7
repeat
  Step 7
until  $\neg Y.is\text{Empty}()$ 
return Return something

```

---

## 2 L<sup>A</sup>T<sub>E</sub>X Features

### 2.1 Figures

There are a few ways to include figures in a tex document. However, note that not all image formats are treated equal. You can google how to include a specific image format, but the easiest to include in a document is eps (Enriched Post Script) format.

Basically you use the “figure” environment. Environments in L<sup>A</sup>T<sub>E</sub>X typically start with a `\begin` command and end with a `\end` command. Figure 1 shows an example of this.



Figure 1: Figure included in document

There is also a much fancier way of generating images. You can use the package `tikz`. I would give an example, but each thing you want to do with it requires a different example. So if you are interested I can give you more information.

## 2.2 Tables

Tables are fairly easy to manage. Use a `tabular` environment in  $\text{\LaTeX}$ . There are many, many examples and small settings to get alignments the way you want. Ask if you can't figure something out. Simple example:

Centered text	Left Justified text	Right Justified text
Hey look	its	row two.

## 3 Citing references

Cite using the `\cite` command [1]. Then add the reference either using Bibtex or at the end of your text document using  $\text{\LaTeX}$ 's embedded citation system. Bibtex is much nicer but it is more difficult to setup.

## References

- [1] Leslie Lamport,  *$\text{\LaTeX}$ : a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.