MAS116/MAS117 PRESENTATION LAB 6

Start a new document in your preferred editor, using the default preamble on the presentation materials webpage or one from a recent document. Make sure you include the AMS packages, as usual.

1. Displaying computer code

1.1. **Using verbatim.** As mentioned in an earlier lecture, inserting large chunks of computer code can be done with the `verbatim` environment.

Start a new section called 'Inserting computer code' and a subsection called 'The verbatim environment'. Insert an introductory paragraph reading "The code below is an implementation of the 'Higher and Lower' game." Go to the course webpage, and download the `higher_lower.py` file from the extras section. Copy and paste the text into your document inside a `verbatim` environment.

The verbatim environment takes everything between `\begin{verbatim}` and `\end{verbatim}` and prints it out letter by letter – i.e., *verbatim* – ignoring any special characters or commands. As a consequence of this, you *must* have `\end{vertbatim}` on its own line.

Process your file and look at the code. What do you think? You might notice some overfull boxes, i.e., text escaping into the right margin. You might have to add extra line breaks by hand to stop that happening. The code should look okay, but we will see below that we can do better with the listings package.

There is a corresponding *inline* version of verbatim if you want to put small pieces of code, such as variable names or command names, in a sentence. This is the `\verb` command. Enter the following

```
You start an enumerated list with \verb!\begin{enumerate}!
and end it with \verb+\end{enumerate}+.
```

Here `\verb` has a slightly different syntax to most commands: instead of `\verb{stuff}` you type something like `\verb!stuff!`. The character immediately after `\verb` – above we used `!` or `+` – is what LaTeX looks for to end the command. If we insisted on using curly brackets then we couldn't to this: `}`.

1.2. **The listings package.** The listings package will alllow us to display computer code is a much nicer fashion.

(1) In your preamble put the following.

```
\usepackage{listings}
\lstset{
    breaklines=true,
    language=Python,
    frame=single,
    numbers=left,
    showstringspaces=false,
}
```

(2) Create a subsection called 'The listings package'. Copy and paste the text of the `higher_lower.py` file inside a `lstlisting` environment.

(be careful with the typing) and process the file. This should have dealt with lines spilling off the page.

(3) Experiment with the options in the `\lstset` command. For instance, change `single` and `left` to `none`.

(4) You can search the web for other options you can use.

There is an inline version of the listings environment analogous to the `\verb` command. Try the following.

```
We have a variable \lstinline!height_max!.
```

1.3. **Including a file with listings.** The `\lstinputlisting` command allows you read in a file – as long as it is in the same directory as your LaTeX file – without having to copy and paste the code into your file. Copy the file `higher_lower.py` into the same directory as your LaTeX file.

Start a new subsection called 'Including a file of code'. Type the following in your new section.

```
Here is the source code of a file.
\lstset{
    basicstyle=\ttfamily\footnotesize,
    commentstyle=\color{brown},
    keywordstyle=\color{red},
    frame=none,
}
\lstinputlisting{higher_lower.py}
```

Process your file and see what you get.

2. More LaTeX commands

This is the last lab on LaTeX and we will finish with a miscellany of features. There are many, many more things you can do with LaTeX. Start a section 'Miscellany'.

(1) Try the following code to get a function which is defined case-by-case.

```
We define a function as follows:
\[
    f(x) =
    \begin{cases}
        x^2 & \text{if } x \ge 0,\\
        0 & \text{if } x < 0.
    \end{cases}
\]
```

(2) In English there are many types of dashes, here are examples of some of them.

minus sign: $3 - 4 = -1$

hyphen: double-barrelled

en-dash: pages 7–11

Type the following, then process your file and compare the dashes and the gaps either side.

```
These are different: $3 - 4  = -1$,
double-barreled, pages~7--11.
```

In maths mode a single dash '`-`' becomes a minus sign, in text mode it becomes a hyphen. An en-dash is different to a hyphen – it is the width of a letter N – and it is mainly used for number ranges, as above, and for parenthetical comments, as in this sentence. You get an en-dash with two dashes '`--`'. In American English they also use the em-dash—which is the width of an M—but this is not usual in British English.

Typeset the following and check you have the correct dashes.

(a) The number before 0 is −1.

(b) Their opening hours are 09:00–17:00.

(3) Sometimes you have a mathematical expression which is too long to fit on a single line. In this case you need the `multline` command. Notice that the name only has one 'i' in it! It has the following form.

```
\begin{multline*}
    first part of the expression \\
    second part of the expression
\end{multline*}
```

Use this to typeset the following.

$$f(x) := \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}y_0 + \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}y_1 + \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}y_2.$$

What happens if you remove the two `*`s?

(4) You can define your own commands. Enter the following in your document.

```
We have $\left|\sin(x)\right| \le 1$
for all $x \in \mathbb{R}$.
```

Process it and check it looks fine. Now go to your preamble and enter the following.

```
\newcommand{\R}{\mathbb{R}}
```

Now go back to the sentence in your document and change `\mathbb{R}` to `\R`. Check that it works. It is common to define `\R` to be the real numbers and `\C` to be the complex numbers.

3. Further reading

If there are things you need to do which haven't been covered so far in these classes, they will probably appear in the excellent *Short Introduction to LaTeX* found at the following location.

`http://www.ctan.org/tex-archive/info/lshort/english/lshort.pdf`

Another very useful resource is the LaTeX Wikibook.

`http://en.wikibooks.org/wiki/LaTeX`

This is probably more useful as a reference guide to a specific topic. Of course, you can also search the web.

Homework

The homework this week is the mini-project.