LaTeX for Logicians

Using **`beamer.cls`**

An Intentionally Incomplete Guide

Contents

Guide version 0.7 April 2007: corrections/suggestions to `ps218@cam.ac.uk`
LaTeX for Logicians is at `http://www.phil.cam.ac.uk/teaching_staff/Smith/LaTeX/`

1 Introduction

Till Tantau's document class `beamer.cls` offers LaTeX macros for producing presentations. The package is powerful and enables you to do complex things – the document `beameruserguide.pdf` gives 214 pages of detailed instructions covering scores of commands. This guide cheerfully ignores nearly all the power and complexity, and so can be *very* much shorter.

`beamer.cls` is primarily designed for *slides* to be shown using a projector linked to your laptop; and so the class macros are designed to make use of colour, animation effects, and so on. In previous versions of this Guide, I concentrated on using the macros for making black-and-white *transparencies* to be printed out and shown using a steam-age overhead projector (hence the colour-free illustrations still in the guide). I think I was in good company in being sceptical about the benefits of using all-singing, all-dancing slide presentations: and I know I wasn't the only one to have been seduced into playing with PowerPoint, or the even prettier Keynote, only to have reverted for a while to using good old-fashioned transparencies. But nowadays lecture rooms are equipped with built-in data-projectors, departments have portable projectors to use in small seminar rooms, and so using data-projectors is a lot less hassle than it was. So this version of the Guide assumes that you too will want to use a data-projector, while wanting to keep things simple and elegant.

2 Keeping it simple

By all means skip this section. But before starting the real guide, here are some prejudiced reasons *against* bothering with fancy presentations, and *for* using `beamer.cls`.

So first, three reasons against using lap-top driven presentations with all the bells and whistles.

1. You'll waste time. Far too much time. You'll start fiddling around choosing colours and backgrounds and shadings and fonts and setting up animations and stuff. And that *always* takes much longer than you ever intended. Life is too short.

2. Over complication distracts from content.

3. Your students and colleagues still won't be that impressed and will probably think you are trying too hard. (The philosopher Stephanie Lewis, in a stage whisper during a conference talk: 'Power corrupts: PowerPoint corrupts absolutely.')

So don't bother. Keep your presentations as simple as you can. But why use `beamer.cls` to produce them? Four reasons:

1. Well, you don't want to use PowerPoint, do you? You want to use LaTeX because you *really* like its fundamental concept that you should keep the business of specifying the logical structure and content of your documents sharply distinct from the business of specifying layout.

2. You may well want to be using some logical or mathematical formulae, and we all know you can't beat LaTeX for that.

3. You are using LaTeX in a modern installation on a personal computer so are pretty likely to be using pdfTeX – e.g. via TeXShop – to produce PDF output, and `beamer.cls` works with pdfTeX (unlike e.g. `prosper.cls`). It outputs PDF files, which you display using e.g. Adobe Reader.

4. `beamer.cls` is state-of-the-art, in excellent shape, with a lot of features and good support from the author. It is the 'best-in-class'.

3 Installing `beamer.cls`

Go to `sourceforge.net/projects/latex-beamer` and download each of the zipped folders

```
latex-beamer
pgf/Tikz
```

Install these folders into your place for adding LaTeX packages. For example, if you are using Mac OSX, go to your home directory, and install them into folders in the directory

```
~/Library/texmf/tex/latex
```

Then go to the LaTeX for Logicians site and (from the LaTeX in Classroom section) download the simple blue-and-white theme file

```
logicthemelive.sty
```

and put this into the appropriate newly created folder, which will be something like

```
~/Library/texmf/tex/latex/latex-beamer-3.06/themes/theme
```

Do whatever else is necessary, depending on your installation, to get LaTeX for recognize the presence of new files; and you are done.

4 Frames: the core concept

A presentation file is written by putting together

1. an initial invocation of the `beamer` class, followed by
2. a preamble (selecting the 'theme' to use, and fixing e.g. the overall title of the presentation, etc.), followed by
3. the body of the presentation which essentially defines the content of the slides using *frames* interspersed with some (optional) structuring commands to divide the presentation into sections.

The new concept to explain here is that of a *frame*. A frame *can* generate one or more slides (e.g. by successively disclosing more bullet points). But in fact we'll begin with simple usages, where a frame correlates to a single slide.

A single slide typically has a *header* and *footer* whose content is automatically generated. For example, if you use the `logicthemelive.sty` style file, the header will contain the current section title (this header is printed large enough for your audience to read); and the footer will contain the author name, title of the whole presentation, and the slide number (this is printed much smaller, as the info is mainly for your benefit). Other 'themes' or style files arrange things differently.

Now the crucial thing: *the content between the header and the footer – the frame content, as we'll call it – is generated by the command*

`\frame{`*framecontent*`}`

where *framecontent* is, of course, more or less whatever text or other content you choose. Equivalently, you can use

```
\begin{frame}
framecontent
\end{frame}
```

And actually I find these days I use that structuring as it is less easy to "lose" terminating parentheses.

Just four points for the moment about frame contents:

1. You won't want to put large slabs of unbroken text into frames. So `beamer` slightly redefines the familiar LaTeX list environments such as `itemize` and `enumerate` to make them suitable use inside frames.

2. `beamer` also defines/re-defines several environments for displaying material in a suitable form: these are `theorem`, `corollary`, `definition`, `definitions`, `fact`, `example`, and `examples`.

3. There are in addition a number of special commands for putting content into frames: the most important of these is

   ```
   \frametitle{frametitle}
   ```

   which inserts *frametitle* at the top of the frame content.

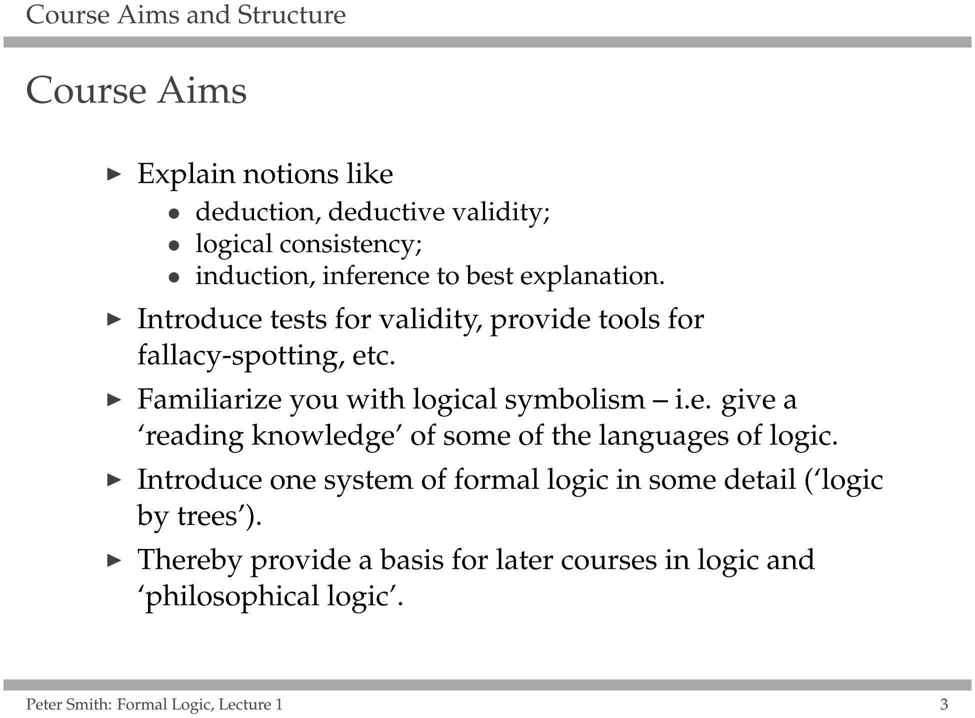
4. There is a special issue about using `\verb` or `\verbatim` contexts in frames: see Section 8.1.

Here then is a real-life example illustrating these commands: within a `beamer` document with the `logicthemelive` style file in play (and with e.g. the author name defined in the preamble), the sequence of commands

```
\frame
{
\frametitle{Course Aims}
\begin{itemize}
\item  Explain notions like
        \begin{itemize}
        \item deduction, deductive validity;
        \item logical consistency;
        \item induction, inference to best explanation.
        \end{itemize}
\item  Introduce tests for validity, provide tools for
       fallacy-spotting, etc.
\item  Familiarize you with logical symbolism -- i.e. give a
       'reading knowledge' of some of the languages of logic.
\item  ...
\item  ...
\end{itemize}
}
```

produces the single slide in Figure 1 on the next page.

It really is as simple as that!



Course Aims and Structure

Course Aims

- Explain notions like
  - deduction, deductive validity;
  - logical consistency;
  - induction, inference to best explanation.
- Introduce tests for validity, provide tools for fallacy-spotting, etc.
- Familiarize you with logical symbolism – i.e. give a 'reading knowledge' of some of the languages of logic.
- Introduce one system of formal logic in some detail ('logic by trees').
- Thereby provide a basis for later courses in logic and 'philosophical logic'.

Peter Smith: Formal Logic, Lecture 1 3

Figure 1: A slide using `logicthemelive.sty` with the option `gray`

5 Invoking the `beamer` class

To invoke the beamer class at the very beginning of your file you use

```
\documentclass[some_options]{beamer}
```

And the main options are:

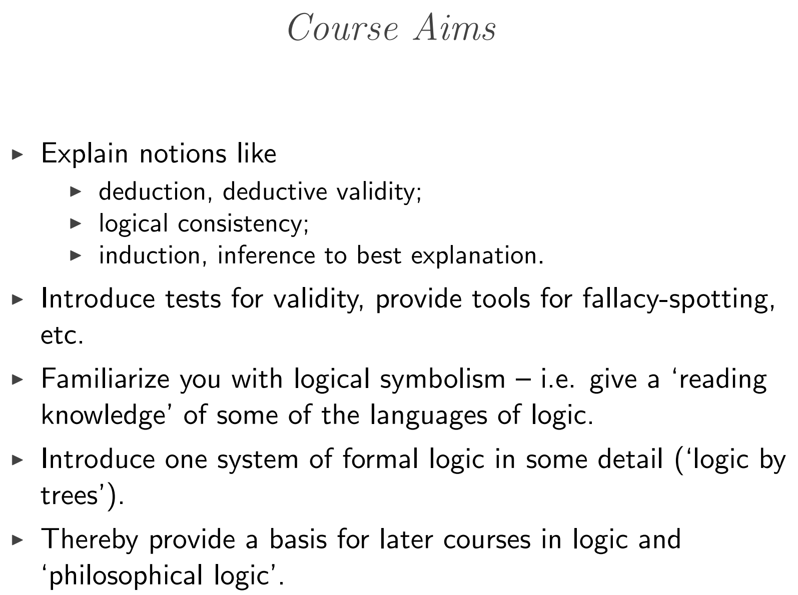
Font type: by default, presentations use sans serif font. If you want to use some serif font, then specify the option `serif`.

Font size: by default, presentations use 11pt font as the base font size. If you want to set the type somewhat smaller or larger, then you can use the options `10pt` and `12pt`.

Decolourize: if you want to be really spartan, then use `gray` to make everything greyscale. But a very modest amount of colour as in `logicthemelive` does work better, I think.

Two comments:

1. In my view, neither the Computer Modern serif font nor Times is very suitable for slides. Of the serif fonts standardly available in LaTeX, Palatino – used in the example in Figure 1 – works much better; that will, of course, need to be specified in the preamble in the usual way.
2. You may be initially surprised by the small size of the fonts being used. But in fact the notional size of a slide is set to be 12.8cm by 9.6cm. If you want to print out slides as



Course Aims

- Explain notions like
  - deduction, deductive validity;
  - logical consistency;
  - induction, inference to best explanation.
- Introduce tests for validity, provide tools for fallacy-spotting, etc.
- Familiarize you with logical symbolism – i.e. give a 'reading knowledge' of some of the languages of logic.
- Introduce one system of formal logic in some detail ('logic by trees').
- Thereby provide a basis for later courses in logic and 'philosophical logic'.

Figure 2: A slide using `beamerthemeplain.sty`

transparencies for an overhead projector, print out at double size (in landscape mode, of course). So in fact our three standard font-size options correspond to 20pt, 22pt, and 24pt type on a transparency, which about covers the range of optimal sizes if you want highly legible presentations.

6 The preamble to a presentation

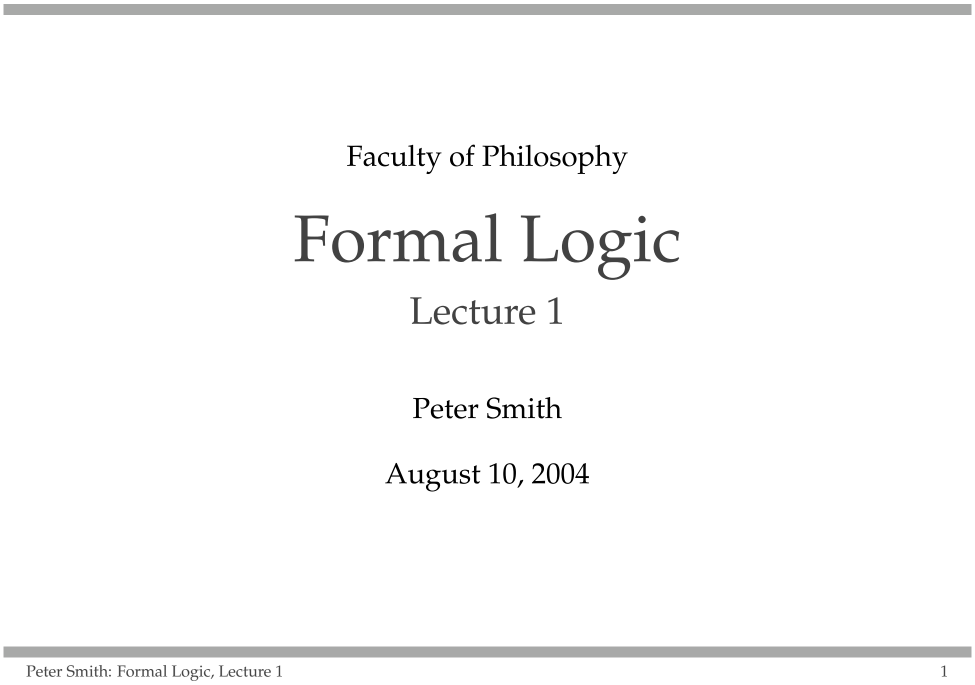
In the preamble to a `beamer` presentation you need to do three things:

1. Select a theme.
2. Define the *author*, *title*, etc.
3. Invoke other package files you need (e.g. to select fonts).

1. *Selecting a theme* `beamer` themes are standard `.sty` package files. So you simply apply a theme by issuing the command

`\usepackage{`*theme_name*`}`

Themes can be found in the folder `themes/theme` inside your folder for downloaded LaTeX packages. However, with the exceptions of the ultra simple `beamerthemeplain.sty` (see Figure 2) and `beamerthemeboxes.sty`, the themes that come with the standard distribution are not really very suitable for those who want minimalist slides. You might well therefore want to use my `logicthemelive.sty` style file instead.



Faculty of Philosophy

Formal Logic

Lecture 1

Peter Smith

August 10, 2004

Peter Smith: Formal Logic, Lecture 1 1

Figure 3: A title page using `logicthemelive.sty`

Slides come with clickable small navigation icons which I don't find very useful. You might want to add the command

```
\beamertemplatenavigationsymbolsempty
```

to remove them.

2. *Adding definitions*   Now add to the preamble the following:

```
\title[short_title_for_footer]{title_for_presentation}
\subtitle{subtitle_for_presentation}
\author{author}
\institute{institutional_affiliation}
\date{date}
```

The *short_title* option can of course be omitted. If you are using `logicthemelive.sty`, the *only* function of `\institute` is to set the over-title on the title page of the presentation as in Figure 3. You can cheerfully omit it, or use it to print some other info.

3. *Invoking further packages*   It just remains to add any further packages you need for setting the content of your slides. An obvious candidate if you want to use a serif font is

```
\usepackage{mathpazo}
```

And you way well want to invoke e.g. `amssymb` to give you access to the $\mathcal{AMS}$ symbols.

7 The body of a presentation

The basic form of the body of a presentation is very simply as follows:

```
\begin{document}
\maketitle
\frame{...}
\frame{...}
\frame{...}
...
\frame{...}
\end{document}
```

where `\maketitle` produces a title page. Or of course, if you prefer,

```
\begin{document}
\maketitle
\begin{frame}
     ...
\end{frame}
\begin{frame}
     ...
\end{frame}
\begin{frame}
     ...
\end{frame}
...
\end{document}
```

However, if you have more than a few slides, you will probably want to divide your presentation into sections, and every so often show slides indicating which section you've reached in the presentation.

You organize this in two stages. First, to mark the beginning of a section, insert between frames the command

`\section{`*section_title*`}`

Note, this command does *not* generate a new slide: but the section title will feature in the header of subsequent slides, and will appear in tables of contents. This command also has a starred version `section*` which produces a section-title – e.g. 'Table of contents' – that may appear in the header of a frame, but which won't be listed in tables of contents.

Second, when you want to show the table of contents as a slide, you use

```
\frame{\tableofcontents[current]}
```

Here, the modifier `[current]` is optional. Without it, you get a frame listing all the sections of the presentation. With the modifier, you still get a frame listing all the sections of the presentation, but with all but the current section-title greyed out, as in Figure 4.

So putting all that together, here's a characteristic structure for a whole presentation, starting with a outline summary of section headings and finishing with a reminder of that summary:

```
\documentclass[some_options]{beamer}
% start preamble
\usepackage{logicthemelive.sty}
\title{title_for_whole_presentation}
\subtitle{subtitle}
\author{author}
\institute{institutional_affiliation}
\date{date}
\usepackage{some_other_packages}
% end preamble
\begin{document}
     \maketitle
\section*{Outline}
     \frame{\tableofcontents}
\section{title_for_first_section}
     \frame{some_content}
     \frame{some_content}
\section{title_for_next_section}
     \frame{\tableofcontents[current]}
     \frame{some_content}
     \frame{some_content}
\section{title_for_next_section}
     \frame{\tableofcontents[current]}
     \frame{some_content}
     ...
\section*{Outline}
     \frame{\tableofcontents}
\end{document}
```

And that – in nine pages flat! – is basically all you need to know to produce extremely acceptable slides using `beamer.sty`.

In the rest of this Guide, I add just a little more useful information: but you can get by perfectly well using no more than you've met already.

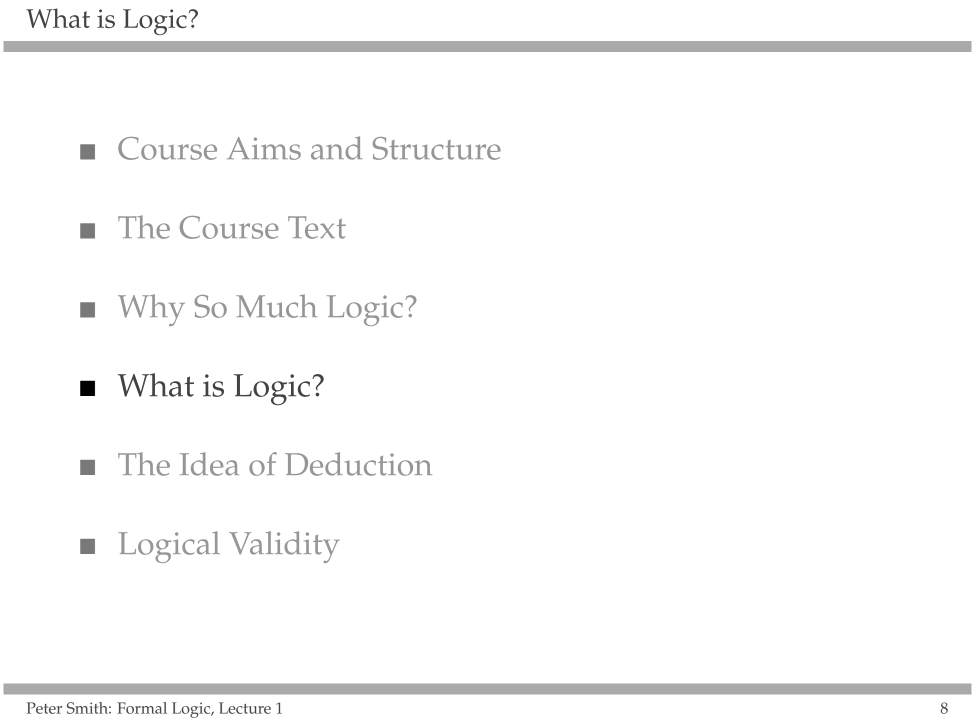
8 Extras

It would rather defeat the object of this Guide if I now replicated all the further information that can be found in `beameruserguide.pdf`. What follows is a small selection of additional information that might be particularly useful to those producing slides.

8.1 More on frames

1. *Plain frames* Sometimes, while using a theme like `logicthemelive` which has headers and footers, you'll want to insert a slide that puts material on a completely plain background (e.g. you want to insert a graphic that occupies more of less a whole slide, or give a proof that takes all of a slide). Use the option `plain` thus:

```
\frame[plain]{...}
```



What is Logic?

- Course Aims and Structure
- The Course Text
- Why So Much Logic?
- What is Logic?
- The Idea of Deduction
- Logical Validity

Peter Smith: Formal Logic, Lecture 1 8

Figure 4: A table of contents page, using the `[current]` option

or

```
\begin{frame}[plain]{...}
```

2. *Verbatim frames* Sometimes you'll want to insert verbatim material in a frame, using `\verb` or `\verbatim`. You need in this case to declare the option `containsverbatim` thus:

```
\frame[containsverbatim]{...}
```

And this is not compatible with using overlays (to be explained) with this frame. The document `beameruserguide.pdf` explains some more complex procedures for use if you *must* insert verbatim material while using overlays.

8.2 Introducing overlays

If you are producing a slide presentation then, yes, you may want bullet points to appear sequentially in successive slides to build up a composite slide. So in this sub-section, I introduce some of the 'overlay' commands available in `beamer`.

1. *Automatically revealing bullet points* If you just want bullet points to appear one by one in sequence as you click through the presentation, then just add the option `[<+->]` after the `\begin{itemize}` command, to give the frame-structure

```
\frame
{
\frametitle{Title}
\begin{itemize}[<+->]
\item ...
\item  ...
\item  ...
\end{itemize}
}
```

2. *Overlay specifications* For more hand-tuning, the basic idea to understand is that we can numerically qualify various commands used in building frame-contents with *overlay specifications* like `<1>`, `<2>`, `<3>`. *These will have the effect of generating a number of different slides from the same frame*, with text in the scope of *command*`<1>{...}` only appearing on the first slide; text in the scope of *command*`<2>{...}` appears on the second slide; and so on.

More complex markers are also available: thus text marked `<2->` will appear from slide 2 onwards in the collection generated by the frame, and text marked `<-2>` will appear up to slide 2, and text marked `<2-4>` will appear on slides 2 to 4.

3. *Overlay modifications* A number of ordinary commands are modifiable using overlay specifications. In particular note the following example.

```
\frame
{
This frame produces three slides/transparencies
\begin{itemize}
     \item<1-> This appears from the first slide.
     \item<2>  This appears only on the second slide.
     \item<2-> This appears from the second slide.
     \item<3>  This appears on the third and last slide from this frame.
\end{itemize}
}
```

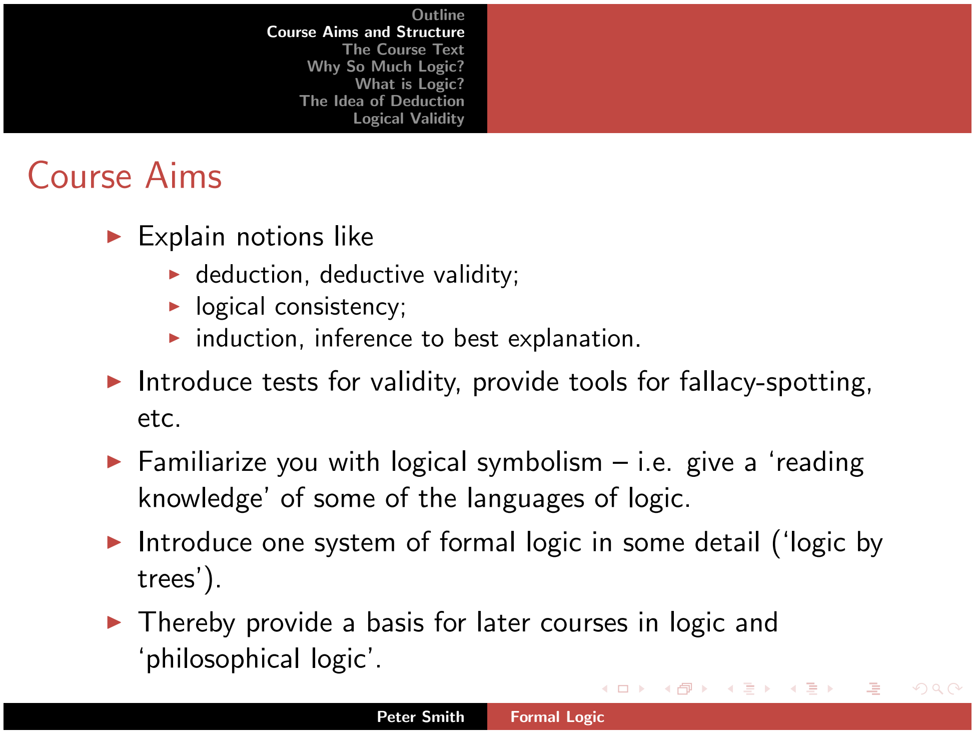
As well as items in list environments, commands like `\textbf` can take overlay specifications (so certain text can be marked as bold on some but not all slides generated by a frame).

4. *Special commands* There are also a number of special commands used in producing overlay effects. In particular note

```
\onslide<o_spec>
\only<o_spec>{...}
\visible<o_spec>{...}
\invisible<o_spec>{...}
```

To explain: material preceded by `\onslide<`*o_spec*`>` will only appear on the slide(s) specified by *o_spec*; the command stays in force through the frame until followed by another `\onslide<`*o_spec*`>`, or is cancelled by plain `\onslide`.

Material in the context `\only<`*o_spec*`>{...}` only appears in the slide(s) specified by *o_spec*, and otherwise is suppressed (not leaving a space).



Outline
Course Aims and Structure
The Course Text
Why So Much Logic?
What is Logic?
The Idea of Deduction
Logical Validity

Course Aims

- Explain notions like
  - deduction, deductive validity;
  - logical consistency;
  - induction, inference to best explanation.
- Introduce tests for validity, provide tools for fallacy-spotting, etc.
- Familiarize you with logical symbolism – i.e. give a ‘reading knowledge’ of some of the languages of logic.
- Introduce one system of formal logic in some detail (‘logic by trees’).
- Thereby provide a basis for later courses in logic and ‘philosophical logic’.

Peter Smith Formal Logic

Figure 5: A slide using `beamerthemesplit.sty`

Material in the context `\visible<`*o_spec*`>{...}` only appears in the slide(s) specified by *o_spec*, but is invisibly there on other slides. And the other way about, of course, for the command `\invisible`.

8.3 Handouts

You may well want to make handouts from a presentation. Easy. Use appropriate initial options e.g. `\documentclass[11pt, sans, gray, handout]{beamer}`, choose a simple theme if you've used something fancy for the presentation. And you'll want to “comment out” all the entries `\frame{\tableofcontents[current]}` which are unnecessary on a handout. Print four slides to a page.

8.4 And what else . . . ?

And that, I hope, is at least enough to enable you to produce classy slides for a lecture course – for though you can produce fancier effects, do you *really* need to? Is it *really* worth the time?

To produce fancier slides, explore the templates provided with the `beamer` package. The default colour scheme for themes is bluish. You can also try the `\documentclass` options `red` or `brown` (e.g. as in Figure 5).

And that's it. If you want to know more – in particular, want information about how to tinker with the templates used ‘under the bonnet’ which govern the layout of frames, headers, footers, etc. – then see the extensive documentation `beameruserguide.pdf`.