

# **A demonstration of the program environment**

Martin Ward  
Centre for Software Maintenance  
University of Durham  
04/22/21

**Contents**

# 1 Example with `\first\do5(s)etn` and `.`

The `program` style defines two environments, `program` and `programbox` for typesetting programs and algorithms. Within the `program` environment:

1. Newlines are significant;
2. Each line is in math mode, so for example spaces in the input file are not significant;
3. The command `\\` within a line causes an extra linebreak in the output;
4. The indentation of each line is calculated automatically;
5. To cause extra indentation, use the commands `\tab` to set a new tab, and `\untab` to remove it (see the examples below);
6. Vertical bars are used to delimit long variable names with underscores (and other unusual characters).

Here is a small program: It shows how to typeset mathematics as part of a program. Since each line is typeset in maths mode, all spacing is done automatically. The set brackets expand automatically, for example in this program (which also demonstrates the `\tab` and `\untab` commands):

```
[Sorry. Ignored \beginprogram ... \endprogram]
```

You can use `nvariable\do5(n)amesn` in text or math mode: `.` Names can have `nodd\do5(c)haracters:! $@# x. y. y>x$`  (*yes, I use these in my programs!*)  
*I often use bold letters to represent program fragments, formulas etc.*  
*so I have setup commands \$, etc. for the most common ones.*  
*The commands have one argument (a subscript, eg \$1, \$2, \$23)*  
*or a sequence of k prime "characters": \$', \$'' etc.*  
*If you want both a subscript and one or more primes, then you must use math mode,*  
*eg \$2j. Consider the difference between typing ```\S2''` which gives `k$2` and ```$2l`:*  
*which gives `k$2l`*

```
[Sorry. Ignored \beginprogrambox ... \endprogrambox]
```

Note that `then` and `else` should be at the *start* of a line (as in the examples above), not at the end. This is so that you can line them up in short if statements, for example:

```
[Sorry. Ignored \beginprogram ... \endprogram]
```

If the test is long, then you probably want an extra linebreak:

```
[Sorry. Ignored \beginprogram ... \endprogram]
```