

# Program Structure

[Deutsch \(de\)](#) | [English \(en\)](#) | [español \(es\)](#) | [français \(fr\)](#) | [italiano \(it\)](#) | [日本語 \(ja\)](#) | [한국어 \(ko\)](#) | [русский \(ru\)](#) | [中文（中国大陆）\(zh\\_CN\)](#) |



Basics 1A - Program Structure (author: Tao Yue, state: changed)

The basic structure of a Pascal program is:

```
PROGRAM ProgramName (FileList);

CONST
  (* Constant declarations *)

TYPE
  (* Type declarations *)

VAR
  (* Variable declarations *)

  (* Subprogram definitions *)

BEGIN
  (* Executable statements *)

END.
```

The **PROGRAM** statement is optional in Free Pascal.

The elements of a program must be in the correct order, though some may be omitted if not needed. Here's a program that does nothing, but has all the required elements:

```
program DoNothing;
begin
end.
```

Comments are portions of the code which do not compile or execute. Free Pascal supports two types of comments, free-form and line-based. Free-form comments either start with a `(*` and end with a `*)`, or more commonly, start with `{` and end with `}`. You cannot nest comments:

```
(* (* *) *)
```

will yield an error because the compiler matches the first `(*` with the first `*)`, ignoring the second `(*` which is between the first set of comment markers. The second `*)` is left without its matching `(*`. This problem with begin-end comment markers is one reason why many languages use line-based commenting systems.

Free Pascal also supports `//` as a line-based comment. When two slashes appear, other than in a quoted string or a free-form comment, the rest of the line is ignored.

Turbo Pascal and most other modern compilers support free-form brace comments, such as `{Comment}`. The opening brace signifies the beginning of a block of comments, and the ending brace signifies the end of a block of comments. Brace comments are also used for compiler directives.

Commenting makes your code easier to understand. If you write your code without comments, you may come back to it weeks, months, or years later without a guide to why you coded the program that way. In particular, you may want to document the major design of your program and insert comments in your code when you deviate from that design for a good reason.


In addition, comments are often used to take problematic code out of action without deleting it. Remember the earlier restriction on nesting comments? It just so happens that braces `{ }` take precedence over parentheses-stars `( * *)`. You will not get an error if you do this:

```
{ (* Comment *) }
```

Whitespace (spaces, tabs, and end-of-lines) are ignored by the Pascal compiler unless they are inside a literal string. However, to make your program readable by human beings, you should indent your statements and put separate statements on separate lines. Indentation is often an expression of individuality by programmers, but collaborative projects usually select one common style to allow everyone to work from the same page.



Category: [Object Pascal Introduction](#)



navigation

- [Main Page](#)
- [Documentation](#)
- [FAQ](#)
- [Downloads](#)
- [Glossary](#)
- [Index](#)
- [Recent changes](#)
- [Random page](#)
- [Help](#)

search

tools

- [What links here](#)
- [Related changes](#)
- [Special pages](#)
- [Printable version](#)
- [Permanent link](#)
- [Page information](#)