



Fundamentals

Program Structure

A Pascal program generally follows this structure:

```
program ProgramName;

uses
  {Units}; // e.g., crt, sysutils

const
  {Constants}; // e.g., MaxValue = 100;

type
  {Type Definitions}; // e.g., String20
  = string[20];

var
  {Variable Declarations}; // e.g.,
  counter : integer;

procedure ProcedureName;
begin
  {Procedure Body}
end;

function FunctionName : ReturnType;
begin
  {Function Body}
  FunctionName := ReturnValue;
end;

begin
  {Main Program Body}
end.
```

Data Types

Integer Types	integer, shortint, longint, byte, word```
Real Types	real, single, double, extended```
Character Type	char```
Boolean Type	boolean```
String Type	string[max_length] // or simply string```

Variable Declaration

Variables must be declared before use.

```
var
  variableName: DataType;
  anotherVariable: DataType;

Example:
var
  age: integer;
  name: string[50];
```

Control Structures

Conditional Statements

If-Then-Else

```
if condition then
begin
  {Statements}
end
else
begin
  {Statements}
end;
```

Case Statement

```
case variable of
  value1: begin
    {Statements}
  end;
  value2: begin
    {Statements}
  end;
  else
begin
  {Statements}
end;
end;
```

Looping Structures

For Loop

```
for i := startValue to endValue do
begin
  {Statements}
end;
```

While Loop

```
while condition do
begin
  {Statements}
end;
```

Repeat-Until Loop

```
repeat
  {Statements}
until condition;
```

Procedures and Functions

Procedure Definition

```
procedure ProcedureName(parameter1:
  DataType; parameter2: DataType);
var
  {Local Variables};
begin
  {Procedure Body}
end;
```

Function Definition

```
function FunctionName(parameter1:
  DataType; parameter2: DataType):
  ReturnType;
var
  {Local Variables};
begin
  {Function Body}
  FunctionName := ReturnValue;
end;
```

Parameters

Value Parameters	The value of the actual parameter is copied to the formal parameter.
Variable (Var) Parameters	The formal parameter becomes a reference to the actual parameter. Changes to the formal parameter affect the actual parameter.

Input/Output and Standard Functions

Input/Output

Reading Input

```
read(variable1, variable2, ...);
readln(variable1, variable2, ...); // Reads a line
```

Writing Output

```
write(expression1, expression2, ...);
writeln(expression1, expression2, ...); // Writes a line
```

Standard Functions

<code>abs(x)</code>	Returns the absolute value of x.
<code>sqr(x)</code>	Returns the square of x.
<code>sqrt(x)</code>	Returns the square root of x.
<code>sin(x), cos(x)</code>	Returns the sine and cosine of x (in radians).
<code>arctan(x)</code>	Returns the arctangent of x.
<code>exp(x)</code>	Returns e raised to the power of x.
<code>ln(x)</code>	Returns the natural logarithm of x.
<code>round(x)</code>	Rounds x to the nearest integer.
<code>trunc(x)</code>	Truncates x to the integer part.