# Pascal Programming Reference Sheet

Built In Data Types & Literals

Integers

Integer, ShortInt, LongInt

(eg: 5, 10, 15)

Floating Point Numbers

Single, Double, Extended

(eg: 3.1, 2.5, 2.1)

Strings and Characters

String, Char (eg: 'Hello', 'H')

Boolean

Boolean (eg: True, False)

Working with Strings

Assignment (giving a string a value)

name := 'Fred';

Concatenation (joining strings)

name := name + ' Smith'

Comparison

**if** name = 'Fred Smith' **then**

Construction from other types:

name := 'R' + IntToStr(2) + 'D'

+ IntToStr(2);

Programs and Modules

Creating a program

**program** HelloWorld;

… // declare things here

**begin** … **end**.

Using a module

**program** MyGame;

**uses** SwinGame, sgTypes, SysUtils;

**begin** … **end**;

Custom Types

Records

**type** Contact = **record**

name: String; …

**end**;

**var** friend: Contact;

friend.name := 'Fred';

Enumerations

**type** Grade = (Pass, Credit, …);

**var** myGrade: Grade;

myGrade := Pass;

Arrays

Declaration

**var** scores: array [0..5] **of** Integer;

**var** friends: array **of** Contact;

Access

scores[0] := 10;

friends[0] := ReadContact();

Loop

**for** i := Low(scores) **to** High(scores) **do**

scores[i] := i \* 100;

Other Things

Reading from Terminal

ReadLn(age, name);

Writing to Terminal

WriteLn('Hello ', name, ' aged ', age);

Comments

// single line

Compiling

fpc -S2 HelloWorld.pas

Declaring Functions & Procedures

Declare a procedure with parameters:

**procedure** SayHello(toName: String);

**var** … **begin** … **end**;

Declare a functions:

**functions** ReadContact() : Contact;

**var** … **begin** … result := …; … **end**;

Pass by reference:

**procedure** Swap (**var** v1, v2: Integer);

**procedure** Print(**const** friends:   
 **array of** Contact);

Simple Programming Statements

Constant declaration

const PI = 3.1415, MAX = 10;

Variable declaration

**var** name: String; age: Integer;

Assignment

name := 'Fred'; age := MAX;

Procedure Call

WriteLn('Hello World', name, age);

Sequence of statements - grouped

**begin** … **end**;

Structured Programming Statements

If statement

**if** done **then** … **else** …

case statement

**case** age **of** 1: …; 2: …; **else** …; **end;**

while loop

**while** not done **do** …

repeat loop

**repeat** … **until** done;

For loop

**for** i := 0 **to** 10 **do** …

Boolean Operators and Other Statements

Comparison: equal, less, larger, not equal, less eq

=, <, >, <>, <=, >=

Boolean: And, Or and Not

and or not

Skip an iteration of a loop

**continue**;

End a loop early

**break**;

End a function/procedure:

**exit**;