

INTRODUCTION

The following guide shows the format pseudocode will appear in the examined components. It is provided to allow you to give learners familiarity before the exam. Learners are not expected to memorise the syntax of this pseudocode and when asked may provide answers in any style of pseudocode they choose providing its meaning could be reasonably inferred by a competent programmer.

Variables

Variables are assigned using the = operator.

```
x=3
```

```
name="Bob"
```

A variable is declared the first time a value is assigned. It assumes the data type of the value it is given.

Variables declared inside a function or procedure are local to that subroutine.

Variables in the main program can be made global with the keyword global.

```
global userid = 123
```

Casting

Variables can be typecast using the int str and float functions

```
str(3) returns "3"
```

```
int("3") returns 3
```

```
float("3.14") returns 3.14
```

Outputting to Screen

```
print(string)
```

Example:

```
print("hello")
```

Taking Input from User

```
variable=input(prompt to user)
```

Example:

```
name=input("Please enter your name")
```



Iteration - Count Controlled

```
for i=0 to 7
    print("Hello")
next i
```

Will print hello 8 times (0-7 inclusive).

Iteration - Condition Controlled

```
while answer!="computer"
    answer=input("What is the password?")
endwhile

do
    answer=input("What is the password?")
until answer=="computer"
```



Logical Operators

AND OR NOT

eg

while x<=5 AND flag==false

AND (conjunction)		
INPUT		OUTPUT
A	B	$A \wedge B$
T	T	T
T	F	F
F	T	F
F	F	F

OR (disjunction)		
INPUT		OUTPUT
A	B	$A \vee B$
T	T	T
T	F	T
F	T	T
F	F	F

NOT (negation)	
of $\neg A$	
A	$\neg A$
T	F
F	T

Comparison Operators

==	Equal to
!=	Not equal to
<	Less than
<=	Less than or equal to
>	Greater than
>=	Greater than or equal to

Aithmetic Operators

+	Addition eg $x=6+5$ gives 11
-	Subtraction eg $x=6-5$ gives 1
*	Multiplication eg $x=12*2$ gives 24
/	Division eg $x=12/2$ gives 6
MOD	Modulus eg $12\text{MOD}5$ gives 2
DIV	Quotient eg $17\text{DIV}5$ gives 3
^	Exponentiation eg 3^4 gives 81



Selection

Selection will be carried out with if/else and switch/case:

if/else

```
if entry=="a" then
    print("You selected A")
elseif entry=="b" then
    print("You selected B")
else
    print("Unrecognised selection")
endif
```

switch/case

```
switch entry:
    case "A":
        print("You selected A")
    case "B":
        print("You selected B")
    default:
        print("Unrecognised selection")
endswitch
```



String Handling

To get the length of a string:

```
stringname.length
```

To get a substring:

```
stringname.substring(startingPosition, numberOfCharacters)
```

NB: The string will start with the 0th character.

Example:

```
someText="Computer Science"  
print(someText.length)  
print(someText.substring(3,3))
```

Will display:

```
16  
put
```



Subroutines

```
function triple(number)
return number*3
endfunction
```

Called from main program

```
y=triple(7)

procedure greeting(name)
    print("hello"+name)
endprocedure
```

Called from main program

```
greeting("Hamish")
```

Unless stated values passed to subroutines can be assumed to be passed by value. If this is relevant to the question byVal and byRef will be used. In the case below x is passed by value and y is passed by reference.

```
procedure foobar(x:byVal, y:byRef)
    ...
    ...
endprocedure
```



Arrays

Arrays will be 0 based and declared with the keyword array.

```
array names[5]
names[0]="Ahmad"
names[1]="Ben"
names[2]="Catherine"
names[3]="Dana"
names[4]="Elijah"

print(names[3])
```

Example of 2D array:

```
Array board[8,8]
board[0,0]="rook"
```



Reading to and Writing from Files

To open a file to read from `openRead` is used and `readLine` to return a line of text from the file.

The following program makes `x` the first line of `sample.txt`

```
myFile = openRead("sample.txt")
x = myFile.readLine()
myFile.close()
```

`endOfFile()` is used to determine the end of the file. The following program will print out the contents of `sample.txt`

```
myFile = openRead("sample.txt")
while NOT myFile.endOfFile()
    print(myFile.readLine())
endwhile
myFile.close()
```

To open a file to write to `openWrite` is used and `writeLine` to add a line of text to the file. In the program below `hello world` is made the contents of `sample.txt` (any previous contents are overwritten).

```
myFile = openWrite("sample.txt")
myFile.writeLine("Hello World")
myFile.close()
```

Comments

Comments are denoted by `//`

```
print("Hello World") //This is a comment
```

