We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright © 2015 OCR. All rights reserved.

Copyright
OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.


Registered office: 1 Hills Road
Cambridge
CB1 2EU

OCR is an exempt charity.
CONTENTS

INTRODUCTION ........................................... 4
VARIABLES ............................................. 4
CASTING ................................................. 4
OUTPUTTING TO SCREEN ............................ 4
ITERATION – COUNT CONTROLLED .............. 5
ITERATION – CONDITION CONTROLLED .......... 5
LOGICAL OPERATORS ................................. 6
SELECTION ............................................. 7
STRING HANDLING .................................... 8
SUBROUTINES ........................................ 9
ARRAYS ................................................ 10
READING TO AND WRITING FROM FILES ...... 11
COMMENTS ............................................. 11
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.

```plaintext
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.

```plaintext
global userid = 123
```

Casting

Variables can be typecast using the int str and float functions

```plaintext
str(3) returns “3”
int(“3”) returns 3
float(“3.14”) returns 3.14
```

Outputting to Screen

print(string)

Example:

```plaintext
print(“hello”)
```

Taking Input from User

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

Example:

```plaintext
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

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT</th>
<th>OUTPUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>¬ A</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Comparison Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
</tbody>
</table>

Arithmetic Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Addition eg x=6+5 gives 11</td>
</tr>
<tr>
<td>-</td>
<td>Subtraction eg x=6-5 gives 1</td>
</tr>
<tr>
<td>*</td>
<td>Multiplication eg x=12*2 gives 24</td>
</tr>
<tr>
<td>/</td>
<td>Division eg x=12/2 gives 6</td>
</tr>
<tr>
<td>MOD</td>
<td>Modulus eg 12MOD5 gives 2</td>
</tr>
<tr>
<td>DIV</td>
<td>Quotient eg 17DIV5 gives 3</td>
</tr>
<tr>
<td>^</td>
<td>Exponentiation eg 3^4 gives 81</td>
</tr>
</tbody>
</table>
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:

```java
someText="Computer Science"
println(someText.length)
println(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
```
We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: [www.ocr.org.uk/expression-of-interest](http://www.ocr.org.uk/expression-of-interest)

**OCR Resources: the small print**

OCR's resources are provided to support the teaching of OCR specification(s), but in no way constitute an endorsed teaching method that is required by the Board and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

© OCR 2015 - This resource may be freely copied and distributed, as long as the OCR logo and this message remain intact and OCR is acknowledged as the originator of this work.

OCR acknowledges the use of the following content:

Thumbs up: alex_white/Shutterstock.com, Thumbs down: alex_white/Shutterstock.com

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: [resources.feedback@ocr.org.uk](mailto:resources.feedback@ocr.org.uk)