

PSEUDOCODE SYNTAX

For OCR GCSE Computer Science

pseudocode

noun: pseudocode; noun: COMPUTING
a notation resembling a
simplified programming
language, used in
program design

Your Name :

Your Class :

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|---|---|--|-------|
| Casting | | | |
| str(3) returns "3" int("3") returns 3 float("3.14") returns 3.14 | 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 | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|----------------------|--|--|-------|
| print(string) | Taking Input from User variable=input(prompt to user) | <pre>print("hello") name=input("Please enter your name")</pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|---|--|---|-------|
| Iteration | | | |
| <pre>for i=0 to 7 print("Hello") next i</pre> | <p>Will print hello 8 times (0-7 inclusive).</p> | <pre>for i=0 to 7 print("Hello") next i</pre> | |
| <pre>while answer!="computer" answer=input("What is the password?") endwhile do answer=input("What is the password?") until answer=="computer"</pre> | | <pre>while answer!="computer" answer=input("What is the password?") endwhile do answer=input("What is the password?") until answer=="computer"</pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes | | | | | | | | | | | | | | | | | | | | | |
|---|--------------------------------|--------------|-------|----------|---|--------|---|---|--------------|---|---|---|---|---|---|---|---|---|---|---|---|--|--|--|
| Logical Operators | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="3">AND Conjunction</th> </tr> <tr> <th colspan="2">Input</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>$A \wedge B$</th> </tr> </thead> <tbody> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> </tbody> </table> | AND Conjunction | | | Input | | Output | A | B | $A \wedge B$ | T | T | T | T | F | F | F | T | F | F | F | F | <p>AND OR NOT eg while $x \leq 5$ AND $flag == false$</p> | | |
| AND Conjunction | | | | | | | | | | | | | | | | | | | | | | | | |
| Input | | Output | | | | | | | | | | | | | | | | | | | | | | |
| A | B | $A \wedge B$ | | | | | | | | | | | | | | | | | | | | | | |
| T | T | T | | | | | | | | | | | | | | | | | | | | | | |
| T | F | F | | | | | | | | | | | | | | | | | | | | | | |
| F | T | F | | | | | | | | | | | | | | | | | | | | | | |
| F | F | F | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="3">OR Disjunction</th> </tr> <tr> <th colspan="2">Input</th> <th>Output</th> </tr> <tr> <th>A</th> <th>B</th> <th>$A \vee B$</th> </tr> </thead> <tbody> <tr> <td>T</td> <td>T</td> <td>T</td> </tr> <tr> <td>T</td> <td>F</td> <td>T</td> </tr> <tr> <td>F</td> <td>T</td> <td>T</td> </tr> <tr> <td>F</td> <td>F</td> <td>F</td> </tr> </tbody> </table> | OR Disjunction | | | Input | | Output | A | B | $A \vee B$ | T | T | T | T | F | T | F | T | T | F | F | F | | | |
| OR Disjunction | | | | | | | | | | | | | | | | | | | | | | | | |
| Input | | Output | | | | | | | | | | | | | | | | | | | | | | |
| A | B | $A \vee B$ | | | | | | | | | | | | | | | | | | | | | | |
| T | T | T | | | | | | | | | | | | | | | | | | | | | | |
| T | F | T | | | | | | | | | | | | | | | | | | | | | | |
| F | T | T | | | | | | | | | | | | | | | | | | | | | | |
| F | F | F | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <thead> <tr> <th colspan="2">NOT Negation Of $\neg A$</th> </tr> <tr> <th>A</th> <th>$\neg A$</th> </tr> </thead> <tbody> <tr> <td>T</td> <td>F</td> </tr> <tr> <td>F</td> <td>T</td> </tr> </tbody> </table> | NOT Negation Of $\neg A$ | | A | $\neg A$ | T | F | F | T | | | | | | | | | | | | | | | | |
| NOT Negation Of $\neg A$ | | | | | | | | | | | | | | | | | | | | | | | | |
| A | $\neg A$ | | | | | | | | | | | | | | | | | | | | | | | |
| T | F | | | | | | | | | | | | | | | | | | | | | | | |
| F | T | | | | | | | | | | | | | | | | | | | | | | | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|-----------------------------|--------------------------|---------|-------|
| Comparison Operators | | | |
| == | Equal to | | |
| != | Not Equal to | | |
| < | Less Than | | |
| <= | Less Than or Equal to | | |
| > | Greater Than | | |
| >= | Greater Than or Equal to | | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|--|---|---|-------|
| Selection | | | |
| <pre> if entry=="a" then print("You selected A") elseif entry=="b" then print("You selected B") else print("Unrecognised selection") endif </pre> | <p>Selection will be carried out with if/else and switch/case</p> | <pre> if entry=="a" then print("You selected A") elseif entry=="b" then print("You selected B") else print("Unrecognised selection") endif </pre> | |
| <pre> switch entry: case "A": Statement case "n": Statement default: Statement endswitch </pre> | | <pre> switch entry: case "A": print("You selected A") case "B": print("You selected B") default: print("Unrecognised selection") endswitch </pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|---|--|--|-------|
| Strings | | | |
| <code>stringname.length</code> | To get the length of a string The string will start with the 0th character. | | |
| <code>stringname.substring(startingPosition, numberOfCharacters)</code> | To get a substring The string will start with the 0th character. | <pre>someText="Computer Science" print(someText.length) print(someText.substring(3,3))</pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|--|------------------------------|--|-------|
| Subroutines | | | |
| function Name (Par) Statement endfunction | | <pre>function triple(number) return number*3 endfunction</pre> | |
| procedure Name (Par) Statement endprocedure | Called from the main program | <pre>y=triple(7) procedure greeting(name) print("hello"+name) endprocedure</pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|--|--|--|-------|
| Arrays | | | |
| <pre> array ArrayName[index] names[0]=data names[1]=data names[n]=data </pre> | <p>Arrays will be 0 based and declared with the keyword array.</p> | <pre> array names[5] names[0]="Ahmad" names[1]="Ben" names[2]="Catherine" names[3]="Dana" names[4]="Elijah" print(names[3]) </pre> | |

PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|---|---|---|-------|
| Reading and Writing to Text Files | | | |
| <pre>myFile = openRead("sample.txt") x = myFile.readLine() myFile.close()</pre> | <p>To open a file to read from openRead is used and readLine to return a line of text from the file.</p> | <pre>myFile = openRead("sample.txt") x = myFile.readLine() myFile.close()</pre> | |
| <pre>myFile = openRead("sample.txt") while NOT myFile.endOfFile() print(myFile.readLine()) endwhile myFile.close()</pre> | <p>endOfFile() is used to determine the end of the file. The following program will print out the contents of sample.txt</p> | <pre>myFile = openRead("sample.txt") while NOT myFile.endOfFile() print(myFile.readLine()) endwhile myFile.close()</pre> | |
| <pre>myFile = openWrite("sample.txt") myFile.writeLine("Hello World") myFile.close()</pre> | <p>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).</p> | <pre>myFile = openWrite("sample.txt") myFile.writeLine("Hello World") myFile.close()</pre> | |



PSEUDOCODE SYNTAX

| Syntax | Meaning | Example | Notes |
|----------------------------------|----------------------------|---|-------|
| Comments | | | |
| <code>//This is a comment</code> | Comments are denoted by // | <pre>print("Hello World") //This is a comment</pre> | |

pseudocode

noun: pseudocode; noun: COMPUTING
a notation resembling a
simplified programming
language, used in
program design