

1.1 Algorithm key terms		Computer Science Year 10 Cycle 3	
input	Data which is inserted <b>into a system</b> for processing and/or storage	string length	Counts all characters in the string
output	Data which is <b>sent out</b> of a system	substrings	Continuous characters within a string
process	An <b>action</b> that takes place in an algorithm	concatenation	Join strings and variables
decision	A <b>yes/no/true/false</b> decision made in an algorithm	change case	Alternate between <b>upper/lower case</b> characters
logic	The <b>aim</b> of the program	ASCII conversion	Convert from ASCII into binary
1.2 testing		1.6 Algorithms	
test data	the data <b>selected for analysis</b>	abstraction	The <b>process of removing</b> unnecessary information
boundary/ extreme	data the is accepted, but is <b>close to</b> the required checking area	decomposition	The <b>process of breaking down</b> a problem into subroutines
valid	possible data that the program should <b>accept</b> and <b>process</b>	algorithmic thinking	Using <b>logic</b> and <b>reasoning</b> to solve common problems
erroneous	data that the program can not process and <b>should not accept</b>	designing algorithms	Producing <b>flowcharts</b> or <b>pseudocode</b>
input validation	a test to ensure the <b>correct data</b> type has been inserted	searching algorithms	<b>Binary and linear search algorithms</b> for locating data
iterative	carried out while a program is <b>being developed</b> . ... The process repeats (iterates) until the module works as intended.	sorting algorithms	<b>Bubble, merge and insertion</b> used to create solutions
final	program is tested <b>as a whole</b> to ensure that it functions	1.7 Programming fundamentals	
1.3 maintainability		variables	A <b>location in memory</b> that can be changed and reassigned during the running of the program
maintainability	allows edits and updates of created programs <b>easily</b>	constants	A <b>location in memory</b> that cannot be changed throughout the running of the program.
debug	locate and resolve an <b>error</b>	operators	Can be either <b>comparison</b> or <b>arithmetic operators</b>
comments	provide <b>additional information</b> , ignored by the program	data types	How data is <b>represented</b> in a computer program.
sensible variable names	pertaining to the <b>data type or function</b> of the variable	string manipulation	<b>Altering the formatting</b> of the characters
indentation	formatting to show which lines of code are <b>linked</b>	array	Fixed length static structure to <b>store data</b> . One/ two D
1.4a program errors		SQL	<b>Structured query language</b>
syntax error	occurs when rules of programming are <b>not followed</b>	function	Performs a task – <b>does return</b> a value
logic error	an <b>inaccuracy in the way the program functions</b>	procedure	Performs a task – <b>does not return</b> a value
run-time error	undetected during compilation, but discovered whilst the program is <b>running</b>	1.8 Sorting and searching algorithm	
1.4b data types		linear search	<b>One by one</b>
casting	<b>Changing the data type</b> within the variable	binary search	<b>Discards half</b>
string	<b>Alphanumeric</b> characters	bubble sort	<b>Swaps</b> two values
integer	<b>Whole numbers</b> (no decimal numbers)	merge sort	Splits into <b>individual</b> values
float	<b>Decimal numbers</b> (no decimal point limitation)	insertion sort	Places the value in the <b>correct location</b>
real	Synonymous with <b>float</b>		
Boolean	Returns only <b>true/false</b>		

1.9 File handling	
File handling operations	Using <b>Python commands</b> to interact with a document
open	Prepares for data to be written <b>into file</b>
close	<b>Closes and applies</b> changes
read line	<b>Return a specified line</b> from within the file
write line	<b>Adds an additional line to the file</b>
end of file	Sends the cursor to the <b>last character</b>
create a new file	Produce a <b>blank text file</b>

## OCR ERL Cheatsheet

### Commenting

Comment

```
//my note to me
```

### Variables

Assignment

```
myAge = 36
```

Constants

```
const pi = 3.14
```

Global Variables

```
global lives = 3
```

### Casting

To String

```
str(36)
```

To Integer

```
int("13")
```

To Float

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

To Real

```
real("3.14")
```

To Bool

```
bool("True")
```

### Operators

#### Comparison Operators

```
myAge == 36 //equal
lives != 0 //not equal
health < 1 //less
score > 0 //greater
marks <= 40 //less or equal
marks >= 80 //greater or equal
```

#### Arithmetic Operators

```
4 + 5 //add
9 - 6 //subtract
2 * 4 //multiply
5 * 3 //exponent
6 / 3 //divide
7 MOD 2 //modulus
8 DIV 3 //quotient
```

#### Logical Operators

```
age > 18 AND age < 60
hour < 9 OR hour > 17
NOT day == "Sunday"
```

### String Operations

String Length

```
name.length
```

Substrings

```
name.substring(2, 4)
name.left(3)
name.right(5)
```

Concatenation

```
print("Hi" + name)
```

Change Case

```
name.upper
name.lower
```

ASCII Conversion

```
ASC(X)
CHR(75)
```

### Arrays

Declaration

```
array score[5]
array ages["Dan","Ali"]
array users[4, 4]
```

Assignment

```
score[0] = 59
users[1,3] = "Ninja01"
```

Length

```
len(score)
```

Random Numbers

Random Numbers

```
i = random(1,9)
r = random(1.1, 7.5)
```

### File Handling

Open

```
f = open("data.txt")
```

Read Line

```
f.readline()
```

End of File

```
while NOT f.endOfFile()
  print f.readline()
endwhile
```

Create a New File

```
newFile("newdata.txt")
```

Close

```
f.close()
```

Write Line

```
f.WriteLine("Hello")
```

### Sub Programs

Procedure

```
procedure sum(n1,n2)
  print(n1 + n2)
endprocedure
```

Call a Procedure

```
sum(8,9)
```

Function

```
function sum(n1,n2)
  return(n1 + n2)
endfunction
```

Call a Function

```
result = sum(8,9)
```