ey word	definition	example		
	Assigned using the =			
	Assigned using the -			1.2a Programming fundamentals
	operator , can	44		
noric variable	change/undate	age=11	variables	A location in memory that can be changed and reassigned

name = "Laura"

1.5 OCR reference language

change/update

throughout the

generic variable

till oughout the					l I	
program	const yet = 0.25	constants	•	ot be changed throughout the	decomposition	down a pro
			<u> </u>			chunks kno
· ·		operators	Can be either <b>comparison</b> or <b>ari</b>	thmetic operators	"	Using logic
· -	rurposc	data types	How data is represented in a cor	nputer program.		problems
	global userID="12345"		Aller de aller de la les en efilhe de f			producing
·	B.000. 000.12 120.10	casting	Aftering the data type of the info	ormation		common e
B	name = str(input("type in name"))	string	Altering the characters in a stri	ng		binary and
assumes the type of	age=int(input("how old are you?"))	manipulation			<u> </u>	using repe
data it is given	height=float(input(How tall are	"		• • • • • • • • • • • • • • • • • • • •	"	-
	you?"))	operations	close to interact with a documer	nt	Laigoriumis	create solu
changing the data	ago = input(" ")) > ago = int(:t())	array	Fixed length static structure to s	tore data. Can be one or two		
type: str, float, int	age = input( )) -> age=int(input())		dimensional		1.4 Arithmotic or	norators -
displaying data out	print/"NAv nama is Bah")	SQL	Structured query language		+	add
of the program	print( My name is Bob )	function	Built-in code, returns a variable		-	sub
request user to <b>type</b>		procedure	Built in code does not return a	rariable	*	mul
•	name=input()	procedure	Built-iii code, does not return a v	Variable	٨	ехр
					/	divi
repeated sections of	for x = 0 to 4	1.6 String handlin	og/onorations		MOD	mod
code	while age < 10:			example		int/
For loop, instructions	for x=0 to 9					inte
repeated a <b>set</b>	print(x)				quotient	inte
number of times	next x			output 15		
While loop	while answer !="Computer	substrings	.substring(x,i) returns the	subject ="ComputerScience"		
• •	Science"		characters starting from the	subject.substring(3,5)	4.0.0	
	answer=input("favourite		first position, for the length	output "puter"	· · · · · · · · · · · · · · · · · · ·	
	subject?")		specified		==	equ
	end while	left	` '	subject ="ComputerScience"	!=	not
	//this section is a function to					
. •	square a number		string		<	less
	#this is the rand function	right			<del> </del>	less
code			from the <b>end of the string</b>	, , ,	`	1622
	str (alphanumeric)	concatonation	+ used to combine strings		>	grea
variety of accepted	float (decimal)	Concatenation	+, used to combine strings			tha
, ,	real (interchangeable with float)	uppercase	upper, will convert all	,	>=	grea
	, , ,	appereuse			<u> </u>	
	Bool(Boolean, True or False)		, and the second			
				3.2.103.21462		
	if answer ==yes: then	lowercase	.lower, will convert all	subject ="ComputerScience"		
	value remains the same throughout the program keyword global required in assignment assumes the type of data it is given changing the data type: str, float, int displaying data out of the program request user to type into the program repeated sections of code  For loop, instructions repeated a set	value remains the same throughout the program  keyword global required in assignment  assumes the type of data it is given  changing the data type: str, float, int  displaying data out of the program  request user to type into the program  request user to type into the program  repeated sections of code  For loop, instructions repeated a set number of times  While loop, instructions continue until a circumstance is met  // or #. Ignored by the program, but used to describe code  variety of accepted information types  value remains the const vat = 0.25  drivers = "Mastery" "Autonomy"  "Purpose"  load userID="12345"  age = input("type in name"))  age=int(input(How tall are you?"))  height=float(input(How tall are you?"))  age = input("")) → age=int(input())  print("My name is Bob")  print("My name is Bob")  for x = 0 to 4  while age < 10:  for x = 0 to 9  print(x)  next x  while answer !="Computer Science"  answer=input("favourite subject?")  end while  // this section is a function to square a number  #this is the rand function  str (alphanumeric)  float (decimal)  required in assignment  request user to type in name"))  age = infut("Type in name"))  age = infut("Subjection is afunction to square a number  #this is the rand function  brown in the program, but in the program in the	value remains the same throughout the program "Purpose" (ata types global required in assignment assumes the type of data it is given print("mow old are you?")) age=int(input("type in name")) age=int(input("type in name")) age=int(input("how old are you?")) height=float(input(How tall are you?")) age = input("")) → age=int(input()) array a	rolue remains the same throughout the program  const vat = 0.25  drivers = "Mastery" "Autonomy"    "Purpose"  data types of data it is given  assumes the type of data it is given  changing the data type: str, float, int displaying data out of the program  request user to type into the program  request user to type into the program  repeated sections of code  For loop, instructions repeated a set number of times  While loop, instructions repeated a set number of times  While loop, instructions or metal and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while  ### While loop, instructions or metal time and while answer !="Computer Science" answer=input("favourite subject?")  ### while answer !="Computer Science" answer=input("favourite subject?")  ### while answer !="Computer Science" answer=input("favourite subject?")  ### while in the program and prunting of the program.  ### Altering the data type of the info attaining and altering the characters in a string manipulation  ### float (input ("type in name"))  ### and altering the data type of the info and attaining and altering the characters in a string manipulation  ### float (input ("type in name"))  ### and altering the data type of the info and attaining and altering the characters in a string manipulation  ### float (input ("type in name"))  ### and altering the data type of the info  ### and altering the data type of the info  ### and altering the data type of the info  ### and altering the data type of the info  ### and altering the data type of the info  ### array  ### Fixed length static structure to string function  ### SQL Structured query language  ### function  ### Built-in code, does not return and into an aution and into a string length.  ### substrings  ### substrings  ###	roule remains the same throughout the program  value remains the same throughout the program.  same throughout the program sexyword global required in assignment  assumes the type of data it is given  and pallation  file handling  assumes the type of the info	Constants   A location in memory that cannot be changed throughout the rounning of the program.   Suppose   Further   Can be either comparison or arithmetic operators   Can be either program.   Can be either comparison or arithmetic operators   Altering the data type of the information   Disport the program.   Altering the data type of the information   Disport the program.   Altering the data type of the information   Disport the program.   Altering the data type of the information   Disport the data type of the information   Disport the program.   Disport the

ascii conversion

print("that's correct")

print ("error")

print ("that's not correct")

elif

choice/ decision. IF,

ELIF, ELSE

selection

lowercase

ACS(...)

characters in the string to

will return the ASCII binary

**chart value** for the character,

Computing Year 10 Cycle 1 Computer Science

during the running of the program

1.1 Programming constructs

chunks known as subroutines

abstraction

output "computerscience"

subject.upper4)

ASC(A)

output 65

removing unnecessary information from a problem

**down** a problem into smaller, more manageable

producing **flowcharts** or **pseudocode** and identify

binary and linear search algorithms for locating using repetition or mathematical calculations bubble, merge and insertion sorting algorithms create solutions using logic and pattern recognition

Using logic and reasoning to solve common

common errors through trace tables

addition of int/float values subtraction of int/float values multiplication of int/float values exponent (to the power of) division of int/float values

int/float division

equal to not equal to less than

less than or equal to

greater than or equal to

modulus, outputs the remainder of a

integer division to find the quotient integer value before the decimal point

greater Built-in code, returns a variable

	1.1 Network types		Networks	and protocols		e four layer model
LAN	local area network > used in small areas		1	.4a threats to a network	application	encodes/decodes data
			brute force	trial and error > login info > encryption	transport	break data into packets
WAN	wide area network > group of LANs > larger are	a	denial of service	attack target with traffic > shut down servers	network	adds S/D IP address
		T.	data interception	data loss > transmission > packet sniffer	data link	transfers data
Client - server	client requests > server sends		·	·		1.8 Protocols
	Ĺ		SQL injection	malicious code > attack database	protocol	rules for transmission
Peer to Peer	no centralised device > each node acts as serve		poor network policy	inadequate rules for clients	standard	agreed, recognised rules
			malware	group term for malicious software	IPv4	4 <sup>th</sup> IP version, uses 32b
Name	1.2 Network hardware description	image	phishing	social engineering > stealing data > email		· ·
router	forwards packets > provides IP	9 9	people	individual lack of responsibility	IPv6	6 <sup>th</sup> IP version, used 128b
router	addresses > gateway to Internet			1.4b malware	MAC	unique identifier on the
	dudiesses > gateway to internet		viruses	code hidden in files > self-replicate >	address	NIC
hub	broadcasts data to all devices			delete/modify data	ethernet	data transmission LAN
		<u> </u>	worm	not hidden > spread through email > self- replicating	НТТР	Hyper Text Transfer
bridge	connect networks	•	trojan	disguised as legitimate looking programs >	HTTPs	Hyper Text Transfer
		<u> </u>		user installed		(secure)
switch	intelligent > forward packets > uses	8888	spyware	monitor user activity > passwords > websites	TCP/IP	Transmission Control /
	MAC address		ransomware	blackmail user to pay > encrypts data as ransom	FTP	Internet File Transfer
server	provides service to client > higher spec than PC		h and shall a	1.5 packet switching	IMAP	Internet Mail Access
	spec than r c		handshake	agreement by devices to send and receive	<u> </u>	
Wireless Access Poi	·	(((†)))	source	where the data is coming from	SMTP	Simple Mail Transfer
	access the network	• …	destination	where the data is going to	POP3	Post Office
network interface ca	ard circuit board > allows network	7	payload	the information to be transmitted	' ' ' ' '	
	connectivity > RJ45 port >	<b>三</b> 罗	header	contains data for transmission	VOIP	Voice over Internet
	onboard/external		packet sequence	total # of packets and reload order		
GSM chip	standard (ETSI) > mobile device to			1.6 encryption		odes of connection
	phone network (SIM)	<b>[93</b> ]	encryption	applying algorithm > encoding > unreadable	wired	cabled > Fibre Optic / Ethernet
transmission media	medium for data transfer	<u></u>	decryption	Data made readable again	wireless	radio signals >
		<b>令《*》</b>	cipher text	encrypted data		translated by router
Name	1.3 transmission media		Caesar Cipher	Julias Caesar > uses a key to change letters	Bluetooth	radio waves > short
Name coaxial cable	description  description  high frequency > low latency > TV/p	phone/internet	algorithm	a program to scramble data		distance .0 terms for PC
			private key	kept safe > used for decryption	standalone	
Ethernet cable	RJ45 port > internet transmission > s	standard IEEE (1983)	public key	can be used to encrypt a message	node	]       =
duplex cable	send & receive > used in Fibre Optic		asymmetric	encryption using public and private keys	device	<b>┤                                    </b>
fibre optic cable	coated glass > data =light > very low	v latency > high speed	symmetric	using the same key to encrypt/decrypt	client workstation	

	1.1 Algorithm key terms	A	lgorithm	s and programming		Fil	e handling	
input	Data which is inserted <b>into a system</b> for processing and/or		s	tring operations	File handli	ng Usir	ng <b>Python co</b>	mmands to
'	storage	string length	Counts a	all characters in the string	operations		ract with a d	
output	Data which is <b>sent out</b> of a system				open			a to be written
process	An <b>action</b> that takes place in an algorithm	substrings		ous characters within a string	close		file ses and appl	os changes
·		concatenation	Join stri	ngs and variables	read line			ed line from
decision	A yes/no/true/false decision made in an algorithm	change case	Alternat	e between <b>upper/lower case</b> characters	''		nin the file	ed inic ironi
logic	The <b>aim</b> of the program	ASCII conversion	Convert	from ASCII into binary	write line	Add	<b>ls</b> an additio	nal <b>line to the</b>
	1.2 testing			1.3 Algorithms	end of file	file	ds the curso	to the last
test data	the data selected for analysis	abstraction	The pro	cess of removing unnecessary information		cha	racter	
boundary/ extreme	data the is accepted, but is <b>close to</b> the required checking	decomposition	•	cess of breaking down a problem into	create a ne	ew Pro	duce a <b>blank</b>	text file
valid	possible data that the program should accept and process	algorithmic	Subrout	gic and reasoning to solve common				
Valla	possible data that the program should decept and process	thinking	problen		OCR ERI	. Chea	tsheet	KNOWITALL AND NINIA
erroneous	data that the program can not process and <b>should not</b>	designing		ng flowcharts or pseudocode	Commenting	Casting	Operators	<b>,</b> <u>-</u>
	accept	algorithms			Comment //my note to me	To String str(36)	Comparison Operator	Arithmetic Operators
input validation	a test to ensure the <b>correct data</b> type has been inserted	searching	Binary a	and linear search algorithms for locating	Variables	To Integer	lives != 0 //not health < 1 //les	2 * 4 //multiply
iterative	carried out while a program is being developed The	algorithms	data		Assignment	int("13") To Float	score > 0 //gre marks <= 40 //les equ	or 6 / 3 //divide al 7 MOD 2 //modulus
	process repeats (iterates) until the module works as	sorting	Bubble,	merge and insertion used to create	myAge = 36	float("3.14")	marks >= 80 //gre or	
	intended.	algorithms	solution		const pi = 3.14	real("3.14")	Logical Operators  age > 18 AND age	c 60
final	program is tested <b>as a whole</b> to ensure that it functions			ramming fundamentals	Global Variables	To Bool	hour < 9 OR hour NOT day == "Sunda	17
	1.3 maintainability	variables		on in memory that can be changed and	global lives = 3	bool("True")	String Operations	Arrays
maintainability	allows edits and updates of created programs easily			ned during the running of the program	Input/Output		String Length	Declaration
	, , ,	constants		on in memory that cannot be changed	pwd = input("Please o	enter a password")	name.length Substrings	array score[5] array ages["Dan","Ali"] array users[4, 4]
debug	locate and resolve an <b>error</b>	operators		out the running of the program. either comparison or arithmetic operators	Output print("You have logge	4 ( 11 11)	name.substring(2, name.left(3)	4) Assignment
comments	provide additional information, ignored by the program	•		<u> </u>	Selection	ed in successucty")	name.right(5) Concatenation	score[0] = 59 users[1,3] = "Ninja01"
sensible variable	pertaining to the <b>data type or function</b> of the variable	data types	How da	ta is <b>represented</b> in a computer program.	If-Then-Else		print("Hi" + name Change Case	len(score)
names	<b>0</b>	string	Altering	the formatting of the characters	if hour < 12 then print("Good Mornin elseif (hour < 18) th	ng!")	name.upper name.lower	Random Numbers
indentation	formatting to show which lines of code are linked	manipulation			print("Good Aftern	noon!")	ASCII Conversion	Random Numbers
		array	Fixed le	ngth static structure to <b>store data</b> . One/	print("Good Evening endif	ng!")	CHR (75)	i = random(1,9) r = random(1.1, 7.5)
	1.4 program errors		two D		Switch switch day:		File Handling	
syntax error	occurs when rules of programming are <b>not followed</b>	SQL	Structu	red query language	case 6: print("Saturday case 7:	(°)	f = open("data.tx	Close (") f.close()
logic error	an inaccuracy in the way the program functions	function	Perform	ns a task – <b>does return</b> a value	print("Sunday") default: print("Weekday'		Read Line f.readLine()	Write Line  f.writeLine("Hello")
run-time error	undetected during compilation, but discovered whilst the	procedure	Perform	ns a task – <b>does not</b> return a value	endswitch		End of File	
	program is <b>running</b>	procedure	1 0110111	as a task ascentition a value	Iteration FOR Loop		while NOT f.endOf print f.readLi	
	1.4 data types				for i = 0 to 9		Create a New File	
casting	Changing the data type within the variable		2 Sorting	and searching algorithm	print ("i = " + i) next i		newFile("newdata.	txt")
string /	Alphanumeric characters	linear search		One by one	WHILE Loop  while password != "Pa	assword123"	Sub Programs	
integer \	Whole numbers (no decimal numbers)	binary search		Discards half	password = input( endwhile	'Guess again")	Procedure procedure sum(n1,	Function 12) function sum(n1,n2)
float	Decimal numbers (no decimal point limitation)	bubble sort		Swaps two values	DO WHILE Loop		print(n1 + n2) endprocedure	return(n1 + n2) endfunction
	Synonymous with <b>float</b>	merge sort		Splits into individual values	do password = input(' while password != "Pa		Call a Procedure	Call a Function result = sum(8,9)
Boolean F	Returns only <b>true/false</b>	insertion sort		Places the value in the correct location	winte passiford := "Pa	ounded UIES	Sun(8,9)	resure - Sum(8,9)

Systems Architecture	
1.1 Systems Architect	ure
CPU	Central Processing Unit
Cores	•
	An individual processor within a CPU
Cache	Incredibly fast, but very expensive volatile (temporary) memory close to the CPU
Clock speed	The number of FDE cycles that a CPU can carry out in one second
Levels of cache	L1, L2, L3 – L1 is the fastest and most expensive cache level
Overclocking	Processor running at a higher speed than the manufacturer has recommended
CU	Control Unit
MAR	Memory address register
MDR	Memory data register
PC	Program counter
Accumulator	Small, fast register, used to keep track of the data <b>currently</b> being processed
ALU	Arithmetic Logic Unit
FDE cycle	Basis of the <b>Von Neumann</b> architecture
1.2 Memory and stor	rage
RAM	Primary, volatile internal storage.
ROM	Non-volatile memory, stores the boot up sequence for the computer
Virtual memory	Short-term, allocated when the RAM is at capacity
Primary storage	Volatile storage, used to temporary hold data
Compression	Reducing the file size
Lossy	Data is <b>removed from</b> a file to reduce the file size.
Lossless	redundant data is <b>removed</b> for sending, then <b>replaced</b> upon receipt.
Data capacity	How much data the storage type can hold, measured in bits
Secondary Storage	Permanent, non-volatile methods of keeping data
Features of storage	Capacity, speed, portability, reliability and cost
1.3 Systems Software	
Operating System	controls all the hardware and software for the PC.
User management	Allocation of an account, access rights and security
File management	Naming, allocating to folders, moving <b>files</b> and saving
Utility software	Programs on the computer that help the user keep the computer running
1.3b Functions of the	Onorating System
Functions	The role and responsibilities of the <b>Operating System</b>
MUMPS	An acronym for the functions of the Operating System
Multi-tasking	Allows more than one <b>program</b> to run at the same time
User interface	Windows, menus, icons and a pointing device (WIMP) to assist the user
Memory	Gives over RAM and CPU memory to programs requiring it
Management	, , , , , , , , , , , , , , , , , , , ,
Peripheral	Allowing mice/keyboards/printers to work
Management	-
Security	Keeping data protected from modification/deletion

Devices required to maintain a network Network hardware The Internet A world-wide collection of hardware DNS Domain Naming System translates an IP address into a domain name Housing, maintaining and serving files on a server Hosting Modes of How **devices** are **connected**, for example: Wired, wirelessly or Bluetooth connection Encryption The process of **converting** data into code IP addressing a unique string of numbers separated by full stops Standards Standards that allow hardware/software to interact **Protocols** A set of rules for transmitting data 1.5 Ethical, Legal, Cultural and environmental impacts of digital technology Ethical morally right or wrong when discussing computing Legal Within or outside the confines of law How technology **impacts** on different societies across the globe Cultural **Environmental** Discussing how the **environment** is impacted by technology **Regulation, storing and use** of personally identifiable information Privacy Data Protection Act Updated in 2018 to GDPR, the law governs how people and businesses can use information relating to their clients Released in 1990, the law governs use of other peoples computer and **Computer Misuse** Act outlines the consequences of doing so Released in 1988, the law governs who can use the property of others and Copyright, designs the information that cannot be used as it belongs to the creator and Patents Act 1.6 F-D-E Cycle

Registers

Main Memory

A network involving the client (user machine) sending requests to the

A network where devices are physically connected to each other with an

ALU

3. Process data

4. Execute instruction

server. The server processes the request and sends the data

1.4 Computer networks, connections and protocols

Variations of network

Ethernet cable

Control Unit

2. Decode instruction

1. Fetch instruction

Types of network

**Client- Server** 

Peer-to-peer

	1.1 Systems Software
operating system	The Operating System controls all the <b>hardware</b> and <b>software</b> for the PC.
memory management	Allocates memory to a process when the process requests it and
	deallocates the memory when the process has terminated
user interface	Provides the user with a WIMP (Windows, menus, icons and pointer).
	Can be either graphical or command line
multi-tasking	Allows the user to <b>combine tasks</b> and processes
peripheral management	Installs the <b>drivers</b> for the new hardware so it can be used without error
security	uses password protection to protect user data and prevents
	unauthorized access to programs and user data.
user management	Allocation of an account, access rights and security
file management	Naming, allocating to folders, moving files and saving
utility software	Programs on the computer that help the user keep the computer running smoothly.

1.1b defragmentation	
What?	Why?
Defragmentation > reorganizing the data on the hard drive > related data	repeated access > data becomes fragmented
back together.	speed of access

1.2 Ethical, Legal, Cultural and environmental impacts of digital technology		
ethical	Ethics relates to what is morally right or wrong when discussing	
	computing	
legal	Within or outside the confines of law	
cultural	How technology <b>impacts</b> on different societies across the globe	
environmental	Discussing how the environment is impacted by technology	
privacy	Laws that relate to the <b>regulation, storing and use</b> of personally	
	identifiable information	
Data Protection Act	Updated in 2018, the law governs how people and businesses can use	
	information relating to their clients	
Computer Misuse Act	Released in 1990, the law governs use of other peoples computer and	
	outlines the consequences of doing so	
Copyright, designs and Patents Act	Released in 1988, the law governs who can use the property of others	
	and the information that cannot be used as it belongs to the creator	
software licences	Understand that software can be <b>open source</b> or <b>proprietary</b> .	

1.3 Algo		
abstraction	decomposition	algorithmic thinking
	] → □	000 1111

	1.3 Algorithms
abstraction	The <b>process</b> of <b>removing</b> unnecessary information from a
	problem, so the solution only focusses on the essential
	information
decomposition	The <b>process</b> of <b>breaking down</b> a problem into smaller, more
	manageable chunks known as subroutines
algorithmic thinking	Using logic and reasoning to solve common problems
designing algorithms	Producing <b>flowcharts</b> or <b>pseudocode</b> and identify common
	errors through trace tables
searching algorithms	Binary and linear search algorithms for locating a data
	location using repetition or mathematical calculations
sorting algorithms	Bubble, merge and insertion sorting algorithms used to
	create solutions using logic and pattern recognition

	1.4 Programming fundamentals
variables	A location in memory that can be changed and reassigned
	during the running of the program
constants	A location in memory that cannot be changed throughout
	the running of the program.
operators	Can be either comparison or arithmetic operators
data types	How data is represented in a computer program.
casting	Altering the data type of the information
string manipulation	Altering the case of the characters
file handling	Using commands from within Python to open, read, write
operations	and close to interact with a document
array	Fixed length static structure to store data. Can be one or
	two dimensional
SQL	Structured query language
functions and	Built-in code
procedures	

### Computing Year 9 Cycle 3 Programming in Python

1.1a interpre	eter vs compiler
Python 3.4.2 Shell   File Edit Shell Debug Options Windows Help   Python 3.4.2 (v3.4.2:ab2c023a9432, Oct 6 2014, 22:15:05) tel) ] on win32   Type "copyright", "credits" or "license()" for more inform >>>	Wintitled   File Edit Format Run Options Window Help
Executes one line of code at a time	Translates data into high-level language
Suitable for <b>testing</b>	Compiles all lines of code before executing
Highlights <b>errors</b> in each line, making it easier to	Will not run if an <b>error</b> exists in the code – can be
fix	hard to locate the error
Faster than a <b>compiler</b> as it translates and	Translates and executes entire program
executes at the same time	

1.1b high and low level languages		
Low Level Programming Language  10100180110110101 10011101000011 10101101101011 1001111010101111 10011110110	High Level Programming Language    Sale_price = 1.66   (I(sale_price > 2) ( discount = 0.1 )     I sale_price > 2) ( discount = 0.05 )     Processing time	
written to be applied to the system architecture	Close to human language	
very close to <b>binary</b>	often called <b>pseudocode</b>	
hard for humans to understand	easy to modify as it uses <i>English-like</i> statements	
contains <b>few recognisable</b> English words	easy to <b>debug</b>	
fast to <b>execute</b>	portable – can run on many different machines	
examples include: assembly language, machine code, binary	examples include: C++, Java, Pascal, Python, Visual Basic	

1.2 testing				
test data	the data selected for analysis			
boundary/ extreme	data the is accepted, but is <b>close to</b> the required checking area			
valid	possible data that the program should accept and process			
erroneous	data that the program can not process and should not accept			
input validation	a test to ensure the <b>correct data</b> type has been inserted			
iterative	carried out while a program is <b>being developed</b> The process			
	repeats (iterates) until the module works as intended.			
final	program is tested <b>as a whole</b> to ensure that it functions			

1.3 maintainability				
maintainability	allows edits and updates of created programs easily			
debug	locate and resolve an <b>error</b>			
comments	provide <b>additional information</b> , ignored by the program			
sensible variable	pertaining to the <b>data type or function</b> of the variable			
names				
indentation	formatting to show which lines of code are linked			

1.4 program errors				
syntax error	occurs when rules of programming are not followed			
logic error	an inaccuracy in the way the program functions			
run-time error	undetected during compilation, but discovered whilst the program is <b>running</b>			

1.5 logic gates			
Symbol	Logic gate	description	
	and	output is 1 when both inputs are 1	
$\rightarrow$	or	output is 1 when 1 of the inputs are 1	
$\rightarrow$	not	output is 1 when input is 0. Contains an invertor	
invertor		Circle on the not gate, <b>inverts</b> (flips) the input to form the output	

# OCR ERL Cheatsheet



# Commenting

Comment

To String

//my note to me

# Variables

int("13") To Integer str(36)

To Float

Assignment

36 myAge =

float("3.14")

Constants

const pi = 3.14

real("3.14")

ro Real

Global Variables

m global lives =

bool("True")

To Bool

# Comparison Operators

Arithmetic Operators

ednal //ednal //less /not 36 0 health < 1 myAge == lives !=

//subtract //multiply /exponent

//add

+

//modulus //quotient \* 4 ^ 3 MOD DIV equal

/divide

9

# Logical Operators

age > 18 AND age < 60 < 9 0R hour > hour

### Input

Input/Output

a password") pwd = input("Please enter

### Output

print("You have logged in succesfully")

# Selection

# If-Then-Else

print("Good Afternoon!") print("Good Morning!") Evening!") 18) then if hour < 12 then elseif (hour < print("Good endif

### Switch

print("Saturday") print("Weekday") print("Sunday") switch day: default: case 7: case 6: endswitch

## Iteration

### FOR Loop

6 П ç ("i 0 for i = print next i

# WHILE LOOP

input("Guess again") while password != "Password123" password = endwhile

# DO WHILE LOOP

input("Guess again") "Password123" while password != password = မ

ednal //greater //less or marks >= 80 //greater 40 0 score > marks

or

== "Sunday" NOT day

# String Operations

# String Length

Declaration

name.length

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

# Substrings

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

# Concatenation

"Ninja01"

23

score[0] = users[1,3]

Assignment

name) print("Hi"

# Change Case

len(score)

Length

name.upper name.lower

Random Numbers

Random Numbers

# ASCII Conversion

CHR (75) ASC(X)

.5

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

# File Handling

Open

f = open("data.txt")

f.close() Write Line

Close

### Read Line

f.readLine()

f.writeLine("Hello")

### End of File

while NOT f.endOfFile() print f.readLine() endwhile

# Create a New File

newFile("newdata.txt")

# Sub Programs

# Procedure

procedure sum(n1,n2) endprocedure print(n1

# Call a Procedure

sum(8,9)

### Function

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

# Call a Function

result = sum(8,9)