MINISTRY OF EDUCATION, SINGAPORE
in collaboration with
CAMBRIDGE ASSESSMENT INTERNATIONAL EDUCATION
General Certificate of Education Ordinary Level

COMPUTING

7155/02

Paper 2 Lab-based

For examination from 2025

SPECIMEN INSERT

2 hours 30 minutes

Quick Reference Guide for Python

This quick reference guide shows some examples of the Python language constructs. The complete Python language is not limited to these examples.

This document consists of **4** printed pages.

Singapore Examinations and Assessment Board

Cambridge Assessment International Education

1 Identifiers

When naming variables, functions and modules, the following rules must be observed:

- Names should begin with character 'a'–'z' or 'A'–'Z' or '_' and followed by alphanumeric characters or '_'.
- Reserved words should not be used.
- User-defined identifiers are case sensitive.

2 Comments and Documentation Strings

```
# This is a comment
"""
This is a documentation string over
multiple lines
"""
```

3 Input/Output

```
s = input("Prompt for data: ")

print("This is a string")

f = open("input.txt", "r")
line = f.readline()
character = f.read(1)
f.close()

with open("output.txt", "w") as f:
    f.write("Output Line\n")
```

4 Import

```
import <module>
from <module> import <name>
```

5 Data Types

Type	Example	Notes
int	`-3`	integer
float	`3.1415926`	real number
bool	`True`	boolean
str	`"Hello"`	string (immutable)
list	`[2, 3, 5]`	series of values
dict	`{'key':'value'}`	key-value pairs

6 Assignment

Statement	Notes
`a = 1`	normal assignment
`b += c`	augmented assignment equivalent to `b = b + c`
`x[y] = z`	assigns `z` to index `y` of list `x` or assigns `z` to key `y` of dictionary `x`
`del a`	deletes variable `a`
`del x[y]`	deletes key `y` and its value from dictionary `x`

7 Arithmetic Operators

Operator	Notes
`+ -`	add, subtract
`* /`	multiply, divide
`%`	remainder or modulus
`**`	exponential or power
`//`	floor division

8 Relational Operators

Operator	Notes
`==`	equal to
`!=`	not equal to
`> >=`	greater than, greater than or equal to
`< <=`	less than, less than or equal to

9 Boolean Expressions

Boolean Expression	Notes
`a and b`	logical and
`a or b`	logical or
`not a`	logical not

10 Sequence (List/String) Operations

Operator	Notes
`<seq> + <seq>`	concatenation
`<int> * <seq>`	repetition
`<seq>[index]`	indexing
`<seq>[start:stop]`	slicing
`<seq>[start:stop:step]`	slicing with step
`<value> in <seq>`	membership testing

11 Selection

Type 1

```
if condition(s):
    <statement(s)>
```

Type 2

```
if condition(s):
    <statement(s)>
else:
    <statement(s)>
```

Type 3

```
if condition(s):
    <statement(s)>
elif condition(s):
    <statement(s)>
else:
    <statement(s)>
```

12 Iteration

while loop

```
while condition(s):
    <statement(s)>
```

for loop

```
for i in range(n):
    <statement(s)>

for record in records:
    <statement(s)>
```

13 Functions

```
# Function definition
def <function name> (<parameters>):
    <function body>
    return <return value>

# Function call
<function name>(<arguments>)
```

14 Built-in Functions

(a) Basic Functions

abs()	chr()	float()	input()
int()	len()	max()	min()
open()	ord()	print()	range()
round()	str()	<file>.close()	<file>.read()
<file>.readline()	<file>.write()	<str>.endswith()	<str>.find()
<str>.format()	<str>.isalnum()	<str>.isalpha()	<str>.isdigit()
<str>.islower()	<str>.isspace()	<str>.isupper()	<str>.lower()
<str>.split()	<str>.startswith()	<str>.upper()	

(b) Math Module

ceil()	floor()	pow()	sqrt()	trunc()

(c) Random Module

randint()	random()

15 Reserved Words

Reserved words are part of the syntax of the language. They cannot be used as identifiers.

False	None	True	and	as
assert	break	class	continue	def
del	elif	else	except	finally
for	from	global	if	import
in	is	lambda	nonlocal	not
or	pass	raise	return	try
while	with	yield		