



MINISTRY OF EDUCATION, SINGAPORE
in collaboration with
CAMBRIDGE ASSESSMENT INTERNATIONAL EDUCATION
General Certificate of Education Advanced Level
Higher 2



COMPUTING

Paper 2 (Lab-based)

INSERT

9569/02

October/November 2020

3 hours



This document consists of **7** printed pages and **1** blank page.



Singapore Examinations and Assessment Board



**Cambridge Assessment
International Education**

1 Python

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

```
print ("This is a string")
```

```
s = input ("Instructions to prompt for data entry.")
```

4 Import

```
import <module>
```

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

5 Data Type

Data Type	Notes
int	integer
float	real number
bool	boolean
str	string (immutable)
list	series of values
dict	key-value pairs
tuple	series of values (immutable)

6 Assignment

Assignment Statement	Notes
a = 1	integer
b = c	variable
d = "This is a string"	string
mylist = [1, 2, 3, 4, 5]	list
mydict = {'key': 'value'}	dict

7 Arithmetic Operators

Operator	Notes
+ -	plus, subtract
* /	multiply, divide
%	remainder or modulus
**	exponential or power
//	quotient of the floor division

8 Relational Operators

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

9 Boolean Expression

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



10 Iteration

while loop	for loop
while condition(s): <statement(s)>	for i in range(n): <statement(s)> for record in records: <statement(s)>

11 Selection

Type 1	Type 2	Type 3
if condition(s): <statement(s)>	if condition(s): <statement(s)> else: <statement(s)>	if condition(s): <statement(s)> elif condition(s): <statement(s)> else: <statement(s)>

12 Functions

Function definitions

@<optional decorator(s)>

def <function name> (<parameters>):
 <function body>

Function calls

<function name>(<value>, <name>=<value>)

13 Object-Oriented Programming

class <class name> (<optional parent class>):

def __init__(**self**, <parameters>):
 <constructor body>

def <method name> (**self**, <parameters>):
 <method body>



14 Built-in Functions and Attributes

<code>__file__</code>	<code><file>.readlines()</code>	<code><list>.copy()</code>	<code>print()</code>	<code><str>.isdigit()</code>
<code>__name__</code>	<code><file>.write()</code>	<code><list>.index()</code>	<code>range()</code>	<code><str>.islower()</code>
<code>abs()</code>	<code>float()</code>	<code><list>.insert()</code>	<code>round()</code>	<code><str>.isspace()</code>
<code>bin()</code>	<code>hex()</code>	<code><list>.pop()</code>	<code>staticmethod()</code>	<code><str>.isupper()</code>
<code><bytes>.decode()</code>	<code>input()</code>	<code><list>.remove()</code>	<code>str()</code>	<code><str>.lower()</code>
<code>chr()</code>	<code>int()</code>	<code><list>.reverse()</code>	<code><str>.encode()</code>	<code><str>.startswith()</code>
<code><dict>.clear()</code>	<code>len()</code>	<code><list>.sort()</code>	<code><str>.endswith()</code>	<code><str>.upper()</code>
<code><dict>.copy()</code>	<code>list()</code>	<code>max()</code>	<code><str>.format()</code>	
<code><file>.close()</code>	<code><list>.append()</code>	<code>min()</code>	<code><str>.index()</code>	
<code><file>.read()</code>	<code><list>.extend()</code>	<code>open()</code>	<code><str>.isalnum()</code>	
<code><file>.readline()</code>	<code><list>.clear()</code>	<code>ord()</code>	<code><str>.isalpha()</code>	

csv module	datetime module	math module
<code>reader()</code> <code>writer()</code> <code><writer>.writerow()</code>	<code>datetime()</code> <code>datetime.now()</code> <code>datetime.strptime()</code> <code><datetime>.isoformat()</code> <code><datetime>.strftime()</code> <code><datetime>.year</code> <code><datetime>.month</code>	<code><datetime>.day</code> <code><datetime>.hour</code> <code><datetime>.minute</code> <code><datetime>.second</code> <code><timedelta>.days</code> <code><timedelta>.seconds</code> <code>ceil()</code> <code>exp()</code> <code>floor()</code> <code>log()</code> <code>pow()</code> <code>sqrt()</code> <code>trunc()</code>

os.path module	random module	sqlite3 module	socket module	sys module
<code>basename()</code> <code>dirname()</code> <code>isdir()</code> <code>isfile()</code> <code>join()</code>	<code>random()</code> <code>randint()</code> <code>randrange()</code> <code>shuffle()</code>	<code>connect()</code> <code><connection>.commit()</code> <code><connection>.close()</code> <code><connection>.execute()</code> <code><connection>.rollback()</code> <code><connection>.row_factory</code> <code><cursor>.fetchone()</code> <code><cursor>.fetchall()</code> <code>Row</code>	<code>socket()</code> <code>bind()</code> <code>listen()</code> <code>accept()</code> <code>connect()</code> <code>recv()</code> <code>sendall()</code>	<code>exit()</code>

15 Additional Functions and Attributes

pymongo module	flask module
<code>MongoClient()</code> <code><client>.database_names()</code> <code><client>.get_database()</code> <code><client>.drop_database()</code> <code><client>.close()</code> <code><database>.collection_names()</code> <code><database>.get_collection()</code> <code><database>.drop_collection()</code> <code><collection>.insert_one()</code> <code><collection>.insert_many()</code> <code><collection>.find_one()</code> <code><collection>.find()</code>	<code>Flask()</code> <code><flask application>.route()</code> <code><flask application>.run()</code> <code>render_template()</code> <code>request.files</code> <code>request.form</code> <code>request.method</code> <code>send_from_directory()</code> <code>redirect()</code> <code>url_for()</code> <code>secure_filename()</code> <code><uploaded file>.save()</code>



2 SQL Statements

CREATE TABLE <i>table_name</i> (<i>column1_name</i> COLUMN1_TYPE COLUMN1_CONSTRAINTS , <i>column2_name</i> COLUMN2_TYPE COLUMN2_CONSTRAINTS , ... PRIMARY KEY (<i>column1_name</i> , <i>column2_name</i> , ...), FOREIGN KEY (<i>column_name</i>) REFERENCES <i>table_name</i> (<i>column_name</i>));	
SELECT <i>column1_name</i> , <i>column2_name</i> , ... FROM <i>table_name</i> WHERE <i>where_expression</i> ORDER BY <i>order_expression</i> ASC ;	SELECT <i>column1_name</i> , <i>column2_name</i> , ... FROM <i>table_name</i> WHERE <i>where_expression</i> ORDER BY <i>order_expression</i> DESC ;
SELECT <i>table1_name.column1_name</i> , <i>table2_name.column2_name</i> , ... FROM <i>table_name</i> , <i>table2_name</i> WHERE <i>where_expression</i> ;	
SELECT <i>table1_name.column1_name</i> , <i>table2_name.column2_name</i> , ... FROM <i>table1_name</i> INNER JOIN <i>table2_name</i> ON <i>join_expression</i> ;	
SELECT <i>table1_name.column1_name</i> , <i>table2_name.column2_name</i> , ... FROM <i>table1_name</i> LEFT OUTER JOIN <i>table2_name</i> ON <i>join_expression</i> ;	
SELECT <i>COUNT</i> (*), <i>MAX</i> (<i>column1_name</i>), <i>MIN</i> (<i>column2_name</i>), <i>SUM</i> (<i>column3_name</i>), ... FROM <i>table_name</i> ;	
INSERT INTO <i>table_name</i> (<i>column1_name</i> , <i>column2_name</i> , ...) VALUES (<i>column1_value</i> , <i>column2_value</i> , ...);	
UPDATE <i>table_name</i> SET <i>column1_name</i> = <i>column1_expression</i> , <i>column2_name</i> = <i>column2_expression</i> , ... WHERE <i>where_expression</i> ;	
DELETE FROM <i>table_name</i> WHERE <i>where_expression</i> ;	
DROP TABLE <i>table_name</i> ;	

3 SQLite Types, Constraints, Functions and Operators

Types	Constraints	Functions	Operators			
NULL	NOT NULL	COUNT()		/	<	AND
REAL	PRIMARY KEY	MAX()	+	%	<=	OR
INTEGER	AUTOINCREMENT	MIN()	-	=	>	IS
TEXT	UNIQUE	SUM()	*	!=	>=	IS NOT



4 PyMongo Operators

Comparison

\$eq	\$gt	\$gte	\$lt	\$lte
\$ne	\$in	\$nin		

Logical

\$and	\$not	\$or
-------	-------	------

Element

\$exists

Update

\$set	\$unset
-------	---------

5 HTML Elements, Attributes and Character References

The first line of a HTML document must be: <!doctype html>

Type	Elements	Attributes
<i>Common</i>		id, class
<i>Required</i>	<html>, <head>, <title>, <body>	
<i>Metadata</i>	<link>	rel, href
<i>Structure</i>	<h1>, <h2>, <h3>, <p>, <div>, , <hr>	
<i>Text and Media</i>	, <i>	
	<a>	href
		src, alt
<i>Table</i>	<table>, <tr>, <th>, <td>	
<i>Form</i>	<form>	action, enctype, method
	<input>	name, type, value
	<textarea>	name

Character	&	<	>	"
Reference	&	<	>	"

6 Jinja2 Filters

length	safe
--------	------



7 CSS Properties

Common	Box Model		Typography
display background color	height width border border-bottom border-left border-right border-top margin margin-bottom	margin-left margin-right margin-top padding padding-bottom padding-left padding-right padding-top	font-family font-size font-style font-weight text-align text-decoration



BLANK PAGE

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

Cambridge Assessment International Education is part of the Cambridge Assessment Group. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is itself a department of the University of Cambridge.

