MINISTRY OF EDUCATION, SINGAPORE
in collaboration with
UNIVERSITY OF CAMBRIDGE LOCAL EXAMINATIONS SYNDICATE
General Certificate of Education Advanced Level
Higher 2

COMPUTING

9569/02

Paper 2 (Lab-based)

Practice Session

SPECIMEN PAPER

1 hour

Additional Materials:	Electronic version of `MONITORS.txt` data file
	Electronic version of `PRINTERS.txt` data file
	Electronic version of `LAPTOPS.txt` data file
	Electronic version of `TASK3stack.txt` data file
	Electronic version of `TASK3queue.txt` data file
	Electronic version of `STUDENTLIST.txt` data file

READ THESE INSTRUCTIONS FIRST

Answer Task 3, Task 4 and supplementary task.

All tasks must be done in the computer laboratory. You are not allowed to bring in or take out any pieces of work or materials on paper or electronic media or in any other form.

Approved calculators are allowed.

Save each task as it is completed.

The use of built-in functions, where appropriate, is allowed for this paper unless stated otherwise.

The number of marks is given in brackets [ ] at the end of each question or part question.
The total number of marks for this paper is 61.

This document consists of **6** printed pages and **1** blank page.

Singapore Examinations and Assessment Board

CAMBRIDGE
International Examinations

3 Name your Jupyter Notebook as
`Task3_<your name>_<centre number>_<index number>.ipynb`

A programmer is writing a program to manipulate different data structures using Object-Oriented Programming (OOP).

The superclass, `DataStructure`, will store the following data:

- a linked list of the data items
- head pointer, pointing to the first element in the linked list
- tail pointer, pointing to the last element in the linked list

This class has one method to display all the current contents in the structure, in the order they are stored in the linked list.

The superclass is used to implement a stack and a linear queue.

The subclass `Stack` has the following methods:

- `insert(value)` appends the parameter to the stack.
- `delete()` returns and removes the next value in the stack.
- `display()` method should display the stack in reverse order (e.g. the most recently added element first) and should override the `DataStructure` display method.

The subclass `Queue` has the following methods:

- `insert(value)` appends the parameter value to the queue.
- `delete()` returns and removes the next value in the queue.
- `display()` method should display the queue contents in order (e.g. the earliest added element first) and should override the `DataStructure` display method.

Each method updates its appropriate pointers, and produces suitable errors (or returns different values) to indicate if the actions are not possible, e.g. if the structure is empty.

For each of the sub-tasks, add a comment statement, at the beginning of the code using the hash symbol '#', to indicate the sub-task the program code belongs to, for example:

```
In [1]:  #Task 3.1
         Program code
```

```
In [2]:  #Task 3.2
         Program code
```

```
In [3]:  #Task 3.3
         Program code
```

```
In [4]:  #Task 3.4
         Program code

         Output:
```

Task 3.1

Write program code for the superclass `DataStructure`. [3]

Task 3.2

Write program code for the subclass `Stack`.

Use appropriate inheritance and polymorphism in your designs. [5]

Task 3.3

Write program code for the subclass `Queue`.

Use appropriate inheritance and polymorphism in your designs. [5]

Task 3.4

The files `TASK3stack.txt` and `TASK3queue.txt` store data to test your program.

Write program code to:

- create a new stack and add the data in the file `TASK3stack.txt` to the stack
- create a new queue and add the data in the file `TASK3queue.txt` to the queue
- output the current contents of both the stack and queue
- remove and output two items from both the stack and queue
- output the contents of both the stack and queue after the removal of the items.

All outputs should have appropriate messages to indicate what they are showing.

You are required to present the output of stack and queue both before and after the removal of items.

Save your Jupyter Notebook for Task 3.

[9]

4 A large company currently keeps records on paper of all the computing equipment it owns. Every computer device has its information recorded when it is purchased.

The company decided to trial a database to manage its computing equipment records. It is expected that the database should be normalised.

When a computer device is purchased, the following information is recorded:

- SerialNumber – unique serial number of device
- Type – type of device ('Monitor', 'Laptop' or 'Printer')
- Model – model of device
- Location – where the device is used
- DateOfPurchase – date of purchase
- WrittenOff – whether the device is still in use ('TRUE' means device is written off and NOT in use, 'FALSE' means device is still in use)

For monitors, the following extra information is recorded:

- DateCleaned – the last date the monitor was cleaned

For laptops, the following extra information is recorded:

- WeightKg – the weight in kilograms

For printers, the following extra information is recorded:

- Toner – type of toner required
- DateChanged – the last date the toner cartridge was changed

The information is to be stored in four different tables:

```
Device
Monitor
Laptop
Printer
```

Task 4.1

Create an SQL file called TASK4_1_<your name>_<centre number>_<index number>.sql to show the SQL code to create the database equipment.db with the four tables. The table, Device, must use SerialNumber as its **primary key**. The other tables must refer to the SerialNumber as a **foreign key**.

Save your SQL code as
TASK4_1_<your name>_<centre number>_<index number>.sql [5]

Task 4.2

The files `MONITORS.txt`, `LAPTOPS.txt` and `PRINTERS.txt` contain information about the company's monitors, laptops and printers respectively for insertion into the equipment database. Each row in the three files is a comma-separated list of information about a single device.

For `MONITORS.txt`, information about each monitor is given in the following order:
`SerialNumber,Model,Location,DateOfPurchase,WrittenOff,DateCleaned`

For `LAPTOPS.txt`, information about each laptop is given in the following order:
`SerialNumber,Model,Location,DateOfPurchase,WrittenOff,WeightKg`

For `PRINTERS.txt`, information about each printer is given in the following order:
`SerialNumber,Model,Location,DateOfPurchase,WrittenOff,Toner,DateChanged`

Write a Python program to insert all information from the three files into the `equipment` database, `equipment.db`. Run the program.

Save your program code as
`TASK4_2_<your name>_<centre number>_<index number>.py` [5]

Task 4.3

Write SQL code to show the serial number, model and the location of each monitor, with the date it was last cleaned. Run this query.

Save this code as
`TASK4_3_<your name>_<centre number>_<index number>.sql` [4]

Task 4.4

The company wants to filter the devices by `Location` and display the results in a web browser.

Write a Python program and the necessary files to create a web application that:

- receives a `Location` string from a HTML form, then
- creates and returns a HTML document that enables the web browser to display a table tabulating the `SerialNumber` and `Type` of devices still in use at that exact `Location`

Save your Python program as
`TASK4_4_<your name>_<centre number>_<index number>.py`
with any additional files / sub-folders as needed in a folder named
`TASK4_4_<your name>_<centre number>_<index number>`

Run the web application. Save the output of the program when "Office 51" is entered as the `Location` as `TASK4_4_<your name>_<centre number>_<index number>.html` [10]

Supplementary Task

A teacher wants to create records of students with some remarks in a NoSQL database. The list of students is saved in the file `STUDENTLIST.txt`.

Each line is in the format:

```
class,index_no,name
```

- `class` is in the format `20SNN`, where `N` is a digit, for example, `20S01`
- `index_no` is an integer greater than or equal to 1
- `name` is made up of letters only

Task 5.1

Write program code to insert the data from the file `STUDENTLIST.txt` into a NoSQL database `all_classes` under the collection `student_details`. The program should clear the `student_details` collection if it exists.

Save your Python program as

`TASK5_1_<your name>_<centre number>_<index number>.py` [5]

Task 5.2

The teacher wants to add student remarks for the students through a program.

Write a Python program to:

- ask the teacher to key in a class
- list out the students in the class, giving their index number followed by name
- let the teacher key in the index number of the student to add remarks for
- allow the teacher to type in the student remarks to be added
- add the student remarks to the correct student record in the `student_details` collection. If there are previous student remarks, those remarks will be overwritten.
- ask the teacher whether to continue adding remarks for more students. If yes, repeat the process, otherwise list all the entries in the `student_details` collection and end the program.

Assume all inputs are correct.

Save your Python program as

`TASK5_2_<your name>_<centre number>_<index number>.py` [10]

BLANK PAGE