

Test
Semester 1 2020

ISCG 5421 Programming Principles & Practice

School of Computing,
Electrical and Applied
Technology

Date: Wednesday 10 June 2020
Commencement Time: 3:00 p.m.
Time Allowed: 90 minutes *plus* 10 minutes reading time
Weighting: 20%
Marks: 50

This test covers the following learning outcomes:

1.	Implement the designs by writing well-structured programs that follow enforced programming language conventions and programming standards.
2.	Identify the fundamental data requirements of an intermediate-level program.
3.	Apply the logic structures of the language.
4.	Select and use intermediate-level data structures

Instructions:

1. This is an open-book practical test.
2. You may use your textbook and any resources available from the Internet.
3. This is an individual test and you must NOT communicate with other students by ANY means (e.g. Zoom, Moodle, Discord, Email, etc).
4. Read the scenario given on page 2. You must answer all the questions.

In the midst of the COVID-19 pandemic, the New Zealand government has hired you to develop an initial prototype of software used for contact tracing. The program has the following requirements:

- A class of type Person.
- A *constructor/initializer* which takes a private **email** address and saves it to the object instance as a private/hidden field.
- A public **last_visit_at** property which is initially set to the date & time that the object was created.
- A public **locations** property which keeps track of **location_id**'s (integers) a person has visited. You need the ability to check if these locations overlap/intersect with **locations** visited by another person. Select an appropriate data structure for this.
- A public method called *visit()* which takes a **location_id** and adds it to **locations**. Create an InvalidLocation exception and raise it if the **location_id** provided is less than zero. If the user successfully visits a location without raising an exception, then the **last_visit_at** property should be updated to the current date & time.
- A public method called *has_contact()* which takes **other_person** (a Person object) and returns True if they have both visited the same location, and False if they have not. NOTE: It's important that here you have chosen the correct data structure.
- A private method called *notify()* which returns a string saying 'Notifying user X', where X is the person's **email**. NOTE: The email must be printed in all upper-case.
- A public method called *num_locations_visited()* which returns the amount of locations a person has visited.
- Your program must adhere to Python's conventions. Run pylint on your code once you've finished it, and try to get a 10/10 rating.

See the following page for questions:

TEST QUESTIONS

50 Marks

Question A – Class Diagram

[15 marks]

You are required to create a class diagram for the above requirements. The class name is underlined, property names are in bold and method names are in italic. Ensure that you use the appropriate prefixes for properties.

NOTE: You don't need to draw the `InvalidException` class on this diagram.

Save the class diagram as either a jpg or png file, and add it to your repository.

Question B – Implementation

[20 marks]

In `contact_tracer.py`, implement the code from the previous section. Ensure that you use the appropriate prefix (an underscore) for any private/hidden fields.

Make sure you check your code with `pylint`.

Question C – Testing

[15 marks]

Create a file called `contact_tracer_tests.py`.

Using either Python's built-in `unittest` module or `pytest`, create tests for the following. (NOTE: If you forget how to do unit testing, just add your code to your program's entry point section, i.e. the `if __name__ == '__main__':` part.)

Create the following tests:

- A test to ensure that the `visit()` method adds **`location_id`** to **`visited`**.
- A test to ensure that the `visit()` method raises an `InvalidLocation` exception if the provided **`location_id`** is less than zero.
- A test to ensure that the `num_locations_visited()` method returns 0 if the user has visited no locations.
- A test to ensure that the `num_locations_visited()` method returns the correct amount of locations when a person visits more than 0 locations.
- A test to ensure that the `has_contact()` method returns True if two users have visited the same location.
- A test to ensure that the `has_contact()` method returns False if two users have not visited the same location.
- A test to ensure that the `notify()` method returns the person's email address in all capital letters.