University of Strathclyde Glasgow

Department of Computer & Information Sciences

CS995 Introduction to Programming Principles

2023/2024

Practice Exam
Duration: 3 hours

Attempt All Questions (Total 100 Marks)

General instructions

This is an open-book individual programming exam. Students are allowed to access teaching material that is hosted on MyPlace during the exam. External web pages are not available during the exam. Students are not allowed to communicate with other students or anyone else during the exam, following standard exam conditions.

Marking Criteria

- Implementation - each question has an associated number of marks, which correspond to a successful implementation that matches the question. **(90 marks)**
- Commenting - commenting is used appropriately, following examples given in the teaching material. **(5 marks)**
- Style - PEP8 style compliance and naming conventions. **(5 marks)**

Submission

Software source code must be submitted using the MyPlace submission link that is associated with this exam. Source code must be submitted as a single zip file named “solution.zip”. The zip file and the source code must not include personal identifiable data such as the student name or registration number. The source code must be submitted during the time that is associated with the exam. Submissions after the exam has ended will not be accepted.

Q.1 Create a class named `State` in a file named `electric_network.py`. Create a constructor for the `State` class that accepts:

- `id` - integer, default 0.
- `name` - string, default empty string.

The constructor should assign input values to data members of the same name.

(2 marks)

Q.2 Create an `__repr__` function for the `State` class. The `__repr__` function should return a string that can be evaluated to create a duplicate object.

(3 marks)

Q.3 Create a class named `UnitType` in the file named `electric_network.py`. Create a constructor for the `UnitType` class that accepts:

- `id` - integer, default 0.
- `name` - string, default empty string.
- `current` - float, default 0.0.

The constructor should assign input values to data members of the same name.

(2 marks)

Q.4 Create an `__repr__` function for the `UnitType` class. The `__repr__` function should return a string that can be evaluated to create a duplicate object.

(3 marks)

Q.5 Create a class named `Element` in the file named `electric_network.py`. Create a constructor for the `Element` class that accepts:

- `id` - integer, default 0.
- `unit_type` - UnitType object, default None.
- `state` - State object, default None.
- `child_elements` - List of Element objects, default empty list.

The constructor should assign input values to data members of the same name. The `child_elements` data member should be assigned a shallow copy of the input `child_elements`.

(5 marks)

Q.6 Create an `__repr__` function for the `Element` class. The `__repr__` function should return a string that can be evaluated to create a duplicate object.

(5 marks)

Q.7 Create a `total_current` function for the `Element` class. The `total_current` function should return 0 if the `state.name` is not "On". If the `Element` object has no child elements, then `total_current` should return the `unit_type.current`. If the `Element` object has child elements, the function return an overall total by calling the `total_current` function of the child elements.

(10 marks)

Q.8 Create a `find_max_load` function for the `Element` class. The function should return `None` if the `Element` object has no child elements. The function should return the `Element` object of the child object that has the biggest current.

(10 marks)

Q.9 Create a function named `load_states` in the file named `electric_network.py` that accepts:

- `json_data` - a list of dictionaries that have keys "id" and "name".
- `states` - a dictionary that contains a key of `state.id` and a value of a `State` object.

An example `json_data` value is given below, which is taken from the `electric_network.json` file.

```
[
    {
        "id": 1,
        "name": "On"
    }
]
```

The function should clear the `states` dictionary, create `State` objects using the information in `json_data` and assign each object to the `states` dictionary.

(10 marks)

Q.10 Create a function named `load_types` in the file named `electric_network.py` that accepts:

- `json_data` - a list of dictionaries that have keys “id”, “name” and “current”.
- `unit_types` - a dictionary that contains a key of `unit_type.id` and a value of a `UnitType` object.

An example `json_data` value is given below, which is taken from the `electric_network.json` file.

```
[
    {
        "id": 1,
        "name": "Supply controller",
        "current": 0
    }
]
```

The function should clear the `unit_types` dictionary, create `UnitType` objects using the information in `json_data` and assign each object to the `unit_types` dictionary.

(10 marks)

Q.11 Create a function named `load_elements` in the file named `electric_network.py` that accepts:

- `json_data` - a list of dictionaries that have keys “id”, “type_id”, “state_id”, “element_ids”.
- `states` - a dictionary that contains a key of `state.id` and a value of a `State` object.
- `unit_types` - a dictionary that contains a key of `unit_type.id` and a value of a `UnitType` object.
- `elements` - a dictionary that contains a key of `element.id` and a value of an `Element` object.

An example `json_data` value is given below, which is taken from the `electric_network.json` file.

```
[
    {
        "id": 1,
        "type_id": 1,
        "state_id": 1,
        "element_ids": [
            2,
            3,
            4,
            5
        ]
    }
]
```

The function should clear the `elements` dictionary, create `Element` objects using the information in `json_data` and assign each object to the `elements` dictionary. The function does not need to access the values that are associated with “element_ids”.

(10 marks)

Q.12 Create a function named `find_child_elements` in the file named `electric_network.py` that accepts:

- `json_data` - a list of dictionaries that have keys “id”, “type_id”, “state_id”, “element_ids”.
- `elements` - a dictionary that contains a key of `element.id` and a value of an `Element` object.

The function should find the child elements and append them to the parent’s `child_elements` list.

(15 marks)

Q.13 Write a program that loads the data given in `electric_network.json` and prints:

- The element name and id.
- The total current for the element.
- The name of the unit type from the child element that has the biggest current.
- The id of the child element that has the biggest current.

(15 marks)

END OF PAPER

(Dr. W. H. Bell)