Quiz #2: Inheritance

1. Here is code for a class `Roster`, used to keep track of records (stored as a dictionary of tuples). Read this, and then read the context and code for subclass `EmployeeRoster` on the back of this sheet. Your task will be to implement the `__str__` method to `EmployeeRoster`.

```python
class Roster:
    """Represents a roster of members."

    """Attributes"""
    @type members: dict
    The members for this roster.

    def __init__(self):
        """Initialize a Roster."

        @type self: Roster
        @rtype: None
        self.members = {}

    def add(self, member):
        """Add <member> to the roster."

        @type self: Roster
        @type member: tuple
        A tuple that represents a member.
        Precondition: member[0] not in self.members
                     len(member) > 1

        @rtype: None
        self.members[member[0]] = member[1:]

    def remove(self, member):
        """Remove <member> from the roster."

        @type member: tuple
        A tuple that represents a member.
        Precondition: len(member) > 0

        @rtype: None
        self.members.pop(member[0], default=None)

    def __str__(self):
        """Return a string representation of the roster."

        @type self: Roster
        @rtype: str
        raise NotImplemenetedError()
```

**Context:** an employee records system.

An employee roster keeps track of employees that work at a grocery store. Each employee record consists of a surname, a given name, and a unique employee identification number. Employees may join or leave the store, in which case they must be added or dropped from the employee roster. We need to be able to display all employees in the store in the following format:

Surname: Hopper, Given name: Grace, Employee number: 906  
Surname: Bond, Given name: James, Employee number: 7  
...
Surname: <surname>, Given name: <given name>, Employee number: <employee number>

You may make any reasonable assumption about the order of data in each key-value pair in the `members` attribute.

class EmployeeRoster(Roster):  
    """A roster of employees.  
    """  
    @type members: dict  
    inherited from Roster  
    """  
    def __init__(self):  
        """Initialize a EmployeeRoster."""  
        pass  
    # TODO: Implement __str__ for this class.  
    def __str__(self):

2. Read the following use of `EmployeeRoster`.

(a) As currently written, there is a bug in the `EmployeeRoster` class. Circle the line in the following three lines of code that will throw an exception, and state what error will be raised.

```
roster = EmployeeRoster()  
roster.add(('945', 'Alan', 'Turing'))  
str(roster)
```

(b) How would you fix the bug?