INSTRUCTIONS

• You have 10 minutes to complete this quiz.
• The exam is closed book, closed notes, closed computer, closed calculator.
• Mark your answers on the exam itself. We will not grade answers written on scratch paper.
• For multiple choice questions, fill in each option or choice completely.
  – □ means mark all options that apply
  – ○ means mark a single choice

<table>
<thead>
<tr>
<th>Last name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>First name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Student ID number</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CalCentral email (@berkeley.edu)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Discussion Section</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>__ __ __</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>All the work on this exam is my own. (please sign)</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0. Your thoughts? What makes you strong?
1. Oops! ... I Did It Again

(a) Suppose Britney wants to define a `Person` class.

```python
class Person:
    name = None
    def __init__(self, name):
        Person.name = name
    def greet(self):
        return 'Hello, my name is ' + self.name
```

John, however, sees a problem. Mark all appropriate criticisms of this implementation.

- Every `Person`’s name will be equal to the most recently-created `Person`’s name.
- Instantiating a `Person` will cause an error.
- Every `Person`’s name will be `None`.
- Invoking `greet` on a person instance will cause an error.

(b) Consider the following simple class definition.

```python
class Dog:
    def bark(self):
        print('woof!')
```

One day, while using this class, Britney decides she wants her dog, Lacey, to bark differently:

```python
>>> lacey = Dog()
>>> lacey.bark = 'bow wow'
```

Paul quickly points out that this won’t work. “`bark` is supposed to be a method, not a string!” So Britney attempts to reset the `bark` method to what it was before:

```python
>>> lacey.bark = Dog.bark
```

Paul isn’t convinced this will fix it. Mark all appropriate statements about this assignment statement.

- Executing this assignment statement will cause an error.
- After this assignment, invoking `lacey.bark()` will cause an error.
- This assignment statement will have no effect at all.
- None of the above criticisms are valid.

(c) Mark all lines that should be removed so that the expression `N().r()` evaluates to 1.

```python
class M:
    p = 2  # optional
    q = True
    def r(self):
        if self.q:
            return self.p
        return self.r() - 1  # optional

class N(M):
    p = 1
    q = False
    def r(self):
        return self.p + 1
```