# INSTRUCTIONS

- You have 10 minutes to complete this quiz.
- The exam is closed book, closed notes, closed computer, closed calculator.
- The final score for this quiz will be assigned based on **effort** rather than correctness.
- Mark your answers **on the exam itself**. We will **not** grade answers written on scratch paper.
- For multiple choice questions,
  - ☐ means mark all options that apply
  - ☐ means mark a single choice

<table>
<thead>
<tr>
<th>Last name</th>
<th>First name</th>
<th>Student ID number</th>
<th>CalCentral email (@berkeley.edu)</th>
<th>Teaching Assistant</th>
<th>Name of the person to your left</th>
<th>Name of the person to your right</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Alex Stennet</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Angela Kwon</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Ashley Chien</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Joyce Luong</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Karthik Bharathala</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Kavi Gupta</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Kelly Chen</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Michael Gibbes</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Michelle Hwang</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Mitas Ray</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Rocky Duan</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Samantha Wong</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*All the work on this exam is my own. (please sign)*
1. (5 points)  **Oops! ... I Did It Again**

(a) (1 pt) 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
```

Angela, 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) (2 pt) 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!
```

Rocky 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
```

Rocky 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 `fido.bark()` will cause an error.
- This assignment statement will have no effect at all.
- None of the above criticisms are valid.

(c) (2 pt) Mark **all** lines that should be removed so that the expression `N().r()` evaluates to 1.
```python
class M:
    p = 2
    q = True
    def r(self):
        if self.q:
            return self.p
        return self.r() - 1

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