Lab 2: Introduction to Object-Oriented Programming Solution

2) Designing Classes

1. *Read the problem description.*

2. *Decide what classes you need to design.*

```
class Race:
    pass

class Runner:
    pass
```

3. *Sample usage.*

```
class Race:
    """
    === Sample Usage ===

    Create a race registry:
    >>> r = Race()
    >>> r.categories['lt20']
    []
    >>> r.categories['lt30']
    []
    >>> r.categories['lt40']
    []
    >>> r.categories['gt40']
    []

    Registering runners:
    >>> runner_1 = Runner('Gerhard','gerhard@gmail.com')
    >>> r.register(runner_1, 'lt40')
    >>> r.categories['lt40'][0].name
    Gerhard
    >>> runner_2 = Runner('Tom','tom@gmail.com')
    >>> r.register(runner_2, 'lt30')
    >>> r.categories['lt30'][0].name
    Tom
    >>> runner_3 = Runner('Toni','toni@gmail.com')
```

```
        >>> r.register(runner_3, 'lt20')
        >>> r.categories['lt20'][0].name
        Toni
        >>> r.register(runner_1, 'lt30')
        >>> r.categories['lt30'][1].name
        Gerhard
        """
        pass


    ...


    if __name__ == '__main__':
        import doctest
        doctest.testmod()
```

Correct Solution:

```
    class Race:
        """
        === Sample Usage ===

        Create a race registry:
        >>> r = Race()
        >>> r.runners
        []

        Registering runners:
        >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
        >>> r.register(runner_1)
        >>> r.runners[0].name
        'Gerhard'
        >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt30')
        >>> r.register(runner_2)
        >>> r.runners[1].name
        'Tom'
        >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
        >>> r.register(runner_3)
        >>> r.runners[2].name
        'Toni'

        Updating runner in a speed category:
        >>> runner_4 = r.get_runner('Gerhard')
        >>> runner_4.edit_category('lt30')
        >>> runner_4.speed_category
        'lt30'

        Get all runners in a speed category:
        >>> r.get_runners('lt30')
        ['Gerhard','Tom']
        """
```

```
34
35
36      class Runner:
37          """
38          === Sample Usage ===
39          Create a runner:
40          >>> runner = Runner('Gerhard', 'gerhard@gmail.com','lt30')
41          >>> runner.name
42          'Gerhard'
43          >>> runner.email
44          'gerhard@gmail.com'
45          >>> runner.speed_category
46          'lt30'
47          """
48
49      if __name__ == '__main__':
50          import doctest
51          doctest.testmod()
52
53
```

4. *Designing the interface.*

```
1       class Race:
2           """Race Registry
3
4           === Attributes ===
5           runners: a list of runners in race
6
7           === Sample Usage ===
8
9           Create a race registry:
10          >>> r = Race()
11          >>> r.runners
12          []
13
14          Registering runners:
15          >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
16          >>> r.register(runner_1)
17          >>> r.runners[0].name
18          'Gerhard'
19          >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt30')
20          >>> r.register(runner_2)
21          >>> r.runners[1].name
22          'Tom'
23          >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
24          >>> r.register(runner_3)
25          >>> r.runners[2].name
26          'Toni'
27
28          Updating runner in a speed category:
29          >>> runner_4 = r.get_runner('Gerhard')
30          >>> runner_4.edit_category('lt30')
31          >>> runner_4.speed_category
```

```
        'lt30'

        Get all runners in a speed category:
        >>> r.get_runners('lt30')
        ['Gerhard','Tom']
        """

        def __init__(self) -> None:
            pass

        def register(self, runner: Runner, category: str) -> None:
            pass

        def get_runners(self, category: str) -> None:
            pass

        def get_runner(self, name: str) -> None:
            pass

        pass


    class Runner:
        """A runner for the race

        === Attributes ===
        name: the name of runner.
        email: the email of runner.
        speed_category: speed category runner is racing in

        === Sample Usage ===
        Create a runner:
        >>> runner = Runner('Gerhard', 'gerhard@gmail.com','lt30')
        >>> runner.name
        'Gerhard'
        >>> runner.email
        'gerhard@gmail.com'
        >>> runner.speed_category
        'lt30'
        """

        def __init__(self, name: str, email: str, speed_category: str)
    -> None:
            pass

        def edit(self, email: str, speed_category: str) -> None:
            pass

        def withdraw(self) -> None:
            pass

    if __name__ == '__main__':
        import doctest
        doctest.testmod()
```