Lab 2: Introduction to Object-Oriented Programming Solution

2) Designing Classes

1. *Read the problem description.*

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

```python
class Race:
    pass

class Runner:
    pass
```

3. *Sample usage.*

```python
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')
```

```python
        >>> 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:

```python
    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
```

```python
        'lt30'

        Get all runners in a speed category:
        >>> runners = r.get_runners('lt30')
        >>> runners[0].name
        'Gerhard'
        >>> runners[1].name
        'Tom'
        """

    def __init__(self) -> None:
        """Initializes race registry

        >>> r = Race()
        >>> r.runners()
        []
        """
        pass

    def register(self, runner: Runner) -> None:
        """Registers runner to race

        >>> r = Race()
        >>> runner = Runner('Gerhard','gerhard@gmail.com','lt30')
        >>> r.register(runner)
        >>> r.runners[0].name
        'Gerhard'
        """
        pass

    def get_runners(self, category: str) -> None:
        """Returns list of runners in race category

        Precondition: <speed_category> in ['lt20','lt30','lt40','
    gt40','']

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

    def get_runner(self, name: str) -> None:
        """Returns runner in race registry
```

```python
           Precondition: <name> != ''

           >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
           >>> r.register(runner_1)
           >>> fetched_runner_1 = r.get_runners('Gerhard')
           >>> fetched_runner_1.name
           'Gerhard'
           >>> fetched_runner_2 = r.get_runners('Toni')
           >>> fetched_runner_2
           None
           """
           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:
           """Initialize runner

           Precondition: <name> != ''
           Precondition: <email> != ''
           Precondition: <speed_category> in ['lt20','lt30','lt40','
   gt40','']

           >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
           >>> runner.name
           'Gerhard'
           >>> runner.email
           'gerhard@gmail.com'
           >>> runner.speed_category
           'lt30'
           """
           pass

       def edit_email(self, email: str) -> None:
```

5

```
137              """Edits runner email information
138
139              Precondition: <email> != ''
140
141              >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
142              >>> runner.email
143              'gerhard@gmail.com'
144              >>> runner.edit_email('gerhard_2@gmail.com')
145              >>> runner.email
146              'gerhard_2@gmail.com'
147              """
148              pass
149
150        def edit_category(self, speed_category: str) -> None:
151              """Edits runner speed category information
152
153              Precondition: <speed_category> in ['lt20','lt30','lt40','
     gt40']
154
155              >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
156              >>> runner.speed_category
157              'lt30'
158              >>> runner.edit_category('lt20')
159              >>> runner.speed_category
160              'lt20'
161              """
162
163        def withdraw(self) -> None:
164              """Withdraws runner from race
165
166              >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
167              >>> runner.speed_category
168              'lt30'
169              >>> runner.withdraw()
170              >>> runner.speed_category
171              ''
172              """
173              pass
174
175    if __name__ == '__main__':
176          import doctest
177          doctest.testmod()
```

Correct Solution:

```
1          class Runner:
2          """A runner for the race
3
4          === Attributes ===
5          name: the name of runner.
6          email: the email of runner.
7          speed_category: speed category runner is racing in
```

```python
8
9          === Sample Usage ===
10         Create a runner:
11         >>> runner = Runner('Gerhard', 'gerhard@gmail.com','lt30')
12         >>> runner.name
13         'Gerhard'
14         >>> runner.email
15         'gerhard@gmail.com'
16         >>> runner.speed_category
17         'lt30'
18         """
19     name: str # <- Correct Solution
20     email: str # <- Correct Solution
21     speed_category: str # <- Correct Solution
22
23     def __init__(self, name: str, email: str, speed_category: str
   ) -> None:
24         """Initialize runner
25
26         Precondition: <name> != ''
27         Precondition: <email> != ''
28         Precondition: <speed_category> in ['lt20','lt30','lt40','
   gt40','']
29
30         >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
   ')
31         >>> runner.name
32         'Gerhard'
33         >>> runner.email
34         'gerhard@gmail.com'
35         >>> runner.speed_category
36         'lt30'
37         """
38         pass
39
40     def edit_email(self, email: str) -> None:
41         """Edits runner email information
42
43         Precondition: <email> != ''
44
45         >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
   ')
46         >>> runner.email
47         'gerhard@gmail.com'
48         >>> runner.edit_email('gerhard_2@gmail.com')
49         >>> runner.email
50         'gerhard_2@gmail.com'
51         """
52         pass
53
54     def edit_category(self, speed_category: str) -> None:
55         """Edits runner speed category information
56
```

```python
                    Precondition: <speed_category> in ['lt20','lt30','lt40','
gt40']

                    >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
')
                    >>> runner.speed_category
                    'lt30'
                    >>> runner.edit_category('lt20')
                    >>> runner.speed_category
                    'lt20'
                    """

        def withdraw(self) -> None:
                """Withdraws runner from race

                    >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
')
                    >>> runner.speed_category
                    'lt30'
                    >>> runner.withdraw()
                    >>> runner.speed_category
                    ''
                    """
                pass


    class Race:
        """Race Registry

        === Attributes ===
        runners: a list of runners in 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'
```

```python
        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:
        >>> runners = r.get_runners('lt30')
        >>> runners[0].name
        'Gerhard'
        >>> runners[1].name
        'Tom'
        """
        runners: List[Runner] # <- Correct Solution

        def __init__(self) -> None:
            """Initializes race registry

            >>> r = Race()
            >>> r.runners()
            []
            """
            pass

        def register(self, runner: Runner) -> None:
            """Registers runner to race

            >>> r = Race()
            >>> runner = Runner('Gerhard','gerhard@gmail.com','lt30')
            >>> r.register(runner)
            >>> r.runners[0].name
            'Gerhard'
            """
            pass

        def get_runners(self, category: str) -> None:
            """Returns list of runners in race category

            Precondition: <speed_category> in ['lt20','lt30','lt40','
    gt40'] # <- correct solution

            >>> r = Race() # <- Correct Solution
            >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', '
    lt40')
            >>> r.register(runner_1)
            >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt20')
            >>> r.register(runner_2)
            >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
            >>> r.register(runner_3)
            >>> runners = r.get_runners('lt20')
            >>> runners[0].name
            'Tom'
            >>> runners[1].name
```

```python
            'Toni'
            >>> r.get_runners('lt30')
            []
            """
            pass

    def get_runner(self, name: str) -> None:
        """Returns runner in race registry

        Precondition: <name> != ''

        >>> r = Race() # <- Correct Solution
        >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', '
    lt40')
        >>> r.register(runner_1)
        >>> fetched_runner_1 = r.get_runners('Gerhard')
        >>> fetched_runner_1.name
        'Gerhard'
        >>> fetched_runner_2 = r.get_runners('Toni')
        >>> fetched_runner_2
        """
        pass

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