

# Lab 2: Introduction to Object-Oriented Programming

## Solution

### 2) Designing Classes

1. *Read the problem description.*
2. *Decide what classes you need to design.*

```
1  class Race:
2      pass
3
4  class Runner:
5      pass
```

3. *Sample usage.*

```
1  class Race:
2      """
3      === Sample Usage ===
4
5      Create a race registry:
6      >>> r = Race()
7      >>> r.categories['lt20']
8      []
9      >>> r.categories['lt30']
10     []
11     >>> r.categories['lt40']
12     []
13     >>> r.categories['gt40']
14     []
15
16     Registering runners:
17     >>> runner_1 = Runner('Gerhard','gerhard@gmail.com')
18     >>> r.register(runner_1, 'lt40')
19     >>> r.categories['lt40'][0].name
20     Gerhard
21     >>> runner_2 = Runner('Tom','tom@gmail.com')
22     >>> r.register(runner_2, 'lt30')
23     >>> r.categories['lt30'][0].name
24     Tom
25     >>> runner_3 = Runner('Toni','toni@gmail.com')
```

```

26     >>> r.register(runner_3, 'lt20')
27     >>> r.categories['lt20'][0].name
28     Toni
29     >>> r.register(runner_1, 'lt30')
30     >>> r.categories['lt30'][1].name
31     Gerhard
32     """
33     pass
34
35
36     ...
37
38
39     if __name__ == '__main__':
40         import doctest
41         doctest.testmod()

```

### Correct Solution:

```

1     class Race:
2         """
3         === Sample Usage ===
4
5         Create a race registry:
6         >>> r = Race()
7         >>> r.runners
8         []
9
10        Registering runners:
11        >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
12        >>> r.register(runner_1)
13        >>> r.runners[0].name
14        'Gerhard'
15        >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt30')
16        >>> r.register(runner_2)
17        >>> r.runners[1].name
18        'Tom'
19        >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
20        >>> r.register(runner_3)
21        >>> r.runners[2].name
22        'Toni'
23
24        Updating runner in a speed category:
25        >>> runner_4 = r.get_runner('Gerhard')
26        >>> runner_4.edit_category('lt30')
27        >>> runner_4.speed_category
28        'lt30'
29
30        Get all runners in a speed category:
31        >>> r.get_runners('lt30')
32        ['Gerhard','Tom']
33        """

```

```

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

```

```

32         'lt30'
33
34     Get all runners in a speed category:
35     >>> runners = r.get_runners('lt30')
36     >>> runners[0].name
37     'Gerhard'
38     >>> runners[1].name
39     'Tom'
40     """
41
42     def __init__(self) -> None:
43         """Initializes race registry
44
45         >>> r = Race()
46         >>> r.runners()
47         []
48         """
49         pass
50
51     def register(self, runner: Runner) -> None:
52         """Registers runner to race
53
54         >>> r = Race()
55         >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
56         >>> r.register(runner)
57         >>> r.runners[0].name
58         'Gerhard'
59         """
60         pass
61
62     def get_runners(self, category: str) -> None:
63         """Returns list of runners in race category
64
65         Precondition: <speed_category> in ['lt20', 'lt30', 'lt40', '
gt40', '']
66
67         >>> runner_1 = Runner('Gerhard', 'gerhard@gmail.com', 'lt40')
68         >>> r.register(runner_1)
69         >>> runner_2 = Runner('Tom', 'tom@gmail.com', 'lt20')
70         >>> r.register(runner_2)
71         >>> runner_3 = Runner('Toni', 'toni@gmail.com', 'lt20')
72         >>> r.register(runner_3)
73         >>> runners = r.get_runners('lt20')
74         >>> runners[0].name
75         'Tom'
76         >>> runners[1].name
77         'Toni'
78         >>> r.get_runners('lt30')
79         []
80         """
81         pass
82
83     def get_runner(self, name: str) -> None:
84         """Returns runner in race registry

```

```

85
86         Precondition: <name> != ''
87
88         >>> runner_1 = Runner('Gerhard', 'gerhard@gmail.com', 'lt40')
89         >>> r.register(runner_1)
90         >>> fetched_runner_1 = r.get_runners('Gerhard')
91         >>> fetched_runner_1.name
92         'Gerhard'
93         >>> fetched_runner_2 = r.get_runners('Toni')
94         >>> fetched_runner_2
95         None
96         """
97         pass
98
99
100 class Runner:
101     """A runner for the race
102
103     === Attributes ===
104     name: the name of runner.
105     email: the email of runner.
106     speed_category: speed category runner is racing in
107
108     === Sample Usage ===
109     Create a runner:
110     >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
111     >>> runner.name
112     'Gerhard'
113     >>> runner.email
114     'gerhard@gmail.com'
115     >>> runner.speed_category
116     'lt30'
117     """
118
119     def __init__(self, name: str, email: str, speed_category: str)
-> None:
120         """Initialize runner
121
122         Precondition: <name> != ''
123         Precondition: <email> != ''
124         Precondition: <speed_category> in ['lt20', 'lt30', 'lt40', '
gt40', '']
125
126         >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30')
127         >>> runner.name
128         'Gerhard'
129         >>> runner.email
130         'gerhard@gmail.com'
131         >>> runner.speed_category
132         'lt30'
133         """
134         pass
135
136     def edit_email(self, email: str) -> None:

```

```

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

```

```

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
24    ) -> None:
25        """Initialize runner
26
27        Precondition: <name> != ''
28        Precondition: <email> != ''
29        Precondition: <speed_category> in ['lt20', 'lt30', 'lt40', '
30    gt40', '']
31
32    >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
33    ')
34
35    >>> runner.name
36    'Gerhard'
37    >>> runner.email
38    'gerhard@gmail.com'
39    >>> runner.speed_category
40    'lt30'
41    """
42    pass
43
44    def edit_email(self, email: str) -> None:
45        """Edits runner email information
46
47        Precondition: <email> != ''
48
49    >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
50    ')
51
52    >>> runner.email
53    'gerhard@gmail.com'
54    >>> runner.edit_email('gerhard_2@gmail.com')
55    >>> runner.email
56    'gerhard_2@gmail.com'
57    """
58    pass
59
60    def edit_category(self, speed_category: str) -> None:
61        """Edits runner speed category information

```

```

57         Precondition: <speed_category> in ['lt20','lt30','lt40','
gt40']
58
59     >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
')
60
61     >>> runner.speed_category
'lt30'
62     >>> runner.edit_category('lt20')
63     >>> runner.speed_category
'lt20'
64     >>>
65     """
66
67     def withdraw(self) -> None:
68         """Withdraws runner from race
69
70     >>> runner = Runner('Gerhard', 'gerhard@gmail.com', 'lt30
')
71
72     >>> runner.speed_category
'lt30'
73     >>> runner.withdraw()
74     >>> runner.speed_category
''
75     >>>
76     """
77     pass
78
79
80     class Race:
81         """Race Registry
82
83         === Attributes ===
84         runners: a list of runners in race
85
86         === Sample Usage ===
87
88         Create a race registry:
89         >>> r = Race()
90         >>> r.runners
91         []
92
93         Registering runners:
94         >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', 'lt40')
95         >>> r.register(runner_1)
96         >>> r.runners[0].name
'Gerhard'
97
98         >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt30')
99         >>> r.register(runner_2)
100         >>> r.runners[1].name
'Tom'
101
102         >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
103         >>> r.register(runner_3)
104         >>> r.runners[2].name
'Toni'
105
106

```



```

107     Updating runner in a speed category:
108     >>> runner_4 = r.get_runner('Gerhard')
109     >>> runner_4.edit_category('lt30')
110     >>> runner_4.speed_category
111     'lt30'
112
113     Get all runners in a speed category:
114     >>> runners = r.get_runners('lt30')
115     >>> runners[0].name
116     'Gerhard'
117     >>> runners[1].name
118     'Tom'
119     """
120     runners: List[Runner] # <- Correct Solution
121
122     def __init__(self) -> None:
123         """Initializes race registry
124
125         >>> r = Race()
126         >>> r.runners()
127         []
128         """
129         pass
130
131     def register(self, runner: Runner) -> None:
132         """Registers runner to race
133
134         >>> r = Race()
135         >>> runner = Runner('Gerhard','gerhard@gmail.com','lt30')
136         >>> r.register(runner)
137         >>> r.runners[0].name
138         'Gerhard'
139         """
140         pass
141
142     def get_runners(self, category: str) -> None:
143         """Returns list of runners in race category
144
145         Precondition: <speed_category> in ['lt20','lt30','lt40','
gt40'] # <- correct solution
146
147         >>> r = Race() # <- Correct Solution
148         >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', '
lt40')
149         >>> r.register(runner_1)
150         >>> runner_2 = Runner('Tom','tom@gmail.com', 'lt20')
151         >>> r.register(runner_2)
152         >>> runner_3 = Runner('Toni','toni@gmail.com', 'lt20')
153         >>> r.register(runner_3)
154         >>> runners = r.get_runners('lt20')
155         >>> runners[0].name
156         'Tom'
157         >>> runners[1].name

```

```

158         'Toni'
159     >>> r.get_runners('lt30')
160     []
161     """
162     pass
163
164     def get_runner(self, name: str) -> None:
165         """Returns runner in race registry
166
167         Precondition: <name> != ''
168
169         >>> r = Race() # <- Correct Solution
170         >>> runner_1 = Runner('Gerhard','gerhard@gmail.com', '
lt40')
171         >>> r.register(runner_1)
172         >>> fetched_runner_1 = r.get_runners('Gerhard')
173         >>> fetched_runner_1.name
174         'Gerhard'
175         >>> fetched_runner_2 = r.get_runners('Toni')
176         >>> fetched_runner_2
177         """
178     pass
179
180     if __name__ == '__main__':
181         import doctest
182         doctest.testmod()
183
184

```