Classes

Reading: Sec 8.1—8.5

Project 2

Due in 1 week

Goal: basic class design, plus simple example of inheritance.

Project Overview

constructor: Card(i) where i is card id (0..51)

attributes: suit, rank, points

how to make a deck of cards: list comprehension

how to get a random hand: random.sample

derived class: BlackjackCard

inherit rank, suit, etc, but new definition of points

function: points(h) where h is a list of Card objects (and since BlackjackCard is a type of Card h should allow either type of object)
Unit Testing

to run test from command line:

% python3 -m unittest test_Cards.py

discuss source, show example of what failure looks like [edit test]

Aside: A Very Useful Function

Here is a bad/awkward design for the suit method:

```python
if id < 13:
    return 0
elif id < 26:
    return 1
elif ...
```

Same thing for rank:  avoid `elif` (my least favorite Python statement...)
Notice how ranks and suits are related to id:

<table>
<thead>
<tr>
<th>suit</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>1</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
<td>37</td>
<td>38</td>
</tr>
<tr>
<td>3</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
<td>49</td>
<td>50</td>
<td>51</td>
</tr>
</tbody>
</table>

Can you define a mathematical relationship between id and rank/suit? i.e. is there a Python expression that computes suit given id? or rank given id?

This sort of mapping is common enough that Python has a builtin function that computes both values at once:

```python
>>> suit, rank = divmod(id, 13)
```

[demo]
We need to use Unicode to print suit symbols:

- Clubs = ♣ = ‘\u2663’
- Diamonds = ♦ = ‘\u2666’
- Hearts = ♥ = ‘\u2665’
- Spades = ♠ = ‘\u2660’

Question: where do you put the definition of these symbols?

(a) inside __repr__

```python
def __repr__(self):
    syms = { 0 : ‘\u2663’, … }
    return …
```

But this means the dictionary is created each time we print a card :-(

(b) global variable, outside the class

```python
syms = { 0: ‘\u2663’, …

class Card:
    …
    def __repr__(self):
        …
```

But we want to avoid global vars if possible (why?)
**Class Variables**

Define the dictionary inside the class, but not inside any method

The dictionary will be created once, when the class is defined

The name is part of the class namespace

In OOP terminology, such variables are class variables

- only one copy, shared by all instances
- not part of any single instance