Defining Classes

CS 177 - Recitation 11
Recap

Previous week
We know how to define functions

Define a function:
```
def function(value):
    temp = value * value
    return temp
```

Call a function:
```
def main():
    result = function(10)
    print(result)
main()
```
We also used some other classes and objects

How we used classes in Graphics:

```python
myCircle = Circle(Point(0,10), 200)
```

But we did not define any classes yet.

We will learn how to define classes today.

Circle is class

myCircle is object
Define & use a class

How to define a class:

```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(', self.x, ',', self.y, ')')
```

How to use the class:

```python
def main():
    p = Point(10, 50)
    p.printPoint()
main()
```
Define & use a class

How to define a class:

```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
    def printPoint(self):
        print('(', self.x, ',', self.y, ')')
```

How to use the class:

```python
def main():
    p = Point(10, 50)
    p.printPoint()
main()
```
Define & use a class

How to define a class:

class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(' , self.x, ', ' , self.y, ')

def main():
    p = Point(10, 50)
    p.printPoint()

main()
Define & use a class

How to define a class:

```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(', self.x, ',', self.y, ')')

def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

How to use the class:
Define & use a class

How to define a class:

```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(', self.x, ',', self.y, ')')

def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

How to use the class:

pointer to the object itself

constructor function

other input parameters

object, instance
Define & use a class

How to define a class:

```python
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def printPoint(self):
        print('(', self.x, ',', self.y, ')')

def main():
    p = Point(10, 50)
    p.printPoint()

main()
```

Prints:

( 10 , 50 )

How to use the class:
Matrix example
class Matrix:
    def __init__(self, m, n, character):
        self.m = m
        self.n = n
        self.character = character

    def printMatrix(self):
        for i in range(self.m):
            for j in range(self.n):
                print(self.character, end=' ')
        print()

def main():
    m = Matrix(4, 5, '***
    m.printMatrix()

main()
class Matrix:
    def __init__(self, m, n, character):
        self.m = m
        self.n = n
        self.character = character

    def printMatrix(self):
        for i in range(self.m):
            for j in range(self.n):
                print(self.character, end=' ')
            print()
Matrix example

class Matrix:
    def __init__(self, m, n, character):
        self.m = m
        self.n = n
        self.character = character

    def printMatrix(self):
        for i in range(self.m):
            for j in range(self.n):
                print(self.character, end=' ')
        print()

def main():
    m = Matrix(3, 7, '0')
    print(m.m)
    print(m.n)
    print(m.character)

main()
class Matrix:
    def __init__(self, m, n, character):
        self.m = m
        self.n = n
        self.character = character

    def printMatrix(self):
        for i in range(self.m):
            for j in range(self.n):
                print(self.character, end=' ')  
        print()

def main():
    m = Matrix(3, 7, '0')
    print(m.value)

main()
Matrix example

class Matrix:
    def __init__(self, m, n, character):
      self.m = m
      self.n = n
      self.character = character

    def printMatrix(self):
      for i in range(self.m):
        for j in range(self.n):
          print(self.character, end=' ')
      print()

def main():
    m = Matrix(3, 7, '0')
    m.print()
    main()
Multi-side Dice
Example
Multi-side Dice Example

```python
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value
```
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    print(dice.getValue())

main()
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    print(dice.getValue())

main()

Prints: 5
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    print(dice.getValue())
    dice.roll()
    print(dice.getValue())

main()
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    print(dice.getValue())
    dice.roll()
    print(dice.getValue())

main()

Prints:
5
4
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides+1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    dice.setValue(11)
    print(dice.getValue())

main()
from random import randrange

class MSDice:
    def __init__(self, sides):
        self.sides = sides
        self.value = 1

    def roll(self):
        self.value = randrange(1, self.sides + 1)

    def getValue(self):
        return self.value

    def setValue(self, value):
        self.value = value

def main():
    dice = MSDice(6)
    dice.roll()
    dice.setValue(11)
    print(dice.getValue())

main()

Prints:
11
Card example
class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = [ '2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])
class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = ['2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A']

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

card = Card(0)
card.printCard()
card = Card(51)
card.printCard()
class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13
    
    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = [ '2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

card = Card(0)
card.printCard()
card = Card(51)
card.printCard()
for i in range(7, 17):
    card = Card(i)
    card.printCard()
class Card:
    def __init__(self, value):
        self.cardType = int(value / 13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = [ '2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

for i in range(7, 17):
    card = Card(i)
    card.printCard()
Card example

class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = [ '2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

for i in range(5):
    temp = randrange(52)
    card = Card(temp)
    card.printCard()

Output:
class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤',  # spades
                     '♡',  # hearts
                     '♢',  # diamonds
                     '♧']  # clubs
        numberChars = [ '2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

for i in range(5):
    temp = randrange(52)
    card = Card(temp)
    card.printCard()

Output:

♦ 9
♠ 9
♣ 2
♦ J
♣ 6
Card example

class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤', # spades
                     '♡', # hearts
                     '♢', # diamonds
                     '♧'] # clubs

        numberChars = [ '2', '3', '4', '5', '6', '7', '8',
                        '9', '10', 'J', 'Q', 'K', 'A' ]

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

for i in range(5):
    temp = randrange(52)
    card = Card(temp)
    card.printCard()

3 runs:

♠️ 9
♥️ 9
♣️ 4
♦️ 7
♣️ 4
♥️ 7
from random import randrange

class Card:
    def __init__(self, value):
        self.cardType = int(value/13)
        self.number = value % 13

    def printCard(self):
        typeChars = ['♤',  # spades
                     '♡',  # hearts
                     '♢',  # diamonds
                     '♧']  # clubs

        numberChars = ['2', '3', '4', '5',
                        '6', '7', '8',
                        '9', '10', 'J',
                        'Q', 'K', 'A']

        print(typeChars[self.cardType], end=' ')
        print(numberChars[self.number])

    def getRandomCard():
        temp = randrange(52)
        return Card(temp)

card = Card.getRandomCard()
card.printCard()
Card example

What is the probability of having at least one card twice?

```python
for i in range(5):
    card = Card.getRandomCard()
    card.printCard()
```
What is the probability of having at least one card twice?

```python
for i in range(5):
    card = Card.getRandomCard()
    card.printCard()
```

= 1 - probability of having all cards different
= 1 - ( 52/52 * 51/52 * 50/52 * 49/52 * 48/52)
= 0.17971622142081867
Card example

What is the probability of having at least one card twice?

```python
for i in range(5):
    card = Card.getRandomCard()
    card.printCard()
```

1st trial: ♦️ 8, ♠️ 4, ♦️ 8, ♠️ 5, ♦️ 5
2nd trial: ♠️ J, ♣️ Q, ♣️ 9, ♣️ 6, ♠️ 5
3rd trial: ♦️ 4, ♦️ 4, ♠️ 6, ♠️ 8, ♠️ 3
4th trial: ♠️ 7, ♠️ 5, ♠️ 4, ♠️ 5, ♠️ 5
5th trial: Voila!
Card example

Using our Card class, how to pick 5 *unique* cards?

But first…

```python
card1 = Card(30)
card2 = Card(30)

if card1 == card2:
    print('yes')
else:
    print('no')
```

Output?
Card example

Using our Card class, how to pick 5 unique cards?

But first…

```python
import Card

card1 = Card(30)
card2 = Card(30)

if card1 == card2:
    print('yes')
else:
    print('no')
```

Output?

no
Card example

Using our Card class, how to pick 5 unique cards?

Let’s create a Deck!
Card example

Using our Card class, how to pick 5 unique cards?

def撒εック = []

for i in range(52):
    card = Card(i)
    deck.append(card)

for i in range(5):
    card = random.choice(deck)
    card.printCard()
    deck.remove(card)
Card example

Using our Card class, how to pick 5 unique cards?

desk = []

for i in range(52):
    card = Card(i)
    desk.append(card)

for i in range(5):
    card = random.choice(deck)
    card.printCard()
    deck.remove(card)

Output:

8
7
7
Q
2
Card example

Using our Card class, how to pick 5 unique cards?

def card(i):
    pass

deck = []

print(len(deck))
for i in range(52):
    card = Card(i)
    deck.append(card)

print(len(deck))
for i in range(5):
    card = random.choice(deck)
    card.printCard()
    deck.remove(card)

print(len(deck))
Card example

Using our Card class, how to pick 5 unique cards?

dec = []
print(len(deck))
for i in range(52):
    card = Card(i)
    deck.append(card)

print(len(deck))
for i in range(5):
    card = random.choice(deck)
    card.printCard()
    deck.remove(card)

print(len(deck))

Output?

0 52
♥ 5
♥ 7
♣ 7
♦ Q
♥ 9
47
Using our Card class, how to pick 5 unique cards?

deck = []

print(len(deck))

for i in range(52):
    card = Card(i)
    deck.append(card)

print(len(deck))

for i in range(5):
    card = random.choice(deck)
    card.printCard()
    deck.remove(card)

print(len(deck))
Questions

Thanks