

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:

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

Circle is class
myCircle is object

But we did not define any classes yet.

We will learn how to define classes today.

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, ')')
```

How to use
the class:

```
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, ')')
```

constructor
function

How to use
the class:

```
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, ')')
```

How to use
the class:

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

main()
```

constructor
function

pointer to the
object itself

Define & use a class

other input parameters

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, ')')
```

How to use the class:

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

main()
```

constructor function

pointer to the object itself

Define & use a class

other input parameters

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 + ')')
```

How to use the class:

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

main()
```

constructor function

pointer to the object itself

object, instance

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, ')')
```

How to use
the class:

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

main()
```

Prints:

```
( 10 , 50 )
```

Matrix example

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()
```

new line

print without
new line

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

main()
```

Prints:

```
* * * * *
* * * * *
* * * * *
* * * * *
```

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.printMatrix()

main()
```

Prints:

```
0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

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()
```

Prints:

3
7
0

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.value)

main()
```

Error:

```
Traceback (most recent call last):
  File "a.py", line 16, in <module>
    print(m.value)
AttributeError: 'Matrix' object has
no attribute 'value'
```

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()
```

Error:

```
Traceback (most recent call last):
  File "a.py", line 15, in <module>
    m.print()
AttributeError: 'Matrix' object has
no attribute 'print'
```


Multi-side Dice Example

Multi-side Dice Example

```
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
```

Multi-side Dice Example

```
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:



Multi-side Dice Example

```
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

Multi-side Dice Example

```
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:



Multi-side Dice Example

```
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

Multi-side Dice Example

```
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:

?

Multi-side Dice Example

```
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

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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 example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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()
```

Output?

?

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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()
```

Output:

```
♠ 2
♣ A
```

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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()
```

Output?

?

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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()
```

Output:

```
♠ 9
♠ 10
♠ J
♠ Q
♠ K
♠ A
♥ 2
♥ 3
♥ 4
♥ 5
```

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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:

?

Card example

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

    def printCard(self):
        typeChars = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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 = ['\u2664', # spades
                    '\u2661', # hearts
                    '\u2662', # diamonds
                    '\u2667'] # 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:

♠	4
♠	10
♣	7
♣	4
♥	7

♦	3
♣	7
♥	8
♦	2
♣	9

♦	9
♠	9
♦	2
♠	J
♦	6

```
from random import randrange
```

```
class Card:
```

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

```
    def printCard(self):  
        typeChars = ['\u2664', # spades  
                    '\u2661', # hearts  
                    '\u2662', # diamonds  
                    '\u2667'] # 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()
```

self

self

new function

no self

Card example

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

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

Card example

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

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

$$\begin{aligned} &= 1 - \text{probability of having all cards different} \\ &= 1 - \left(\frac{52}{52} * \frac{51}{52} * \frac{50}{52} * \frac{49}{52} * \frac{48}{52} \right) \\ &= 0.17971622142081867 \end{aligned}$$

Card example

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

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

1st trial:

♥	10
♥	7
♦	4
♥	J
♦	8

2nd trial:

♦	8
♦	J
♠	9
♣	9
♦	4

3rd trial:

♠	4
♣	Q
♠	Q
♦	6
♣	2

4th trial:

♦	5
♣	6
♥	K
♥	8
♠	10

5th trial:

♠	5
♠	5
♣	4
♥	7
♣	3

Voila!

Card example

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

But first...

```
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....

```
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?

```
deck = []

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)
```

Output?

?

Card example






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

```
deck = []

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)
```

Output:

	8
	7
	7
	Q
	2

Card example

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))
```



Output?

Card example

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))
```

Output?

```
0  
52  
♥ 5  
♥ 7  
♣ 7  
♦ Q  
♥ 9  
47
```

Card example

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))
```

0
52
♥ 5
♥ 7
♣ 7
♦ Q
♥ 9
47

Questions

Thanks