CS21: INTRODUCTION TO COMPUTER SCIENCE

Prof. Mathieson
Fall 2017
Swarthmore College
3square puzzle examples

- corn often found on this
- type of poem
- a gamble

From: Matt

- You play baseball with this
- Also Known As
- The Indian Wonder of the World

From: Adi
3square puzzle examples

From: Matt

- corn often found on this
- type of poem
- a gamble

From: Adi

- You play baseball with this
- Also Known As
- The Indian Wonder of the World
3square puzzle examples

From: Matt

- corn often found on this
- type of poem
- a gamble

From: Adi

- You play baseball with this
- Also Known As
- The Indian Wonder of the World
Informal quiz (discuss with a partner)

1) What is this class for? How many instance variables are there? How many methods?

2) Complete the `getValue(..)` method.

3) Complete the `roll(..)` method.

4) What is wrong with the `__str__(..)` method?

5) Does a constructor return something? Why or why not?

class Die:

    def __init__(self, num_sides):
        self.sides = num_sides
        self.value = 1 # default starting value

    def roll(self):

    def getValue(self):

    def __str__(self):
        print("%d-sided die, current value: %d" % (self.sides, self.value))
Informal quiz (discuss with a partner)

1) What is this class for? How many *instance variables* are there? How many *methods*? 2, 3 (besides constructor)

2) Complete the `getValue(..)` method.

3) Complete the `roll(..)` method.

4) What is wrong with the `__str__(..)` method?

5) Does a *constructor* return something? Why or why not?

```python
class Die:
    def __init__(self, num_sides):
        self.sides = num_sides
        self.value = 1 # default starting value

    def roll(self):

    def getValue(self):

    def __str__(self):
        print("%d-sided die, current value: %d" % (self.sides, self.value))
```
Informal quiz (discuss with a partner)

1) What is this class for? How many instance variables are there? How many methods? 2, 3 (besides constructor)

2) Complete the getValue(..) method.

3) Complete the roll(..) method.

4) What is wrong with the __str__(..) method?

5) Does a constructor return something? Why or why not?

class Die:

    def __init__(self, num_sides):
        self.sides = num_sides
        self.value = 1 # default starting value

    def roll(self):

    def getValue(self):
        return self.value

    def __str__(self):
        print("%d-sided die, current value: %d" % (self.sides, self.value))
Informal quiz (discuss with a partner)

1) What is this class for? How many *instance variables* are there? How many *methods*?

2) Complete the `getValue(..)` method.

3) Complete the `roll(..)` method.

4) What is wrong with the `__str__(..)` method?

5) Does a *constructor* return something? Why or why not?

```python
class Die:
    def __init__(self, num_sides):
        self.sides = num_sides
        self.value = 1 # default starting value

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

    def getValue(self):
        return self.value

    def __str__(self):
        print("%d-sided die, current value: %d" % (self.sides, self.value))
```
Informal quiz (discuss with a partner)

1) What is this class for? How many instance variables are there? How many methods? 2, 3 (besides constructor)

2) Complete the `getValue(..)` method.

3) Complete the `roll(..)` method.

4) What is wrong with the `__str__(..)` method? `return string, not print`

5) Does a constructor return something? Why or why not?
Informal quiz (discuss with a partner)

1) What is this class for? How many *instance variables* are there? How many *methods*? 2, 3 (besides constructor)

2) Complete the `getValue(..)` method.

3) Complete the `roll(..)` method.

4) What is wrong with the `__str__(..)` method? `return string, not print`

5) Does a *constructor* return something? Why or why not?

```python
class Die:
    def __init__(self, num_sides):
        self.sides = num_sides
        self.value = 1  # default starting value

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

    def getValue(self):
        return self.value

    def __str__(self):
        s = "%d-sided die, current value: %d" % (self.sides, self.value)
        return s
```

The constructor does create an object which we can assign to a variable name, but we do not use “return ____”. 
Outline Nov 22:

- Go over Quiz 4
- Continue: writing classes
- Finish: Student class example
- Start: RandomGene class

Notes

- Lab 9 due Monday after Thanksgiving
- There is lab this week! (Tues/Wed)
- Next ninja session: Sunday after Thanksgiving
Classes
Today

- Class writer vs. class user (example: graphics.py)
- Finish student example
- Start biology example
RandomGene class (TDD)

**Chromosome**

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
```

GTA C[GATCG]GCCCTA

```
| gene: 4-10 | constructor |
| start | end |
```

Query pos: 117, inside? yes!

Query pos: 13, inside? no!

**Instance variables**

* start (int)
* end (int)
* sequence (str)

**Methods**

* checkInside(pos)
  * return boolean
* base(pos)
  * return str