

# CS21: INTRODUCTION TO COMPUTER SCIENCE

---

Prof. Mathieson

Fall 2018

Swarthmore College

# 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?

```
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, 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):
        self.value = random.randrange(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?

```
class Die:

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

    def roll(self):
        self.value = random.randrange(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
```

# 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?

```
class Die:

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

    def roll(self):
        self.value = random.randrange(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 21:

- Continue: writing classes
- Student class example
- Snowflake class example
- Hand back Lab 7

## Notes

- Lab 9 due Monday after Thanksgiving
- There IS lab this week! (Tues/Wed)
- **Next ninja session**: Sunday after Thanksgiving 7-10pm
- Extra office hours: **Monday 2:30-4:30pm** (room 249)



# Classes

# Today

- Student example
- Snowflake example