Classes & Objects

COSC 101: Intro to Computing I
November 29, 2017
Define the following terms:

• Procedural Programming
• Object-Oriented Programming
• Class
• Object
• State (attribute)
• Method
• Constructor (initializer)
• Instantiation (initialization)
Why would we define our own class?
Syntax for defining a class?

```python
class ClassName:
    body
    body
    body
    body
    body
    body
    body
```
Syntax for defining a method?

```python
def method_name(self, [param1, param2, ...]):
    body
    body
    body
    body
    body
    body
    body
```
What is **self**?

### When to use **self**?

```python
class ClassName:
    def method_name(self, param):
        self.attribute = value
        self.other_method()
```

### When **not** to use **self**?

```python
object_name = class_name()

def function(object_name, param):
    object_name.method_name()
```
What does printing an object do?

<ClassName object at 0x1022966d8>

Can you change this?

class ClassName:
    def __str__(param):
        return "ClassName: " + self.attribute
class Course:
    def __init__(self, init_dept, init_num, init_prof, init_max):
        self.dept = init_dept
        self.num = init_num
        self.prof = init_prof
        self.max_enroll = init_max
        self.enrolled = 0
    def set_dept(self, dept):
        self.dept = dept
    def set_num(self, num):
        self.num = num
    def set_prof(self, prof):
        self.prof = prof
    def set_max_enroll(self, max_enroll):
        self.max_enroll = max_enroll
    def set_enrolled(self, enrollment):
        self.enrolled = enrollment
    def __str__(self):
        return self.dept + " " + str(self.num) + " - Prof. " + self.prof + " (" + str(self.enrolled) + " of " + str(self.max_enroll) + " students enrolled)"
thiscourse = Course("COSC", 101, "Madeline E. Smith", 24)
thiscourse.set_enrolled(26)
print(thiscourse)
What is the **output** of this program?

```python
thiscourse = Course("COSC", 101,
                    "Madeline E. Smith", 24)
thiscourse.set_enrolled(26)
print(thiscourse)
```

COSC 101 – Prof. Madeline E. Smith (26 of 24 students enrolled)
Rewrite constructor to set defaults:

class Course:
    def __init__(self, init_dept, init_num, init_prof, init_max):
        self.dept = init_dept
        self.num = init_num
        self.prof = init_prof
        self.max_enroll = init_max
        self.enrolled = 0
Rewrite constructor to set defaults:

class Course:
    def __init__(self, init_dept='', init_num=0, init_prof='', init_max=0):
        self.dept = init_dept
        self.num = init_num
        self.prof = init_prof
        self.max_enroll = init_max
        self.enrolled = 0
Write method to add students

class Course:

    def add_students(self, num_students):
        self.enrolled += num_students

    def add_student(self):
        self.enrolled += 1
Write methods to get data

class Course:
    def get_dept(self):
        return self.dept
    def get_num(self):
        return self.num
    def get_prof(self):
        return self.prof
    def get_max_enroll(self):
        return self.max_enroll
    def get_enrollment(self):
        return self.enrollment
Write function to update enrollment

def update_enrollment(course):
    title = course.get_dept() + " " +
            str(course.get_num())
enrolled = course.get_enrolled()
max_enroll = course.get_max_enroll()
print(max_enroll - enrolled, "seats " +
       "available in", title)
add = int(input("How many new students " +
                "to enroll now? "))
course.add_students(add)
enrolled = course.get_enrolled()
print(enrolled, "students "+
      "are now enrolled.")
Write function to create a course

def create_course():
    dept = input("What department is the " +\n             "course in? ")
    num = int(input("What is the course " +\n                   "number? "))
    prof = input("What is the professor’s" +\n                 " name? ")
    limit = int(input("How many students " +\n                     "may enroll? "))
    course = Course(dept, num, prof, limit)
    return course