LECTURE 24

OBJECT-ORIENTED PROGRAMMING 2

OPERATOR OVERLOADING

MCS 260 Fall 2020

Emily Dumas

REMINDERS

- Start on Project 3 immediately. Do not delay!
- Worksheet 9 available

ASSIGNMENT WITH OPERATION

It is common to write assignments that apply one operation to an existing variable and assign the result to that variable.

Example:

```
x = x + 25
```

There is a shorter syntax for this:

```
x += 25
```

Not repeating the name `x` twice can help avoid errors.

Old way	New way
a = a + b	a += b
a = a - b	a -= b
a = a * b	a *= b
a = a / b	a /= b
a = a ** b	a **= b
a = a % b	a %= b
a = a // b	a //= b

REVIEW

Key concepts from Lecture 23

- **class** -- A type in Python that combines data (attributes) and behavior (methods).
- **instance** or **object** -- A value whose type is a certain class (e.g. `"hello"` is an instance of `str`)
- **attribute** -- A variable local to an object, accessed as `objname.attrname`.
- **constructor** -- The method named `__init__` that is called when a new object is created.

GOALS FOR TODAY

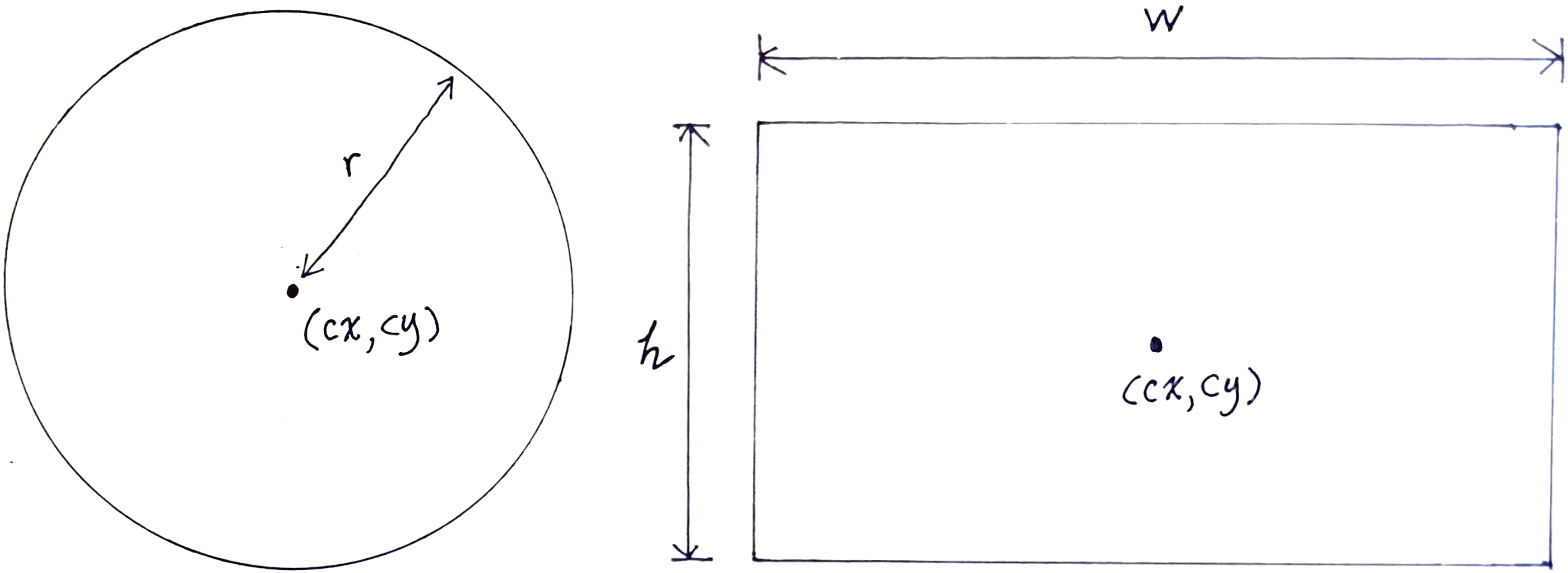
Improve our Rectangle and Circle classes.

Put them in a module.

Introduce operator overloading.

Result: geom.py
(Also see the updated geom.py developed in subsequent lectures.)

CIRCLES AND RECTANGLES



DESIRED METHODS

For both object types:

- Uniform scale about center
- Translation by a vector
- __str__

__REPR__

The __repr__ method converts an object to a string that should represent it perfectly; i.e. the object can be completely recovered from the data in the string.

Contrast to __str__, which emphasizes human readability (maybe at cost of ambiguity).

EQUALITY

How is `A==B` evaluated when `A` and `B` are objects?

By default, it checks whether the names refer to the same object in memory. This is often not what you want.

OVERLOADING

Python allows us to specify our own behavior for operators like `==`. This is called **operator overloading**.

If method `A.__eq__` exists, then `A==B` evaluates to the return value of `A.__eq__(B)`.

ISINSTANCE

The built-in function `isinstance(obj,cl)` returns a bool indicating whether `obj` is an instance of the class `cl`.

Using it sparingly. Often it is better to attempt to use expected behavior or attributes, and let an exception detect any problem.

Many operators can be overloaded, including:

Expression	**Special method**
`A+B`	`A.__add__(B)`
`A-B`	`A.__sub__(B)`
`A*B`	`A.__mul__(B)`
`A/B`	`A.__truediv__(B)`
`A**B`	`A.__pow__(B)`

List of many more in the Python documentation.

OVERLOADING BUILT-IN FUNCTIONS ETC.

Expression	Actually calls
`len(A)`	`A.__len__()`
`bool(A)`	`A.__bool__()`
`A[k]`	`A.__getitem__(k)`
`A[k]=v`	`A.__setitem__(k,v)`

REFERENCES

- In *Downey*:
  - Chapter 17 discusses classes, objects, and methods
- Object-oriented programming is discussed in general terms in Section 6.5 of Brookshear & Brylow.

REVISION HISTORY

- 2020-10-21 Change geom.py link to always go to Lecture 24 version
- 2020-10-17 Initial publication