As you arrive:

1. Start up your computer and plug it in.
2. ***Log into Angel*** and go to CSSE 120.
   Do the ***Attendance Widget*** –
   the PIN is on the board.
3. Go to the ***Course Schedule*** web page.
   Open the ***Slides*** for today if you wish.
4. Checkout today's project: `Session18_DefiningClasses`

Session 18

Defining your OWN classes

Defining your OWN classes

- Instance variables
- Constructors
- Methods

Project Time

Checkout today's project:
`Session18_DefiningClasses`

Are you in the Pydev perspective? If not:

`Window ~ Open Perspective ~ Other` then `Pydev`

Messed up views? If so:

`Window ~ Reset Perspective`

Troubles getting today's project? If so:

No SVN repositories view (tab)? If it is not there:

`Window ~ Show View ~ Other`
then `SVN ~ SVN Repositories`

1. ***In your SVN repositories view (tab), expand your repository*** *(the top-level item) if not already expanded.*
   - If no repository, perhaps you are in the wrong Workspace. Get help.
2. ***Right-click on today's project,*** *then select* ***`Checkout`****.*
   Press ***OK*** *as needed.* The project shows up in the
   `Pydev Package Explorer`
   to the left. Expand and browse the modules under `src` as desired.

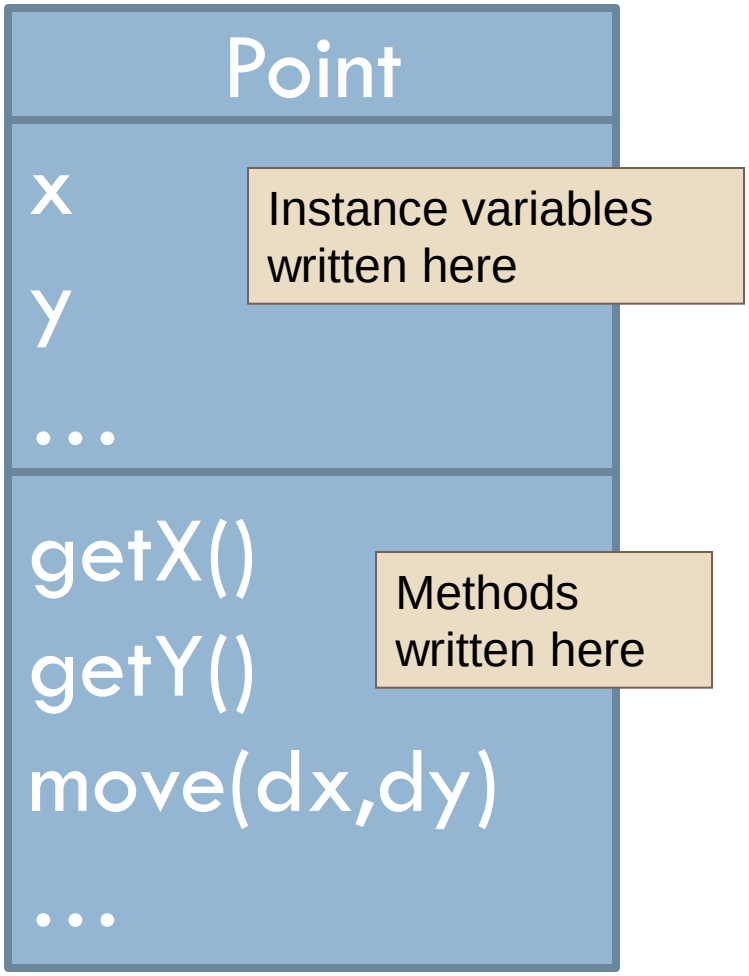
Review: What is an Object?

- An Object is an active data-type:
  - knows things about itself
    - fields
      - a.k.a. instance variables (or fields)
  - can be asked to (based on what it knows)
    - do things
      - mutator methods
    - provide info about itself and/or other objects that it knows about
      - accessor methods

Review: Object Terminology

- Objects are *data types* that might be considered ***active***
  - They **store information** in *instance variables*
  - They **manipulate their data** through *methods*
- Objects are *instances* of some *class*
- Objects are created by calling *constructors*

- **UML class diagram:**



Class analogies – Key Concept!

- A class is an "object factory"
  - Calling the constructor tells the classes to make a new object
  - Parameters to constructor are like "factory options", used to set instance variables
- Or think of class like a "rubber stamp"
  - Calling the constructor stamps out a new object shaped like the class
  - Parameters to constructor "fill in the blanks". That is, they are used to initialize instance variables.

Review: Using Objects in Python

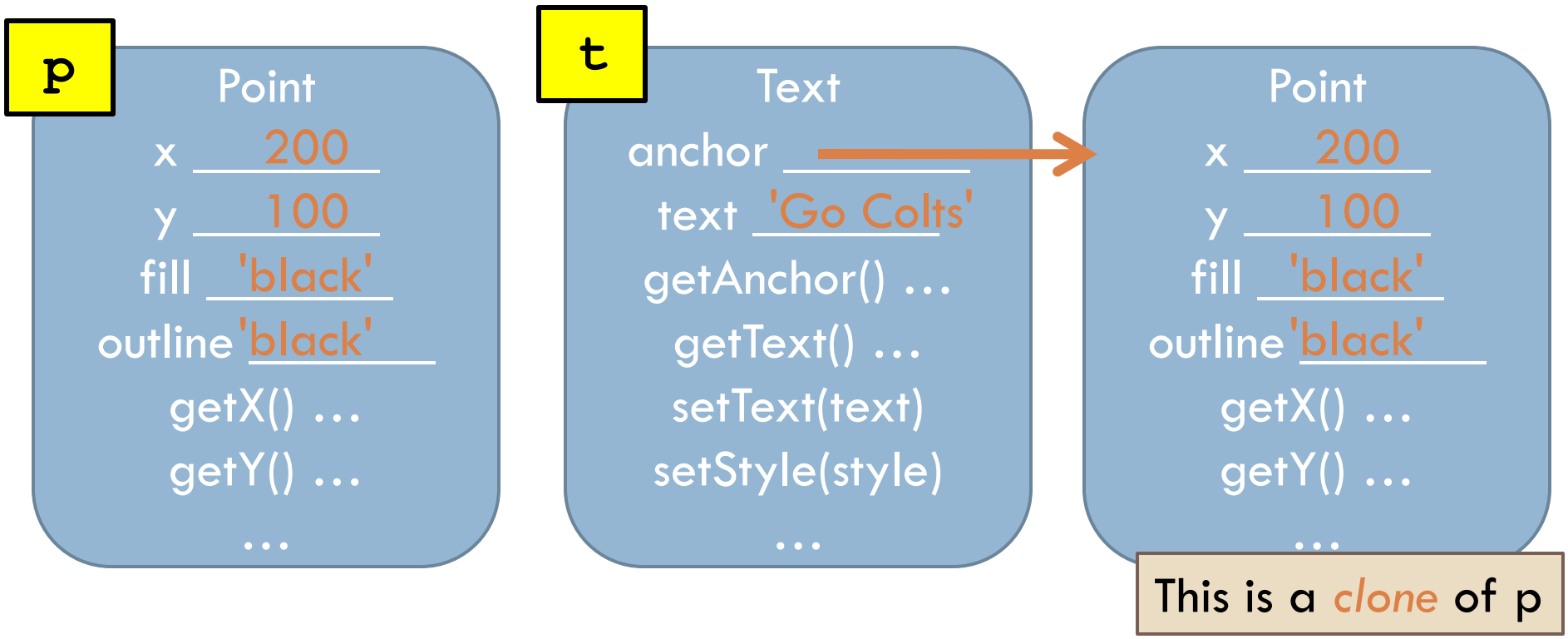
Look at *object_example.py* in today's project

In that example, make sure you understand:

- What is *win*?
- What is *GraphWin*?
- What is *.setStyle*?

Example

- `p = Point(200, 100)`
- `t = Text(p, 'Go Colts!')`

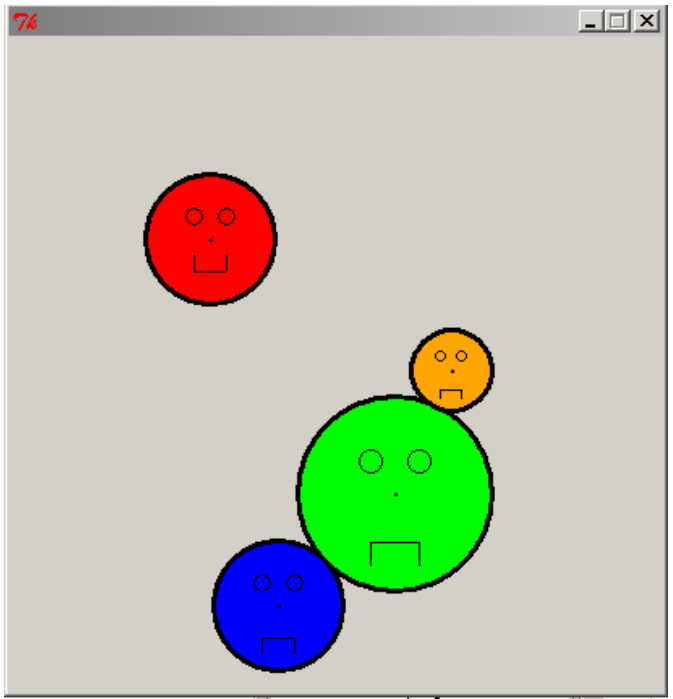


Creating Custom Objects: Defining Your Own Classes

- Custom objects:
  - Hide complexity
  - Provide another way to break problems into pieces
  - Make it easier to pass information around
- Example:

  Moving "Smiley" class.

- Let's create our own custom class and use it to instantiate objects.
- Use modules in project you checked out earlier



Coding MovingSmileys

- Create constructor noting default parameters
  - Defaults are size, color, and isSmiling
  - Study the code for creating parts
  - Explore how parts list is created
- Create draw() method and run scene1
- Add move() method, and run scene1
- Add smile and frown methods, which need to know about size
- Run scene 2, point out that 3 other methods needed for collisions to work

Review of Key Ideas

- *Constructor*:
  - Defined with special name `__init__`
  - Called like `ClassName()`
- *Instance variables*:
  - Created when we assign to them
  - Live as long as the object lives
- `self` formal parameter:
  - Implicitly get the value *before the dot* in the call
  - Allows an object to "talk about itself" in a method

Rest of Session

- Meet with your project team
  - Consider having a standup meeting
  - Continue working on this Sprint
  - ***Decide on time/venue for next meeting***
- **Sources of help after class:**
  - **Assistants in the CSSE lab**

```
CSSE lab: Moench F-217
7 to 9 p.m.
Sundays thru Thursdays
```

  - **Email** `csse120-staff@rose-hulman.edu`
    - You get faster response from the above than from just your instructor