

INTRODUCTION TO OBJECTS [PYTHON]

Unit: Advanced Topics

Evidence	Desired Results
<p>Students can...</p> <ul style="list-style-type: none">• Explain the difference between a class and an instance.• Create classes and declare an instance of that class for their Social Network project.• Create an init method within a class and declare class attributes• Explain why object oriented programming allows for more flexibility in implementation.	<ul style="list-style-type: none">• Social Network Project: User and Network Classes

Materials

Student Prompt / Starter Code / **Sample Code Slides**

Posters and Markers

Reference: **Object-Oriented Design Procedure**

Activity Plan

Teacher Prep: 30 minutes

Write an agenda on the board.
Pass out any materials for the day.
Think of extensions for projects to challenge students who finish early.
Consider questions to ask students as they work independently.
Identify potential misconceptions and errors students might make when writing today's programs.

Introduction/Hook: 30 minutes

Teachers	Students	Time
<p>Run a Think-Pair-Share around the journal prompts.</p> <p>Take notes on the discussion on the whiteboard. Be sure to focus on what kinds of information these applications collect and what functionality they have. Ask students to consider what the "bare minimum" of a social networking site might be.</p>	<p>Respond to the following questions in your journal:</p> <ul style="list-style-type: none">• What social networking applications do you use on a daily basis?• What information do they store?• What can they do?• What is the bare minimum a social network application would do?	15 min
<p>Instructions: Set a timer to 5 minutes. Have students discuss the questions in groups of four.</p> <p>Questions: During the all-class discussion, guide students to consider the following concepts:</p> <ul style="list-style-type: none">• What is the difference between frontend and backend?• How will we test a program that is only back-end?	<p>Discuss the following questions with your groups.</p> <p>Today we are going to build a backend for a social network program.</p> <ul style="list-style-type: none">• What does back-end mean to you?• What would you expect such a program to be able to do?• What information would you expect this program to store?	10 min
<p>Walk through today's agenda with the students.</p>	<p>Listen to the teacher as they walk through the agenda.</p>	5 min

Start the Project: 70 min

Exploration - Introduction to Objects (40 minutes)

Teacher	Students	Time
<p>Explain to students that before they build their Social Network, they are going to taken on a short challenge.</p> <p>Break students into groups of 4, and assign them one of the vehicles from the</p>	<p>Challenge: In your small group, list all of the qualities and functionalities of your item.</p>	10 min
<p>Tell students to do a silent gallery walk to see the work of other teams.</p>	<p>Gallery Walk: Read through the work of other groups. As you walk, consider:</p> <p>What do all of the items of other groups have in common? What distinguishes them?</p>	5 min

Lead a discussion with the classroom.
As you discuss, take notes on the board and draw something like this: https://www.ibm.com/developerworks/lotus/library/l5-object_oriented_LotusScript/figure2.gif

Be sure to hit on the following ideas:

Question 1:

Vehicles - This is like a CLASS in an OOP language (like Python). A class is basically defining a data type or structure that you can interact with in ways that you define.

- **All vehicles have some commonalities.** What are things that all groups had in common?
 - Every one of these vehicles is an **instance** of the class - it is like a blueprint of what you need to create a piece of data that has the same fields
 - Every one of these vehicles has some of the same functionality - these are **methods**.

Question 2:

Cars, Trucks, Planes - In OOP, this idea is inheritance. You can take the blueprint for a datatype and use it as a starting point for more specific objects.

- For example, if I started a car class, I would have the same things as the vehicle class. However, I might add some special methods or fields.
 - What fields might I add?
 - What methods might I add?

Summarize by saying:

- Without objects, data structures and functionality are built separately.
- Objects allow us to unify functionality and data together in a single thing.

Share Out your answers to the questions below:

- If we had to group all of the items together, what would we call this large group?
- What are the different sub-categories that we might have within this class?

10 min

<ul style="list-style-type: none">• Objects allow us to unify functionality and data together in a single thing.• Defining objects in python is like defining a new variable type.<ul style="list-style-type: none">- Show <code>string.format()</code> and talk about how the format method does not make sense on integers, but does on strings.		
<p>Circulate as groups work. If they get stuck, consider giving them ideas like an “animal” or “food” class. If students finish quickly, challenge them to think about what classes they might need to build a social network.</p>	<p>With a partner, use the OOP Vocabulary Slide to try to come up with your own analogy for Objects.</p> <p>Create a poster that shows your analogy.</p>	15 min

Object Oriented Design Procedure (30 min)

Teachers	Students	Time
<p>Explain that now that students have an understanding of the idea of objects, they will start to transfer it to their social network project.</p> <p>Demo the sample program to the students.</p> <p>Show them how to add users, add connections, print user information and print the entire network.</p> <p>Ask students to give inputs for different fields with the hope of “breaking” the program (i.e. add two users with the same name).</p> <p>Do not show them the inside of the code.</p>	<p>Interact with the program as the teacher demonstrates what it does. Volunteer names for new users.</p> <p>Consider ways to “break” the program.</p>	5 min
<p>Purpose: Create a mental model of the program before learning new material.</p> <p>Instructions: Introduce students to the Object Oriented Design Procedure. Explain each step and ask students to rephrase the steps in their own words.</p> <p>Set a timer to 15 minutes. Have groups work through steps 1-3 of the Design Procedure for the Social Network program. Encourage students to make their programs as different from the sample program as they wish.</p> <p>Questions:</p> <ul style="list-style-type: none">• Who is the “user” of your program?• What information are you storing on users?• What actions must users be able to do?	<p>With groups, work through steps 1-3 of the Object Oriented Design procedure. Remember, your program does not have to be identical to the example. Create a poster of your work.</p>	15 min

<p>Have each group present their responses to the design procedure. Encourage students to identify different interpretations of the procedure.</p> <p>Help the class come to a consensus on what steps 1-3 mean and what their programs are supposed to accomplish.</p>	<p>Present and discuss your responses to the Design Procedure and compare to how other groups responded.</p> <p>Identify any misconceptions you or others in your class, may have around the procedure.</p>	10 min
---	---	--------

Introduce New Information (45 mins)

Python Syntax - Objects (45 minutes)

<p>Have students open a new python script. Have them follow along with you as you define a new object. Make it simple, focus only on attributes for now. Use the examples from the slides.</p> <p>As you explain, focus on the following concepts:</p> <ul style="list-style-type: none">• Constructor• What does self mean?• Why is self an argument of the constructor? <p>Be sure to take questions from the students and check for understanding frequently.</p>	<p>Follow along in Python to define different objects.</p>	<p>10 min</p>
<p>Have students respond to the journal prompts, then Turn and Talk about their answers.</p>	<p>Respond to the following prompt:</p> <ul style="list-style-type: none">• What attributes does an instagram or facebook user class have?• What methods might a user class have?	<p>10 min</p>
<p>Add methods to your example classes. Have students follow along as you do this. Focus on the following concepts:</p> <ul style="list-style-type: none">• Why is self always an argument of a class function?• Why don't I have to pass the function self when I call it?	<p>Follow along as the teacher adds new methods to the example classes.</p>	<p>10 min</p>

Purpose: Practice thinking of class-specific methods.

Instructions: Set a timer to 10 minutes. Tell students that they are to write and test two new methods for each of the example classes.

Questions:

- What is the difference between a method and a function?
- Can I use your class method on an integer?

Instructions: When the timer goes off, have students volunteer to show their methods to the class.

Think of new methods for the two example classes. Write and test your new methods!

Volunteer to demonstrate your methods to the class.

15 min

Finish Project (90 mins)

The Network Class (60 minutes)

Teacher	Students	Time
<p>Purpose: Students practice implementing concepts of object oriented programming on their own with your support.</p> <p>Instructions: Set a timer to 50 minutes. Challenge students to think through the class methods they must create on their own. Remind them to test frequently.</p> <p>Also, remind students that they will have the opportunity to add more functionality to their program tomorrow, when they will add extensions to their network.</p> <p>Questions:</p> <ul style="list-style-type: none">• What are the use cases you identified in steps 1-3 of the Object Oriented Design Procedure?• What does a user need to be able to do?• How are you storing user information in your network object?• How can you access a specific user?	<p>Work in pairs or groups of at most 4 to create the methods you will need for your network class. Consider the following possibilities:</p> <ul style="list-style-type: none">• Add user• Add connection• Print network (using <code>__str__</code>)	60 mins
<p>If students are done, suggest they try one of the extension projects.</p>	<p>Extensions for Early Finishers:</p> <ul style="list-style-type: none">• Can we add a <code>remove_user</code> method? How might this “break” parts of your code?	

Building a Testing Interface (30 minutes)

Remind students about the text adventure game they built two weeks ago. Set a timer to 30 minutes and have students build a testing interface for their classes.	Work in your groups to build a main function that helps you run different functions of your program.	30 mins
---	--	---------

Presentations (15 mins)

- Run a **Code Brainstorm**.
- After running the Brainstorm, remind girls that there will be additional work time tomorrow.

Reflection (15 mins)

Allow students time to independently reflect on the following questions in their journals:

- How did using objects make this project easier? Harder?
- What is one problem that you are proud you solved today? How did you solve it?

Wrap-Up (15 mins)

Give a recap of the big ideas of the day:

- Objects - you can create your own variable types with their own unique methods.
- Testing and Revising.
- Collaborative Problem Solving

Ask girls to share their working definition of an object or an example to **Loop** 

Run a **Rose Bud Thorn**.
