

# Lesson 7.01: User-Defined Types (Classes) ## Learning Objectives Students will be able to... \* Define and identify: **class**, **instance**, **object**, **attributes** \* Create a class and instantiate \* Add attributes to an instance \* Manipulate instances and attributes through a function ## Materials/Preparation \* [Do Now] \* [Example] \* [Lab - Create a Color Class] ([printable lab document]) ([editable lab document]) \* [Associated Reading - section 7.1](https://tealsk12.gitbook.io/intro-cs-2/readings#7-1) \* Read through the do now, lesson, and lab so that you are familiar with the requirements and can assist students ## Pacing Guide | **Duration** | **Description** | | ----- | ----- | | 5 Minutes | Do Now | | 10 Minutes | Lesson | | 35 Minutes | Lab | | 5 Minutes | Discussion | ## Instructor's Notes ### 1. Do Now \* Display the Do Now on the board. ### 2. Lesson ##### Discuss the Do Now \* Ask the students what data type they thought would be helpful. \* They probably noticed that this was going to be difficult to do concisely without some other data type. \* Acknowledge that we need something that says "I am type Pet, which has three different qualities or attributes". ##### Instruction - Class \* A user-defined type. \* On board write syntax for creating a class: `class Pet:` ##### Instruction - Object \* The basis of object-oriented programming (OOP). \* Objects correspond to a real world thing that has certain attributes associated with it. ##### Instruction - Instantiated \* You can create **instances** of a class by using `Pet()` (`my_pet_1` is an instance of the `Pet` class). \* If you check the type of an instance it will be `Pet`. \* Instances are mutable, they can be changed or updated. ##### Demonstration \* Show [Example] on board to demonstrate the `Pet` class. \* Ask students what the difference is between this and the do now. \* Have the students identify where the class is created. \* Next have the students identify where the class is instantiated. \* Ask students what they think `pet.full_name` will do. \* This is a good time to explain the concept of an attribute. ##### Instruction - Attribute \* Values assigned to an instance. \* Note that you can also create functions that take in classes and use their attributes. ### 3. Lab \* Students will create RGB colors using a `Color` class. \* They will then create a function to merge two colors. \* They can check their colors on the linked RGB website in the lab. ### 4. Debrief \* Write down any questions you still have about the new terms you learned today? \* Is there anything you still need more clarification on? ## Accommodation/Differentiation Given the big leap taken today with much new terminology and challenging concepts, it's quite possible students will need additional class time to digest the information and finish the lab. For students that are quickly picking up the concepts, have them create their own unique class or have them help explain to struggling students with their own examples what a class, object, instance, or attribute is. ## Forum discussion [Lesson 7.01: User-Defined Types (Classes) (TEALS Discourse Account Required)](https://forums.tealsk12.org/c/2nd-semester-unit-7-classes/lesson-7-01-user-defined-types-classes) [Do Now]:do\_now.md.html [Lab - Create a Color Class]:lab.md.html [Example]:example.md.html [printable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/7\_unit/01\_lesson/lab.pdf [editable lab document]: https://github.com/TEALSK12/2nd-semester-introduction-to-computer-science/raw/master/units/7\_unit/01\_lesson/lab.docx