



## Lesson 2.2: Creating and Storing Objects (Instantiation)

<https://codehs.com/course/6165/lesson/2.2>

<b>Description</b>	In this lesson, students will create and use constructors. The constructor, or signature of a class, allows for the creation of a new object. Students will create objects by calling constructors with parameters. Parameters are values that are passed into a constructor. These are referred to as <i>actual parameters</i> . This lesson corresponds with AP Computer Science A topic 2.2.
<b>Objective</b>	Students will be able to: <ul style="list-style-type: none"><li>• Create and use constructors</li><li>• Create objects by calling constructors with parameters</li></ul>
<b>Activities</b>	<a href="#">2.2.1 Video: Creating and Storing Objects</a> <a href="#">2.2.2 Check for Understanding: Quiz: Constructors</a> <a href="#">2.2.3 Example: The Rectangle Class</a> <a href="#">2.2.4 Example: The Point Class</a> <a href="#">2.2.5 Example: The Student Class</a> <a href="#">2.2.6 Exercise: Using the Rectangle Class</a> <a href="#">2.2.7 Exercise: Student GPA Field</a> <a href="#">2.2.8 Exercise: Instance Variables for Your Dog</a> <a href="#">2.2.9 Exercise: Pizza Time!</a> <a href="#">2.2.10 Badge: Instantiation Badge</a>
<b>Prior Knowledge</b>	<ul style="list-style-type: none"><li>• Declaring and initializing variables in Java</li><li>• Java data types (int, double, char, etc)</li><li>• Creating classes with instance variables</li></ul>
<b>Planning Notes</b>	<ul style="list-style-type: none"><li>• Students may notice that creating a <code>String string str =</code> and using a <code>Scanner scanner input = new Scanner(System.in);</code> are very similar to using a constructor. Since a <code>String</code> is a reference type, it is in fact an object! The <code>String</code> class and the <code>Scanner</code> class are client classes which are classes that have been created by someone else. More detailed information about this will come up in a later lesson.</li><li>• There is a handout associated with this lesson. Consider using it in class or as a homework assignment.</li></ul>
<b>Standards Addressed</b>	
<b>Teaching and Learning Strategies</b>	<b>Lesson Opener:</b>

- Have students brainstorm and write down answers to the discussion questions listed below. Students can work individually or in groups/pairs. Have them share their responses. [5 mins]

### Activities:

- Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding. [5-6 mins]
- Explore the *The Rectangle Class* example. [8 mins]
  - Describe the “path” of the actual parameters 10 and 3. They originate in the constructor, but where are they assigned, initiated, etc?
  - Create a new object using the constructor and print out the results.
- Explore the *The Point Class* example. [8 mins]
  - This example highlights how objects are saved even when new objects with new instance variables are created. How is this possible? Have students discuss in pairs or groups what they think is happening every time they call a new constructor.
- Explore the *The Student Class* example. [8 mins]
  - Have students locate the `toString()` method and change it to print something else. *Concatenation and return values will be covered in further detail in later lessons. Exposure and experimentation is the emphasis here.*
- Complete the *Using the Rectangle* exercise. [5 mins]
  - Have students locate and identify the attributes and methods in the `Rectangle.java` file.
- Complete the *Student GPA Field* exercise. [15 mins]
  - This exercise can be difficult because of the similar variable names used in different places. Encourage students to run code often to see any error messages that might come up. The error messages can help students determine what else is needed for the program to run correctly.
- Complete the *Instance Variables for Your Dog* exercise. [5 mins]
  - This exercise is a good differentiator if the previous exercise is too tough. It covers the same concepts with less edits.
- Complete the *Pizza Time!* exercise. [15 mins]
  - Encourage students to reference their notes and previous exercises, but to not copy and paste! Typing out the variables, constructors etc will help students solidify these new concepts.

### Lesson Closer:

- Have students reflect and discuss their responses to the end of class discussion questions. [5 mins]
- If there is time at the end of class, consider using the *Constructor* handout as additional practice or as an exit ticket. [5-10 min]

### Discussion Questions

#### Beginning of Class:

- What is the relationship between objects and classes?
  - *A class is a template that holds information about what a class object's state and behavior can be, and how objects can be used in programs.*
- What are some important things to think about when deciding what instance variables a class should have?
  - *What attributes could vary from object to object? What attributes might be used in a computation about the object?*

- List out some instance variables you would want if you were writing a Car class that represented a Car. What type would each instance variable be? What would the name of each instance variable be?
  - *private String make; private String model; private int year; etc.*

### End of Class:

- What is the purpose of a constructor?
  - *The constructor, or signature of a class, allows for the creation of a new rectangle object.*
- Describe how you would create a Car object from your Car class that you created in the beginning of class. What values would you give to the constructor?

```
public Car(String carMake, String carModel, int carYear)
{
    make = carMake;
    model = carModel;
    year = carYear;
}
```

- Describe how you would create a Student object out of the class below.
  - *Student firstStudent = new Student("Jennifer", "Smith", 11);*

```
public class Student
{
    private String firstName;
    private String lastName;
    private int gradeLevel;

    public Student(String fName, String lName, int grade)
    {
        firstName = fName;
        lastName = lName;
        gradeLevel = grade;
    }

    public String toString()
    {
        return firstName + " " + lastName + " is in grade: " + gradeLevel;
    }
}
```

### Resources/Handouts

[Pair Programming Guide](#)

[Constructors](#)

[Constructors \(Teacher\)](#)

## Vocabulary

Term	Definition
<a href="#">Instance Variables</a>	Used to store the state, or data of the object instances.

<a href="#">Constructor/Signature</a>	Allows for the creation of a new object. Consists of the constructor name and parameter list.
<a href="#">new</a>	Necessary keyword for instantiating a new class object.
<a href="#">Instantiate</a>	Create an instance of a class object.
<a href="#">Formal and Actual Parameters</a>	Formal parameters are the parameters outlined in the parameter list in the constructor, while actual parameters are the parameters that are input when a new instance of a class object is created.

<b>Modification: Advanced</b>	<b>Modification: Special Education</b>	<b>Modification: English Language Learners</b>
	<ul style="list-style-type: none"> <li>• Pair programming with another student</li> <li>• Print out slides for students to reference</li> <li>• Scaffolding or template with class, object skeletons</li> </ul>	<ul style="list-style-type: none"> <li>• Pair programming with another student</li> <li>• Print out slides for students to reference</li> <li>• Scaffolding or template with class, object skeletons</li> </ul>