# **Name:**

# ***Activity 3: Classes***

In this activity, we'll take a first look at object-oriented programming.

Classes provide a means of bundling data and functionality together.

### **Content Learning Objectives**

*After completing this activity, students should be able to:*

* Write a class definition that has several attributes and methods.
* Explain what a constructor is, when it is called, and what it does.
* Discuss what ``object-oriented'' means using concrete examples.

### **Process Learning Objectives**

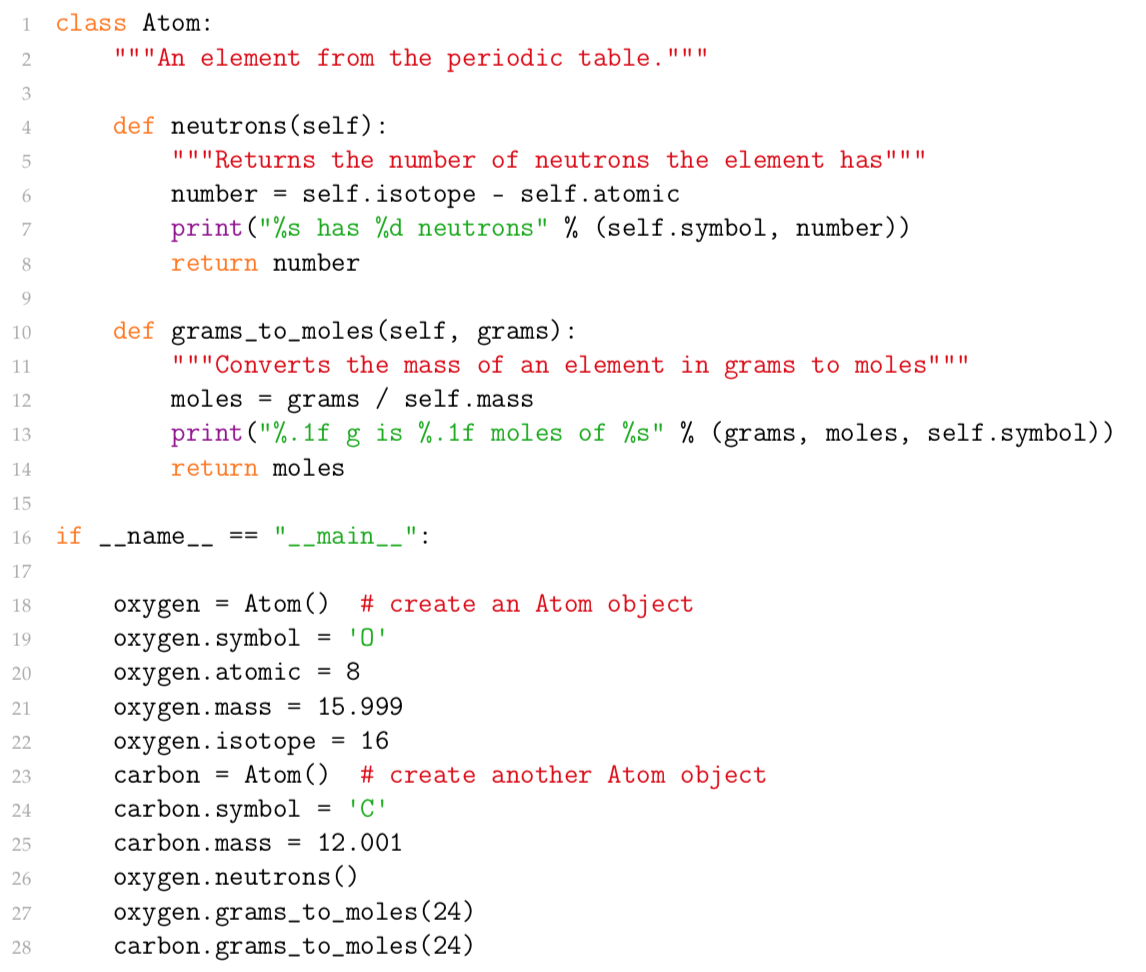
*After completing this activity, students should make progress toward:*

* Developing and testing the design of a program incrementally. (Problem Solving)

## 

## ***Model 1 Attributes and Methods***

Previously you have used built-in types like int, str, and list. Each of these types comes with its own **methods**, such as isdigit and append. You can create new types of data, and methods to go with them, by defining a class. Classes specify the **attributes** (instance variables) and methods (functions) that each object belonging to the class will have.



|  |  |
| --- | --- |
| ***Questions (15 min)*** | start  time: |

1. Examine the **class definition** (the top half of the code):
   1. What is the name of the class?

Atom

* 1. What are the names of the two methods?

Self, grams

* 1. What is the name of the first parameter for all methods?

Self

1. Now examine the "\_\_main\_\_" block of code:
   1. How many different Atom objects were created?

2

* 1. Identify the variable name of each object.

Oxygen

* 1. How many attributes were assigned in the oxygen object? List the names.

Symbol, atomic number, mass, and isotope

* 1. How do the number of arguments for each method call differ from the number of parameters specified in the method definition?

They make new ones like .symbol etc

1. How does the syntax referencing an attribute differ inside vs. outside the class definition?

1. When the grams\_to\_moles method is called (in the last two lines), what is the value of the self parameter?

1. Enter the expression type(oxygen) in a Python Shell. Explain the meaning and significance of the output.

1. Write code to create a new Atom object called hydrogen, and assign one of the attributes listed in Question #2c.

1. Call the neutrons method on carbon in a Python Shell. What is the reason for the error?

{{SAMPLE STOP}}

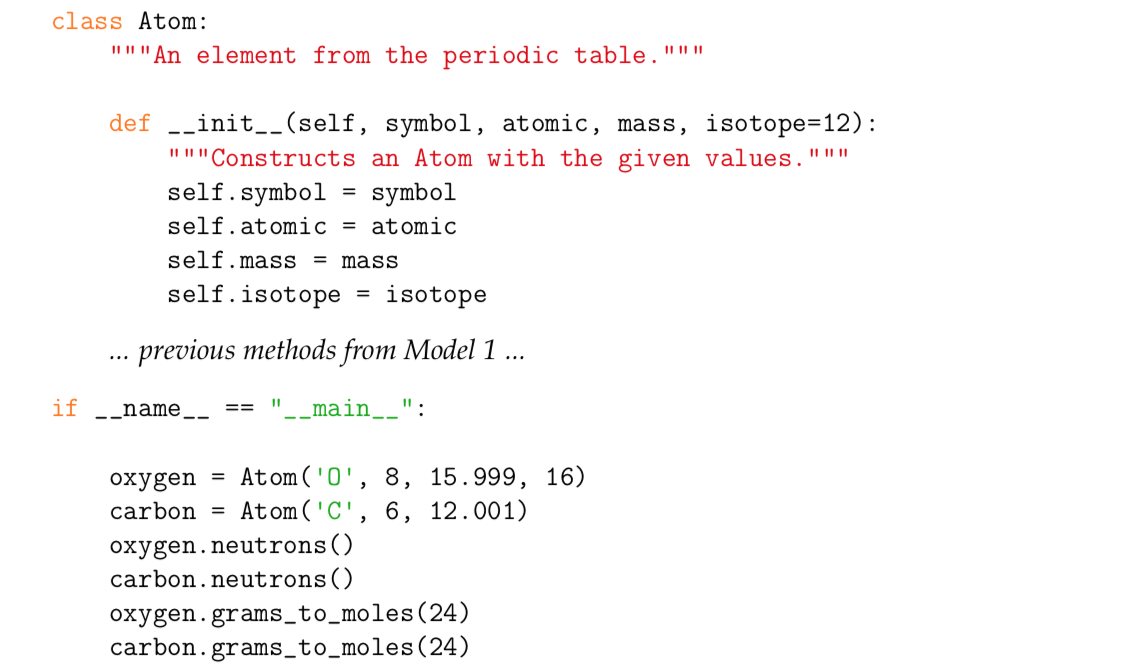
## 

## ***Model 2 Constructors***

For each class defined, you can provide a **constructor** that initializes attributes of a new object.

In Python, the constructor is always named \_\_init\_\_ (with two underscores before and after init). The constructor is called automatically when you create a new object.

Add the following constructor to the top of your Atom class. By convention, the constructor is typically the first method in a class definition. Also edit the "\_\_main\_\_" block of code as shown.



|  |  |
| --- | --- |
| ***Questions (15 min)*** | start  time: |

1. What is always the name of the constructor?

1. Although there is no direct call to the constructor, explain how you know this method is executed when an object is created.

1. Consider your answer to Question #7. What is one advantage of defining a constructor for a class?

1. In a Python Shell, try to create a new Atom object called hydrogen with only two arguments. Write your statement in the space below. What is the reason for the error you see?

1. When creating an object of the Atom class, what is the value of isotope if:
   1. four arguments are given?

* 1. three arguments are given?

1. Print the value of self.isotope in a Python shell.
   1. What is the reason for the error?

* 1. In order to eliminate this error, what should be printed instead?

1. Recall that a variable may be ``local'' (defined within a function), ``global'' (defined in the non-indented or "\_\_main\_\_" block of code), or ``built-in'' (part of Python itself).
   1. Explain why the isotope attribute is not a global variable.

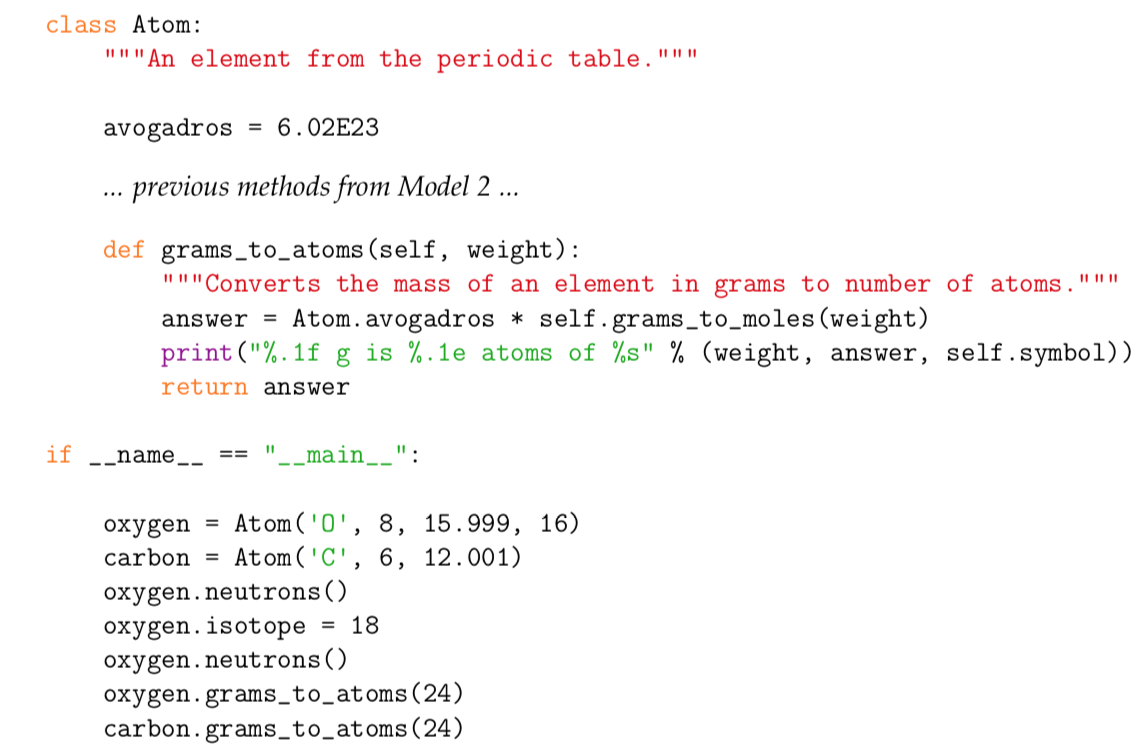
* 1. Explain why the isotope attribute is not a local variable.

* 1. How is each method of the class able to access the isotope attribute?

## 

## ***Model 3 Object-Oriented***

Edit the Atom class further to include the variable avogadros, the method grams\_to\_atoms, and the modified "\_\_main\_\_" block of code. Note that **class variables** (like avogadros) are typically defined before the \_\_init\_\_ method.



|  |  |
| --- | --- |
| ***Questions (15 min)*** | start  time: |

1. Examine the grams\_to\_moles method (from Model 1):
   1. Identify the three main variables used in grams\_to\_moles:

* 1. For each variable, what is its scope? (local or global)

1. What determines whether a variable is defined as an attribute or a local variable?

1. Now examine the grams\_to\_atoms method (from Model 3).
   1. What variable was initialized in the Atom class outside the constructor and methods?

* 1. How does the syntax of a class variable differ from an instance variable?

1. Would it be possible to rewrite the grams\_to\_atoms method as a function instead? If so, explain how the function would differ.

1. How would you rewrite the line oxygen.grams\_to\_atoms(24) to call the function defined in the previous question?

1. Consider the built-in str class:
   1. Given the statement s = "Hello", what data is stored in the str object?

* 1. Show an example line of code that calls the upper method on the object s.

* 1. If the upper method were defined as a global function instead, how would you call it?

1. Based on the previous two questions, explain what the term ``object-oriented'' means.

1. Summarize the advantages you perceive for writing code as methods in classes in comparison to functions.