Section Review

Learn to Code with Ruby

Methods

- Objects have **methods**, which are commands we can send to the object.

- Objects have exclusive and shared methods. There are things only a String can do, things only an Integer can do, and things that both objects can do.

- The object that we invoke the method upon is called the **receiver**.

Method Invocation

- Invoke a method by writing a dot and the method name.
- Ruby supports parentheses after the method name but the community advises against it.
- A method produces a **return value**, which is the final *output* of the method.
- Method chaining involves arranging a sequence (chain) of methods in order. A method may produce a new Ruby object, which will have its own set of methods.

```
“hello world”.length

“RUBY”.downcase

1.succ

3.14.class

“5”.to_i.pred
```

The NoMethodError Exception

- Ruby raises the **NoMethodError** exception when a method does not exist on an object.

The nil Object

- The **nil** object represents emptiness, nothingness, the absence of a value.

- We can declare **nil** in line but usually we'll be receiving it as the return value of a method. **puts** is an example of a method that returns **nil**.

String Interpolation

- String interpolation is the process of injecting content into a string. That content can be anything from a variable to a Ruby expression.

- Use a **hashtag** followed by opening and closing **curly braces**.

```
x = 5

y = 8

puts “The sum of #{x}
and #{y} is #{x + y}!”
```

The gets Method

- The **gets** (**get string**) method receives a string from a user via input.
- The entry will include the newline (\n) from the user's Enter press. Call the **chomp** method on a string to remove a newline character from its end.
- Use a variable to preserve the user's entry if you'd like to use it later.

The class Method

- A **class** is a blueprint for an **object**. The class is an abstract template, the object(s) are concrete entities.
- The object is called an **instance** of the class. The process of creating an object is called **instantiation**.
- The class method returns the class from which an object was made. For example, every string is an instance of the **String** class.

Object Conversion Methods

- The **to_s** method converts an object to a string.
- The **to_i** method converts an object to a integer.
- The **to_f** method converts an object to a float.