Section Review

Learn to Code with Ruby

The File Class

- The **File.open** class method opens a text file. By default, it opens it in read-only mode.
- Iterate over the **File** object with the **each** method to read the file's contents line-by-line.
- Pass a second argument of **w** (write) to the **open** method to permit Ruby to write to the file. Provide a block to specify what to do with the file.
- The **file.puts** method writes a line to the file (with a line break).
- The **file.write** method writes a line to the file without a line break.

The rename and delete Methods

- The **File.rename** method renames a file on your system. Provide the current name of the file and the desired new name.
- The **File.delete** method deletes a file on your system.
- Use the **File.exist?** predicate method to confirm that a file exists.

Command Line Arguments (ARGV)

- We can pass **command line arguments** to a Ruby program when we run it.
- The values flow into the program as strings. Ruby captures them in an **ARGV** array. The array is automatically available.

The load Method

- The **load** method imports a Ruby file into another.
- Ruby will look for the file in the current directory. Include the **.rb** file extension.
- Ruby will load/parse the specified file each time it sees the **load** method.

The require and require_relative Methods

- The **require** method also imports a file but caches it. Subsequent calls to the **require** method for the same file do nothing.
- **require** defaults to looking for the file in the Ruby installation directory. Use ./ in front of the file name to tell it to look in the current directory.
- The complementary **require_relative** method looks for the Ruby file in the same directory as the current file.
- You can omit the **rb** extension with the **require** and **require_relative** methods.