**Part 1:**

***save\_emp\_records.py***

**Text

Description automatically generated**

****

***read\_emp\_records.py***

***Text

Description automatically generated***

******

This program involves writing to files mechanics, as well as reading the input text from the produced file. Even though the textbook presents a way of first opening and then closing the file manually, I decided to implement more pythonic way of writing or reading the content of the file. To do this you invoke *“with open(<file\_name.txt>, <mode>) as <file\_object\_name:”* structure. The main benefit of this approach includes automatic closing and saving of the last changes made to the document.

**Part 2:**

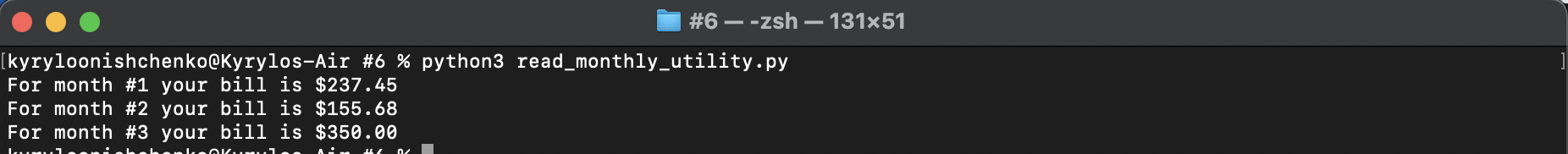
***save\_monthly\_utility.py***

**Text

Description automatically generated**



***read\_monthly\_utility.py***





In this program, I have implemented troubleshooting skills by implement try/except block by passing additional *ValueError* parameter in order to produce value error message in case the input value is incorrect. When it comes to reading a written file, the author of the book suggest to invoke .readline() method that reads one entire line of the file and produces a string. It is okay to implement this approach for small applications, however when you reuse the method over and over it fills up RAM quicker this making your application work slower. In my program, I used .readlines() method on the file object that return a list object of each line as an element of that object, which you can iterate over with a for loop and read it line by line way faster and more efficiently.