```
 1 # ------------------------------------------------------------------------ #
 2 # Title: Assignment 07
 3 # Description: Demonstrates Pickling (Saving Data in Binary Format)
 4 #              and Structured Error Handling
 5 # ChangeLog (Who,When,What):
 6 # RRoot,1.1.2030,Created started script
 7 # RRoot,1.1.2030,Added code to complete assignment 5
 8 # CMathews,8.15.2021,Modified code to complete assignment 6
 9 # CMathews,8.17.2021,Modified output formatting
10 # CMathews,8.23.2021,Modified code to complete assignment 7
11 # ------------------------------------------------------------------------ #
12
13 import pickle
14
15 # Data ------------------------------------------------------------------- #
16 # Declare variables and constants
17 strFileName = "ToDoList.dat"  # The name of the data file
18 objFile = None  # An object that represents a file
19 dicRow = {}  # A row of data separated into elements of a dictionary {Task,Priority}
20 lstTable = []  # A list that acts as a 'table' of rows
21 strChoice = ""  # Captures the user option selection
22 strTask = ""  # Captures the user task data
23 strPriority = ""  # Captures the user priority data
24 strStatus = ""  # Captures the status of processing functions
25
26
27 # Processing  ------------------------------------------------------------ #
28 class Processor:
29     """  Performs Processing tasks """
30
31     @staticmethod
32     def read_data_from_file(strFileName, lstTable):
33         """ Reads data from a file into a list of dictionary rows
34
```

```
35          :param file_name: (string) with name of file:
36          :param list_of_rows: (list) you want filled with file data:
37          :return: (list) of dictionary rows
38          """
39          lstTable.clear()
40          try:
41              objFile = open(strFileName, "rb")
42              lstTable += pickle.load(objFile)
43              objFile.close()
44          except:
45              objFile = open(strFileName, "wb")
46
47      @staticmethod
48      def add_data_to_list(task, priority, list_of_rows):
49          """Adds a task and its priority to the list of dictionary rows
50
51          :param task: (string) with name of task:
52          :param priority: (string) with priority of task:
53          :param list_of_rows: (list) data you want added to list
54          :return: (list) of dictionary rows
55          """
56          dicRow = {"Task": task, "Priority": priority}
57          list_of_rows.append(dicRow)
58          return list_of_rows, '\nSuccess - the task has been added.'
59
60      @staticmethod
61      def remove_data_from_list(task, list_of_rows):
62          """Removes a task and its priority from the list
63
64          :param task: (string) of task to remove
65          :param list_of_rows: (list) of dictionary rows
66          :return: (list) of dictionary rows, (string) message
67          """
68          for item in list_of_rows:
```

```
            if task == item["Task"]:
                list_of_rows.remove(item)
                return list_of_rows, "\nSuccess - the task has been removed."
        else:
            return list_of_rows, '\nError - task not found.'

    @staticmethod
    def write_data_to_file(strFileName, list_of_rows):
        """Writes dictionary data to text file

        :param: strFileName: (object) of text file
        :param: list_of_rows: (list) of dictionary rows
        :return: (list) of dictionary rows, (string) message
        """
        objFile = open(strFileName, "wb")
        pickle.dump(list_of_rows, objFile)
        objFile.close()
        return list_of_rows, '\nSuccess - the data has been saved.'


# Presentation (Input/Output)  -------------------------------------------- #
class IO:
    """ Performs Input and Output tasks """

    @staticmethod
    def print_menu_tasks():
        """  Display a menu of choices to the user

        :return: nothing
        """
        print('''
        Menu of Options
        1) Add a new Task
        2) Remove an existing Task
```

```
103         3) Save Data to File
104         4) Reload Data from File
105         5) Exit Program
106         ''')
107         print()  # Add an extra line for looks
108
109     @staticmethod
110     def input_menu_choice():
111         """ Gets the menu choice from a user
112
113         :return: string
114         """
115         choice = str(input("Which option would you like to perform? [1 to 5] - ")).strip()
116         print()  # Add an extra line for looks
117         return choice
118
119     @staticmethod
120     def print_current_tasks_in_list(list_of_rows):
121         """ Shows the current Tasks in the list of dictionaries rows
122
123         :param list_of_rows: (list) of rows you want to display
124         :return: nothing
125         """
126         print("\n******* Current Tasks: *******\n")
127         for row in list_of_rows:
128             print(row["Task"] + " (" + row["Priority"] + ")")
129         print("\n*******************************")
130         print()  # Add an extra line for looks
131
132     @staticmethod
133     def input_yes_no_choice(message):
134         """ Gets a yes or no choice from the user
135
136         :return: string
```

```
        """
        return str(input(message)).strip().lower()

    @staticmethod
    def input_press_to_continue(optional_message=''):
        """ Pause program and show a message before continuing

        :param optional_message:  An optional message you want to display
        :return: nothing
        """
        print(optional_message)
        input('\nPress the [Enter] key to continue.')

    @staticmethod
    def input_new_task_and_priority():
        task = input("Task name: ")
        priority = input("Task priority: ")
        return task, priority

    @staticmethod
    def input_task_to_remove():
        task = input("What task would you like to remove? ")
        return task


# Main Body of Script  ------------------------------------------------------ #

# Step 1 - When the program starts, Load data from ToDoFile.dat.
Processor.read_data_from_file(strFileName, lstTable)  # read file data

# Step 2 - Display a menu of choices to the user
while True:
    # Step 3 Show current data
    IO.print_current_tasks_in_list(lstTable)  # Show current data in the list/table
```

```
171     IO.print_menu_tasks()  # Shows menu
172     strChoice = IO.input_menu_choice()  # Get menu option
173
174     # Step 4 - Process user's menu choice
175     if strChoice.strip() == '1':  # Add a new Task
176         strTask, strPriority = IO.input_new_task_and_priority()
177         lstTable, strStatus = Processor.add_data_to_list(strTask, strPriority, lstTable)
178         IO.input_press_to_continue(strStatus)
179         continue  # to show the menu
180
181     elif strChoice == '2':  # Remove an existing Task
182         strTask = IO.input_task_to_remove()
183         lstTable, strStatus = Processor.remove_data_from_list(strTask, lstTable)
184         IO.input_press_to_continue(strStatus)
185         continue  # to show the menu
186
187     elif strChoice == '3':  # Save Data to File
188         strChoice = IO.input_yes_no_choice("Save this data to file? (Y/N) - ")
189         if strChoice.lower() == "y":
190             lstTable, strStatus = Processor.write_data_to_file(strFileName, lstTable)
191             IO.input_press_to_continue(strStatus)
192         else:
193             IO.input_press_to_continue("Save Cancelled!")
194         continue  # to show the menu
195
196     elif strChoice == '4':  # Load File Data
197         print("Warning: Unsaved Data Will Be Lost!")
198         strChoice = IO.input_yes_no_choice("\nAre you sure you want to reload data from file? (Y/N) -  ")
199         if strChoice.lower() == 'y':
200             Processor.read_data_from_file(strFileName, lstTable)  # read file data
201             print("\nData has been reloaded.")
202             IO.input_press_to_continue()
203         else:
204             IO.input_press_to_continue("\nFile Reload Cancelled!")
```

```
205         continue  # to show the menu
206
207     elif strChoice == '5':  # Exit Program
208         IO.input_press_to_continue()
209         print("\nGoodbye!")
210         break  # and Exit
211
212     else:
213         print("Please select from the available menu options.")
214
```