Ass2：

Movie.py

*"""  
Name: Minxi Long  
URL: https://github.com/JCUS-CP1404/assignment-2---movies-2-minxilong  
"""  
# TODO: Create your Movie class in this file*class Movie:  
 *"""Represent a Movie Object"""* def \_\_init\_\_(self, title=**''**, year=0, category=**''**, is\_watched=False):  
 *"""Initialise the movies' name, year, category and whether watched"""* self.title = title  
 self.year = year  
 self.category = category  
 self.is\_watched = is\_watched  
  
 def \_\_str\_\_(self):  
 *"""Return details of movie object"""* if self.is\_watched:  
 return **'{} - {} ({}) is watched'**.format(self.title, self.year, self.category)  
 else:  
 return **'{} - {} ({}) have not watched'**.format(self.title, self.year, self.category)  
  
 def unwatched\_watched(self):  
 *"""Add unwatched movies to watched"""* self.is\_watched = True  
  
 def watched\_unwatched(self):  
 *"""Add watched movies to unwatched"""* self.is\_watched = False  
  
 pass

Moviecollection.py

*"""  
Name: Minxi Long  
URL: https://github.com/JCUS-CP1404/assignment-2---movies-2-minxilong  
"""  
# TODO: Create your MovieCollection class in this file*import csv  
from movie import Movie  
  
  
class MovieCollection(Movie):  
 *"""Get a list of Movie objects"""* def \_\_init\_\_(self):  
 *"""Initialise the list"""* super().\_\_init\_\_()  
 self.movies=[]  
  
  
 def \_\_str\_\_(self):  
 mstrs = []  
 for movie in self.movies:  
 mstrs.append(str(movie))  
 return **'**\n**'**.join(mstrs)  
  
  
 def add\_movie(self, movie):  
 *"""Add a single Movie object to the movies attribute"""* self.movies.append(movie)  
  
  
 def count\_watched\_movies(self):  
 *"""Get number of watched movies"""* count\_watched = 0  
 for movie in self.movies:  
 if movie.is\_watched == **'w'**:  
 count\_watched += 1  
 return count\_watched  
  
 def count\_unwatched\_movies(self):  
 *"""Get number of unwatched movies"""* count\_unwatched = 0  
 for movie in self.movies:  
 if movie.is\_watched == **'u'**:  
 count\_unwatched += 1  
 return count\_unwatched  
  
 def load\_movies(self, filename):  
 with open(filename, **'r'**) as file:  
 reader = csv.reader(file)  
 for line in reader:  
 title, year, category, is\_watched = line  
 title = title  
 year = int(year)  
 category = category  
 is\_watched == **'w'** movie = Movie(title, year, category, is\_watched)  
 self.movies.append(movie)  
  
 def save\_movies(self, filename):  
 with open(filename, **'w'**) as out\_file:  
 for movie in self.movies:  
 if movie.is\_watched:  
 is\_watched = **'w'** else:  
 is\_watched = **'u'** out\_file.write(**'{}, {}, {}, {}**\n**'**.format(movie.title, movie.year, movie.category, is\_watched))  
  
 def by\_key(self, key, movie):  
 *"""Sort by the key passed in"""*

key = key.lower()if key == **'title'**:  
 return movie.title  
 elif key == **'year'**:  
 return movie.year  
 elif key == **'category'**:  
 return movie.category  
 elif key == **'watched'**:  
 return movie.is\_watched  
  
 def sort(self, key):  
 self.movies.sort(key=lambda movie: (self.by\_key(key, movie), movie.title))  
  
 pass

A1\_classes.py

*"""  
Name: Minxi Long  
URL: https://github.com/JCUS-CP1404/assignment-2---movies-2-minxilong  
"""  
# TODO: Copy your first assignment to this file, then update it to use Movie class  
# Optionally, you may also use MovieCollection class*import csv  
from movie import Movie  
from moviecollection import MovieCollection  
  
  
def get\_data(filename):  
 *# Get movies data from csv file into a list* list=[]  
 file=open(filename).readlines()  
 for line in file:  
 line = line.strip() *# Remove the \n* parts = line.split(**','**) *# Separate the data into its parts* parts[1] = int(parts[1]) *# Make the number an integer* list.append(parts)  
 return list  
  
  
def list\_movies(list):  
 watched = 0  
 unwatched = 0  
 mark=0  
 for line in list:  
 *# Get the number of watched movies and unwatched movies* if line[3] == **'w'**:  
 watched += 1  
 else:  
 unwatched += 1  
 *# List the movies* print(**'{:2}. {:3^} {:<35} - {:>5} ({})'**.format(mark, **'\*'**if line[3]==**'u'** else **' '**, line[0], line[1], line[2]))  
 mark += 1  
 print(**'{} movies watched, {} movies still to watch**\n**'**.format(watched, unwatched))  
  
  
def watch\_movies(list):  
 *# See if all movies have watched* unwatched=0  
 for line in list:  
 if line[3]==**'u'**:  
 unwatched+=1  
 if unwatched==0:  
 print(**'No more movies to watch!**\n**'**)  
 return  
  
 print(**'Enter the number of a movie to mark as watched'**)  
 *# Ensure valid input* valid\_input=False  
 while not valid\_input:  
 try:  
 mark = int(input(**'>>> '**))  
 while mark < 0:  
 print(**'Number must be >= 0'**)  
 mark = int(input(**'>>> '**))  
 while mark > len(list):  
 print(**'Invalid movie number'**)  
 mark = int(input(**'>>> '**))  
 valid\_input = True  
 except:  
 print(**'Invalid input; enter a valid number'**)  
  
 *# Add an unwatched movie to watched* if list[mark][3]==**'u'**:  
 print(**'{} from {} watched**\n**'**.format(list[mark][0], list[mark][1]))  
 list[mark][3]=**'w'** return  
 else:  
 print(**'You have already watched {}**\n**'**.format(list[mark][0]))  
  
  
def add\_movies():  
 *# Ensure valid input* valid\_title = False  
 while not valid\_title:  
 title = input(**'Title: '**)  
 if title == **''**:  
 print(**'Input can not be blank'**)  
 else:  
 valid\_title = True  
  
 *# Ensure valid input* valid\_year = False  
 while not valid\_year:  
 try:  
 year = int(input(**'Year: '**))  
 while year<0:  
 print(**'Number must be >= 0'**)  
 year = input(**'Year: '**)  
 valid\_year = True  
 except:  
 print(**'Invalid input; enter a valid number'**)  
  
 *# Ensure valid input* valid\_cate = False  
 while not valid\_cate:  
 category = input(**"Category: "**)  
 if category == **''**:  
 print(**'Input can not be blank'**)  
 else:  
 valid\_cate = True  
  
 *# Add an unwatched movie to the list* print(**"{} ({} from {}) added to movie list**\n**"**.format(title, category, year))  
 return[title, year, category,**'u'**]  
  
  
def main():  
 *# Import csv file* filename = **'movies.csv'** list = get\_data(filename)  
 print(**"Movies To Watch 1.0 - by <Minxi Long>**\n**{} movies loaded"**.format(len(list)))  
 while True:  
 *# Get menu* choice = str(input(**"Menu:**\n**L - List movies**\n**A - Add new movie**\n**W - Watch a movie**\n**Q - Quit**\n**>>> "**)).upper()  
  
 *# When choice not in the menu* valid\_choice = **"LAWQ"** if choice not in valid\_choice:  
 print(**"Invalid menu choice.**\n**"**)  
  
 if choice == **'L'**:  
 list\_movies(list)  
  
 if choice == **'W'**:  
 watch\_movies(list)  
  
 if choice == **'A'**:  
 line = add\_movies()  
 list.append(line)  
  
 if choice== **'Q'**:  
 *# Write movies into csv file* out\_file = open(filename,**'w'**)  
 for line in list:  
 out\_file.write(line[0]+**','**+str(line[1])+**','**+line[2]+**','**+line[3]+**'**\n**'**)  
 out\_file.close()  
 print(**'{} movies saved to {}**\n**Have a nice day :)'**.format(len(list),filename))  
 break  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()

App.kv

*# TODO: Create your Kivy layout in the kv language here*BoxLayout:  
 orientation: 'horizontal'  
 BoxLayout:  
 orientation: 'vertical'  
 size\_hint\_x: 0.25  
 Label:  
 text: 'Sort by:'  
 size\_hint\_y: 0.1  
 Spinner:  
 id: sort  
 text: "Year"  
 values: ("Title", "Year", "Category", "Watched")  
 on\_text: app.sort\_movies(self.text)  
 size\_hint\_y: 0.1  
 Label:  
 text: 'Add New Movie...'  
 size\_hint\_y: 0.2  
 Label:  
 text: 'Title:'  
 size\_hint\_y: 0.1  
 TextInput:  
 id: title  
 size\_hint\_y: 0.1  
 multiline: False  
 Label:  
 text: 'Category:'  
 size\_hint\_y: 0.1  
 TextInput:  
 id: category  
 size\_hint\_y: 0.1  
 multiline: False  
 Label:  
 text: 'Year:'  
 size\_hint\_y: 0.1  
 TextInput:  
 id: year  
 size\_hint\_y: 0.1  
 multiline: False  
 Button:  
 text: 'Add Movie'  
 size\_hint\_y: 0.1  
 on\_press: app.add\_movie()  
 Button:  
 text: 'Clear'  
 size\_hint\_y: 0.1  
 on\_press: app.clear()  
 BoxLayout:  
 orientation: 'vertical'  
 Label:  
 id: top\_label  
 size\_hint\_y:0.1  
 BoxLayout:  
 id: movies  
 orientation: 'vertical'  
 Label:  
 id: bottom\_label  
 size\_hint\_y:0.1

Main.py

*"""  
Name: Minxi Long  
Date: 23/01/2021  
Brief Project Description: Movie Program  
GitHub URL: https://github.com/JCUS-CP1404/assignment-2---movies-2-minxilong  
"""  
# TODO: Create your main program in this file, using the MoviesToWatchApp class*from kivy.app import App  
from kivy.lang import Builder  
from kivy.uix.button import Button  
from movie import Movie  
from moviecollection import MovieCollection  
  
  
class MoviesToWatchApp(App):  
 *"""Main application to watch movies"""* def \_\_init\_\_(self, \*\*kwargs):  
 super().\_\_init\_\_(\*\*kwargs)  
 self.movie\_collection = MovieCollection()  
 self.movie\_collection.load\_movies(**"movies.csv"**)  
 self.sorted\_by = **"year"** def build(self):  
 self.title = **"Movies To Watch 2.0"** self.root = Builder.load\_file(**'app.kv'**)  
 self.sort\_movies(self.sorted\_by)  
 return self.root  
  
 def movie\_listed(self):  
 self.root.ids.movies.clear\_widgets()  
 self.root.ids.top\_label.text = **'To watch: {}. Watched: {}'**.format(self.movie\_collection.count\_unwatched\_movies,  
 self.movie\_collection.count\_watched\_movies)  
 for movie in self.movies:  
 button = Button(text=str(movie))  
 button.bind(on\_press=self.movie\_click)  
 button.movie = movie  
 self.root.ids.movies.add\_widget(button)  
 if movie.is\_watched:  
 button.background\_color = **'#6e94b9'** else:  
 button.background\_color = **'#b26c6c'** def movie\_click(self, button):  
 movie = button.movie  
 if movie.is\_watched:  
 movie.watched\_unwatched()  
 self.root.ids.bottom\_label.text = **'You have watched {}'**.format(movie.title)  
 else:  
 movie.unwatched\_watched()  
 self.root.ids.bottom\_label.text = **'You have unwatched {}'**.format(movie.title)  
 self.movie\_listed()  
  
 def sort\_movies(self, sorted\_by):  
 self.sorted\_by = sorted\_by  
 self.movie\_collection.sort(sorted\_by)  
 self.movie\_listed()  
  
 def add\_movie(self):  
 movie = Movie(self.root.ids.title.text, int(self.root.ids.year.text), self.root.ids.category.text, False)  
 self.movies.add\_movie(movie)  
 self.sort\_movies(self.sorted\_by)  
  
 def clear(self):  
 self.root.ids.title.text = **""** self.root.ids.year.text = **""** self.root.ids.category.text = **""** self.root.ids.bottom\_label.text = **""** def over\_app(self):  
 self.movies.save\_movies(**"movies.csv"**)  
  
 pass  
  
  
if \_\_name\_\_ == **'\_\_main\_\_'**:  
 MoviesToWatchApp().run()