# Author: Khrystian Clark  
# Date: 10/12/2020  
# Description: A basic Library Simulator that involves patrons, items and transactions.  
  
  
class LibraryItem:  
 *"""Library class that holds item, title, location, person with book,  
 person requesting book, and date data members established"""* library\_item\_id = ''  
 title = ''  
 location = ''  
 checked\_out\_by = ''  
 requested\_by = ''  
 date\_checked\_out = 0  
  
 def \_\_init\_\_(self, item\_id, title):  
 *"""Initializing method for the private data members, including item id's and the title of the item"""* self.library\_item\_id = item\_id  
 self.title = title  
 self.checked\_out\_by = None  
 self.requested\_by = None  
 self.location = 'ON\_SHELF'  
 self.date\_checked\_out = 0  
  
 def get\_item\_id(self):  
 *"""Retrieves the item ID from the library\_item\_id"""* return self.library\_item\_id  
  
 def get\_title(self):  
 *"""Retrieves the title from the private data member 'title'."""* return self.title  
  
 def get\_location(self):  
 *"""Retrieves the location of a library item"""* return self.location  
  
 def get\_checked\_out\_by(self):  
 *"""Retrieves which patron a library item is checked out to"""* return self.checked\_out\_by  
  
 def get\_requested\_by(self):  
 *"""Refers to the patron that has requested an item in the library, if any"""* return self.requested\_by  
  
 def get\_date\_checked\_out(self):  
 *"""Retrieves the date the library item was checked out"""* return self.date\_checked\_out  
  
 def set\_item\_id(self, item\_id):  
 *"""Sets the library item id to a given value"""* self.library\_item\_id = item\_id  
  
 def set\_title(self, title):  
 *"""Sets the title of an item to a given value"""* self.title = title  
  
 def set\_location(self, location):  
 *"""Sets the library item's location to on hold, on the shelf or checked out"""* self.location = location  
  
 def set\_checked\_out\_by(self, person\_name):  
 *"""Gives a value to which patron has a certain item checked out"""* self.checked\_out\_by = person\_name  
  
 def set\_requested\_by(self, request\_name):  
 *"""Sets up who has requested any arbitrary library item"""* self.requested\_by = request\_name  
  
 def set\_date\_checked\_out(self, day):  
 *"""Sets the date that the library item was checked out"""* self.date\_checked\_out = day  
  
  
class Book(LibraryItem):  
 *"""Subclass inheriting from the LibraryItem that holds information on books in question"""* def \_\_init\_\_(self, item\_id, title, author):  
 super().\_\_init\_\_(item\_id, title)  
 self.author = author  
  
 def set\_author(self, author):  
 *"""Establishes the author of a book in the library"""* self.author = author  
  
 def get\_author(self):  
 *"""Retrieves the author information on a book in the library"""* return self.author  
  
 def get\_check\_out\_length(self):  
 *"""returns the number of days the item can be checked out"""* return 21  
  
  
class Album(LibraryItem):  
 *"""Subclass, inheriting from the LibraryItem that holds information on Albums in question"""* def \_\_init\_\_(self, item\_id, title, artist):  
 super().\_\_init\_\_(item\_id, title)  
 self.artist = artist  
  
 def set\_author(self, artist):  
 *"""Establishes an artists name of a library object"""* self.artist = artist  
  
 def get\_artist(self):  
 *"""Retrieves and artist's name from an object in the library"""* return self.artist  
  
 def get\_check\_out\_length(self):  
 *"""returns the number of days the item can be checked out"""* return 14  
  
  
class Movie(LibraryItem):  
 *"""Subclass, inheritng from the LibraryItem that holds information on Movies in question"""* def \_\_init\_\_(self, item\_id, title, director):  
 super().\_\_init\_\_(item\_id, title)  
 self.director = director  
  
 def set\_director(self, director):  
 *"""Establishes the director of a film in the library"""* self.director = director  
  
 def get\_author(self):  
 *"""Retrieves the director of a movie in the library"""* return self.director  
  
 def get\_check\_out\_length(self):  
 *"""returns the number of days the item can be checked out"""* return 7  
  
  
class Patron:  
 *"""Class that establishes patron identifier, names, their checked out items, and the amount they owe."""* def \_\_init\_\_(self, patron\_id, patron\_name):  
 *"""Initializes patron id's,names, checked out items and their fine amount"""* self.patron\_id = patron\_id  
 self.name = patron\_name  
 self.checked\_out\_items = []  
 self.fine\_amount = 0.00  
  
 def set\_patron\_id(self, patron\_id):  
 *"""Established the identifier for a specific patron"""* self.patron\_id = patron\_id  
  
 def set\_name(self, patron\_name):  
 *"""Establishes the name of a patron"""* self.name = patron\_name  
  
 def set\_fine\_amount(self, amount\_owed):  
 *"""Establishes the amount a patron owes"""* self.fine\_amount = amount\_owed  
  
 def get\_patron\_id(self):  
 *"""Retrieves the id of the patron"""* return self.patron\_id  
  
 def get\_name(self):  
 *"""Retrieves the name of a patron"""* return self.name  
  
 def get\_fine\_amount(self):  
 *"""Retrieves the amount a patron owes"""* return self.fine\_amount  
  
 def add\_library\_item(self, library\_item):  
 *"""adds a library to the list of items a patron has checked out"""* self.checked\_out\_items.append(library\_item)  
  
 def remove\_library\_item(self, library\_item):  
 *"""Removes a library item from a patron's checked out items"""* self.checked\_out\_items.remove(library\_item)  
  
 def amend\_fine(self, charges):  
 *"""An argument that increases or decreases the amount owed by a patron"""* self.fine\_amount = self.fine\_amount + charges  
  
 def get\_checked\_out\_items(self):  
 return self.checked\_out\_items  
  
  
class Library:  
 *"""The class that establishes what is and is not in the library, and the members of the library"""* holdings = []  
 members = []  
  
 def \_\_init\_\_(self):  
 *"""Initializes the list of library items and the current date"""* self.holdings = []  
 self.members = []  
 self.current\_date = 0  
  
 def set\_current\_date(self, day):  
 *"""Sets the current date as a day"""* self.current\_date = day  
  
 def get\_current\_date(self):  
 *"""Retrieves the current date"""* return self.current\_date  
  
 def add\_library\_item(self, library\_item):  
 *"""Takes a library item and adds it to the holdings list"""* self.holdings.append(library\_item)  
  
 def add\_patron(self, patron):  
 *"""Takes on a patron and adds them to the members list"""* self.members.append(patron)  
  
 def get\_library\_item\_from\_id(self, library\_item):  
 *"""This method returns the library item with an input ID parameter. Or else None"""* for item in self.holdings:  
 if library\_item == item.get\_item\_id():  
 return item  
 return None  
  
 def get\_patron\_from\_id(self, patron\_id):  
 *"""This method returns a patron that has a corresponding ID from with the members list"""* for member in self.members:  
 if patron\_id == member.get\_patron\_id():  
 return member  
 return None  
  
 def check\_out\_library\_item(self, person\_id, library\_item\_id):  
 *"""This method checks out a library otem to a patron, or else gives the items status/location"""* patron\_id = self.get\_patron\_from\_id(person\_id)  
 library\_item = self.get\_library\_item\_from\_id(library\_item\_id)  
 if patron\_id is None:  
 return "patron not found"  
 if library\_item is None:  
 return "item not found"  
 if library\_item.get\_location() == "CHECKED\_OUT":  
 return "item already checked out"  
 elif library\_item.get\_location() == "ON\_HOLD\_SHELF":  
 return "item on hold by other patron"  
 library\_item.set\_checked\_out\_by(patron\_id)  
 library\_item.set\_date\_checked\_out(self.current\_date)  
 library\_item.set\_location("CHECKED\_OUT")  
 if library\_item.get\_requested\_by() == person\_id:  
 library\_item.set\_requested\_by(None)  
 patron\_id.add\_library\_item(self, library\_item\_id)  
 return "check out successful"  
  
 def return\_library\_item(self, person\_id, library\_item\_id):  
 *"""This method returns an item to the library, or else gives its status or location"""* library\_item = self.get\_library\_item\_from\_id(library\_item\_id)  
 if library\_item is None:  
 return "item not found"  
 if library\_item.get\_location != "CHECKED\_OUT":  
 return "item already in library"  
 person\_id = self.get\_patron\_from\_id(person\_id)  
 person\_id.remove\_library\_item(library\_item\_id)  
 if library\_item\_id.get\_requested\_by is not None:  
 library\_item\_id.set\_location("ON\_HOLD\_SHELF")  
 else:  
 library\_item\_id.set\_location("ON\_SHELF")  
 library\_item\_id.set\_checked\_out\_by(None)  
 return "return successful"  
  
 def request\_library\_item(self, patron\_id, library\_item\_id):  
 *"""This method is used for a patron to request a library item and returns the item's status  
 if it not available"""* patron = self.get\_patron\_from\_id(patron\_id)  
 library\_item = self.get\_library\_item\_from\_id(library\_item\_id)  
 if patron is None:  
 return "patron not found"  
 if library\_item is None:  
 return "item not found"  
 if library\_item.get\_location == "ON\_HOLD\_SHELF":  
 return "item already on hold"  
 else:  
 library\_item.set\_requested\_by(patron\_id)  
 if library\_item.get\_location == "ON\_SHELF":  
 library\_item.set\_location("ON\_HOLD\_SHELF")  
 return "request successful"  
  
 def pay\_fine(self, patron\_id, amount\_paid):  
 *"""Retrieves a patron from the list, if they are a member, this gives the  
 remainder of the account balance after they paid an amount"""* patron\_id = self.get\_patron\_from\_id(amount\_paid)  
 if patron\_id is None:  
 return "patron not found"  
 else:  
 patron.ammend\_fine(amount\_owed)  
 return "payment successful"  
  
 def increment\_current\_date(self): #This is where i got stuck  
 *"""Takes the amount of days a patron has an item checked out and adds 10 cents for each  
 item that is overdue"""* current\_date = self.current\_date + 1  
 length = current\_date - get\_check\_out\_date  
 if length >= get\_check\_out\_length:  
 return "OVERDUE"  
 else:  
 return length.ammend\_fine(self.current\_date \* .10)