class User:

def \_\_init\_\_(self, user\_id, name, address, contact):

self.user\_id = user\_id

self.name = name

self.address = address

self.contact = contact

class Book:

def \_\_init\_\_(self, book\_id, title, author, publication\_date):

self.book\_id = book\_id

self.title = title

self.author = author

self.publication\_date = publication\_date

class Library:

def \_\_init\_\_(self):

self.users = {}

self.books = {}

self.loans = {}

def register\_user(self, user):

self.users[user.user\_id] = user

def add\_book(self, book):

self.books[book.book\_id] = book

def lend\_book(self, user\_id, book\_id):

if user\_id in self.users and book\_id in self.books:

loan\_id = len(self.loans) + 1

loan\_date = datetime.date.today()

due\_date = loan\_date + datetime.timedelta(days=14)

self.loans[loan\_id] = {'user\_id': user\_id, 'book\_id': book\_id, 'loan\_date': loan\_date, 'due\_date': due\_date}

return f'Book {book\_id} successfully lent to user {user\_id}. Due date: {due\_date}'

else:

return 'Invalid user ID or book ID.'

def return\_book(self, user\_id, book\_id):

for loan\_id, loan\_info in self.loans.items():

if loan\_info['user\_id'] == user\_id and loan\_info['book\_id'] == book\_id:

return\_date = datetime.date.today()

due\_date = loan\_info['due\_date']

days\_late = (return\_date - due\_date).days

if days\_late > 0:

fine\_amount = days\_late \* 2 # Assuming a fine of $2 per day late

return f'Book {book\_id} returned by user {user\_id}. Fine amount: ${fine\_amount}.'

else:

return f'Book {book\_id} returned by user {user\_id}. No fines.'

return 'Book not found in user\'s loan history.'

# Example usage:

if \_\_name\_\_ == '\_\_main\_\_':

library = Library()

user1 = User(1, 'John Doe', '123 Main St', 'john@example.com')

user2 = User(2, 'Jane Smith', '456 Oak Ave', 'jane@example.com')

book1 = Book(101, 'Introduction to Python', 'Alice Johnson', '2022-01-01')

book2 = Book(102, 'Data Science Handbook', 'Bob Williams', '2021-05-15')

library.register\_user(user1)

library.register\_user(user2)

library.add\_book(book1)

library.add\_book(book2)

print(library.lend\_book(1, 101))

print(library.lend\_book(2, 102))

print(library.return\_book(1, 101))

print(library.return\_book(2, 102))