

Code Similarity Analysis Report

Analysis Summary

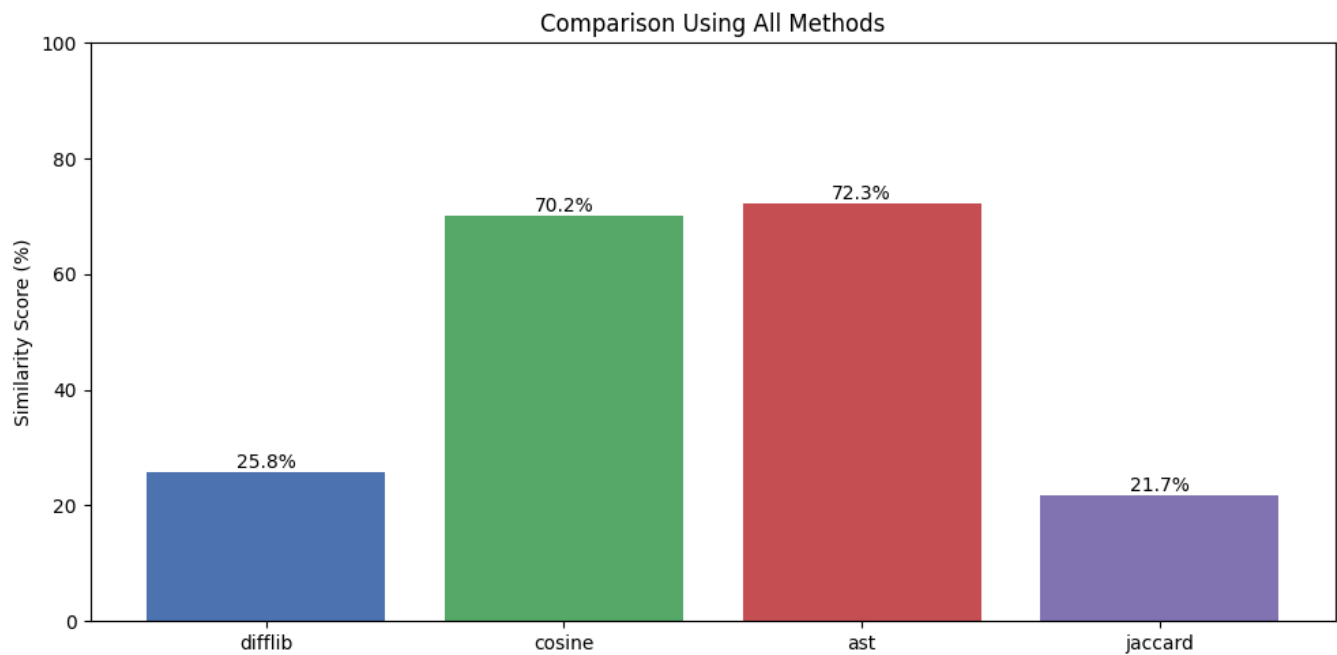
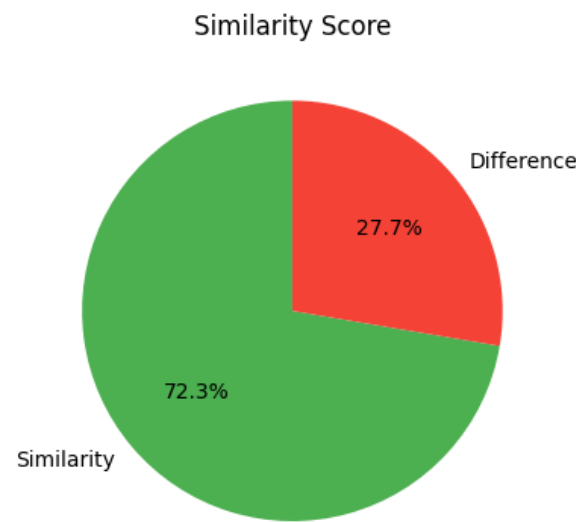
Comparison between: library_system_v1.py and library_system_v2.py

Selected Method: AST

Similarity Score: 72.33%

Plagiarism Threshold (70%) Exceeded: Yes

Similarity Visualizations



Code Similarity Report

Preprocessing Details

Before comparison, the following preprocessing steps were applied:

1. All comments were removed
2. All identifiers were normalized (variables ? vN, functions ? fN, etc.)

Original vs Preprocessed Code

Original library_system_v1.py:

```
class Book:
    def __init__(self, book_id, title, author, available=True):
        self.book_id = book_id
        self.title = title
        self.author = author
        self.available = available

    def __str__(self):
        status = 'Available' if self.available else 'Checked Out'
        return f"{self.book_id}: {self.title} by {self.author} ({status})"

class Member:
    def __init__(self, member_id, name):
        self.member_id = member_id
        self.name = name
        self.borrowed_books = []

    def borrow_book(self, book):
        if book.available:
            self.borrowed_books.append(book)
            book.available = False
            return True
        return False

    def return_book(self, book):
        if book in self.borrowed_books:
            self.borrowed_books.remove(book)
            book.available = True
            return True
        return False

# this is comment
class Library:
    def __init__(self):
        self.b...
```

Preprocessed library_system_v1.py:

```
class c0:

    def __init__(p0, p1, p2, p3, p4=True):
```

Code Similarity Report

```
self.book_id = book_id
self.title = title
self.author = author
self.available = available
```

```
def f0(p0):
    v0 = 'Available' if self.available else 'Checked Out'
    return f'{self.book_id}: {self.title} by {self.author} ({status})'
```

```
class c1:
```

```
def __init__(p0, p5, p6):
    self.member_id = member_id
    self.name = name
    self.borrowed_books = []

def f1(p0, p7):
    if book.available:
        self.borrowed_books.append(book)
        book.available = False
        return True
    return False
```

```
def f2(p0, p7):
    if book in self.borrowed_books:
        self.borrowed_books.remove(book)
        book.available = True
        return True
    return False
```

```
class c2:
```

```
def __init__(p0):
    self.books = []
    self.members = []

def f3(p0, p7):
    self.books.append(book)

def f4(p0, p8):
    self.member...
```

Original library_system_v2.py:

```
class BookItem:
    def __init__(self, id, name, writer):
        self.id = id
        self.name = name
        self.writer = writer
        self.in_stock = True
```

Code Similarity Report

```
def __repr__(self):
    status = "In Stock" if self.in_stock else "Out"
    return f"{self.id} - {self.name} ({self.writer}) [{status}]"
```

class User:

```
def __init__(self, uid, full_name):
    self.uid = uid
    self.full_name = full_name
    self.books_checked_out = []
```

```
def checkout(self, book):
    if book.in_stock:
        self.books_checked_out.append(book)
        book.in_stock = False
        return f"{book.name} borrowed."
    return "Not available."
```

```
def checkin(self, book):
    if book in self.books_checked_out:
        self.books_checked_out.remove(book)
        book.in_stock = True
        return f"{book.name} returned."
    return "Book not in record."
```

class LibraryManager:

```
def __init__(self):
    sel...
```

Preprocessed library_system_v2.py:

class c0:

```
def __init__(p0, p1, p2, p3):
    self.id = id
    self.name = name
    self.writer = writer
    self.in_stock = True

def f0(p0):
    v0 = 'In Stock' if self.in_stock else 'Out'
    return f'{self.id} - {self.name} ({self.writer}) [{status}]'
```

class c1:

```
def __init__(p0, p4, p5):
    self.uid = uid
    self.full_name = full_name
    self.books_checked_out = []

def f1(p0, p6):
    if book.in_stock:
        self.books_checked_out.append(book)
```

Code Similarity Report

```
    book.in_stock = False
    return f'{book.name} borrowed.'
return 'Not available.'
```

```
def f2(p0, p6):
    if book in self.books_checked_out:
        self.books_checked_out.remove(book)
        book.in_stock = True
        return f'{book.name} returned.'
    return 'Book not in record.'
```

```
class c2:
```

```
    def __init__(p0):
        self.catalog = []
        self.users = {}
```

```
    def f3(p0, p6):
        self.catalog.append(book)
```

```
de...
```

Detailed Differences (Preprocessed Code)

```
--- file1
+++ file2
@@ -1,81 +1,85 @@
class c0:

-     def __init__(p0, p1, p2, p3, p4=True):
-         self.book_id = book_id
-         self.title = title
-         self.author = author
-         self.available = available
+     def __init__(p0, p1, p2, p3):
+         self.id = id
+         self.name = name
+         self.writer = writer
+         self.in_stock = True

    def f0(p0):
-         v0 = 'Available' if self.available else 'Checked Out'
-         return f'{self.book_id}: {self.title} by {self.author} ({self.status})'
+         v0 = 'In Stock' if self.in_stock else 'Out'
+         return f'{self.id} - {self.name} ({self.writer}) [{self.status}]'

class c1:

-     def __init__(p0, p5, p6):
-         self.member_id = member_id
```

Code Similarity Report

```
-         self.name = name
-         self.borrowed_books = []
+ def __init__(p0, p4, p5):
+     self.uid = uid
+     self.full_name = full_name
+     self.books_checked_out = []

-     def f1(p0, p7):
-         if book.available:
-             self.borrowed_books.append(book)
-             book.available = False
-             return True
-         return False
+ def f1(p0, p6):
+     if book.in_stock:
+         self.books_checked_out.append(book)
+         book.in_stock = False
+         return f'{book.name} borrowed.'
+     return 'Not available.'

-     def f2(p0, p7):
-         if book in self.borrowed_books:
-             self.borrowed_books.remove(book)
-             book.available = True
-             return True
-         return False
+ def f2(p0, p6):
+     if book in self.books_checked_out:
+         self.books_checked_out.remove(book)
+         book.in_stock = True
+         return f'{book.name} returned.'
+     return 'Book not in record.'

class c2:

    def __init__(p0):
-         self.books = []
-         self.members = []
+         self.catalog = []
+         self.users = {}

-     def f3(p0, p7):
-         self.books.append(book)
+ def f3(p0, p6):
+     self.catalog.append(book)

-     def f4(p0, p8):
-         self.members.append(member)
+ def f4(p0, p7):
+     self.users[user.uid] = user
```

Code Similarity Report

```
def f5(p0, p1):
-     return next((b for v1 in self.books if b.book_id == book_id), None)
+     for v1 in self.catalog:
+         if b.id == id:
+             return b
+     return None

- def f6(p0, p5):
-     return next((m for v2 in self.members if m.member_id == member_id), None)
+ def f6(p0, p4):
+     return self.users.get(uid)

def f7(p0):
-     return [str(book) for v3 in self.books]
+     return [repr(b) for v1 in self.catalog]

def f8(p0):
-     return [f'{m.member_id}: {m.name}' for v2 in self.members]
+     return [f'{uid}: {u.full_name}' for v2, v3 in self.users.items()]

- def f1(p0, p5, p1):
-     v4 = self.find_member_by_id(member_id)
-     v3 = self.find_book_by_id(book_id)
-     if member and book:
-         return member.borrow_book(book)
-     return False
+ def f9(p0, p4, p8):
+     v4 = self.search_user(uid)
+     v5 = self.search_book(bid)
+     if user and book:
+         return user.checkout(book)
+     return 'User or Book not found.'

- def f2(p0, p5, p1):
-     v4 = self.find_member_by_id(member_id)
-     v3 = self.find_book_by_id(book_id)
-     if member and book:
-         return member.return_book(book)
-     return False
+ def f10(p0, p4, p8):
+     v4 = self.search_user(uid)
+     v5 = self.search_book(bid)
+     if user and book:
+         return user.checkin(book)
+     return 'Invalid return attempt.'

if __name__ == '__main__':
-     v5 = Library()
-     library.add_book(Book(101, '1984', 'George Orwell'))
-     library.add_book(Book(102, 'To Kill a Mockingbird', 'Harper Lee'))
-     library.register_member(Member(1, 'Alice'))
-     library.register_member(Member(2, 'Bob'))
```

Code Similarity Report

```
- print('Books in library:')
- print('\n'.join(library.list_books()))
- library.borrow_book(1, 101)
- print('\nBooks after borrowing:')
- print('\n'.join(library.list_books()))
+ v6 = LibraryManager()
+ manager.insert_book(BookItem(201, 'The Hobbit', 'J.R.R. Tolkien'))
+ manager.insert_book(BookItem(202, 'Fahrenheit 451', 'Ray Bradbury'))
+ manager.enroll_user(User(11, 'Charlie'))
+ manager.enroll_user(User(12, 'Dana'))
+ print('Available books:')
+ print('\n'.join(manager.show_all_books()))
+ print('\nIssuing book to Charlie:')
+ print(manager.issue_book(11, 201))
+ print('\nCurrent books:')
+ print('\n'.join(manager.show_all_books()))
```