Class Exercise:

OOP v2

You are tasked with creating a simple library management system using object-oriented programming (OOP) principles in Python. The system should allow you to add books, display their information, borrow and return books, and keep track of the number of available books in the library. Additionally, implement inheritance to create a specialized class for a specific type of book.

In this exercise, we have defined three classes: Book, Novel (inheriting from Book), and Library. The Book class represents a generic book, while the Novel class is a specialized class for novels, including an additional attribute for the genre. The Library class is responsible for managing the books in the library.

The Book class has several methods and special methods:

* \_\_init\_\_(self, title, author, publication\_year): Initializes a book with the provided title, author, and publication year.
* \_\_str\_\_(self): Returns a string representation of the book.
* borrow(self): Marks the book as borrowed if it is currently available.
* return\_book(self): Marks the book as returned if it was borrowed.
* The Novel class inherits from Book and adds the genre attribute. It overrides the \_\_str\_\_() method to include the genre in the string representation.

The Library class has methods for managing the books:

* \_\_init\_\_(self): Initializes an empty list of books.
* add\_book(self, book): Adds a book to the library.
* display\_books(self): Displays information about all the books in the library.
* count\_available\_books(self): Counts and displays the number of available books in the library.

class Book:

    def \_\_init\_\_(self, title, author, publication\_year):

# Your code here

    def \_\_str\_\_(self):

# Your code here

    def borrow(self):

# Your code here

    def return\_book(self):

# Your code here

class Novel(Book):

    def \_\_init\_\_(self, title, author, publication\_year, genre):

# Your code here

    def \_\_str\_\_(self):

# Your code here

class Library:

    def \_\_init\_\_(self):

# Your code here

    def add\_book(self, book):

# Your code here

    def display\_books(self):

# Your code here

    def count\_available\_books(self):

# Your code here

# Example usage

library = Library()

book1 = Book("Python Programming", "John Smith", 2021)

book2 = Book("Data Science Essentials", "Jane Johnson", 2020)

novel1 = Novel("Pride and Prejudice", "Jane Austen", 1813, "Romance")

library.add\_book(book1)

library.add\_book(book2)

library.add\_book(novel1)

library.display\_books()

novel1.borrow()

book2.borrow()

library.count\_available\_books()

novel1.return\_book()

library.count\_available\_books()

Expected output:

Book List:

Title: Python Programming, Author: John Smith, Publication Year: 2021

Title: Data Science Essentials, Author: Jane Johnson, Publication Year: 2020

Title: Pride and Prejudice, Author: Jane Austen, Publication Year: 1813, Genre: Romance

Book borrowed successfully.

Book borrowed successfully.

Number of available books: 1

Book returned successfully.

Number of available books: 2