```
# Python file with functions and classes


# Function to calculate the area of a circle
def calculate_circle_area(radius):
    """Returns the area of a circle given its radius."""
    import math
    if radius <= 0:
        return "Invalid radius"
    return math.pi * radius ** 2


# Function to calculate factorial
def factorial(n):
    """Returns the factorial of a number."""
    if n < 0:
        return "Factorial not defined for negative numbers"
    elif n == 0 or n == 1:
        return 1
    else:
        result = 1
        for i in range(2, n + 1):
            result *= i
        return result


# Function to check if a number is prime
def is_prime(num):
    """Checks if a number is a prime number."""
    if num < 2:
        return False
    for i in range(2, int(num ** 0.5) + 1):
```

```
        if num % i == 0:
            return False
    return True

# Function to find the greatest common divisor (GCD)
def gcd(a, b):
    """Finds the greatest common divisor (GCD) of two numbers."""
    while b:
        a, b = b, a % b
    return a

# Class to represent a bank account
class BankAccount:
    """A simple Bank Account class to manage account operations."""

    def __init__(self, owner, balance=0):
        self.owner = owner
        self.balance = balance

    def deposit(self, amount):
        """Deposits a specified amount into the account."""
        if amount > 0:
            self.balance += amount
            return f"Deposited {amount}. New balance: {self.balance}"
        return "Invalid deposit amount."

    def withdraw(self, amount):
        """Withdraws a specified amount from the account."""
        if amount > self.balance:
```

```
            return "Insufficient balance."
        elif amount <= 0:
            return "Invalid withdrawal amount."
        else:
            self.balance -= amount
            return f"Withdrew {amount}. New balance: {self.balance}"

# Class to represent a rectangle
class Rectangle:
    """A class to represent a rectangle with length and width."""

    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        """Calculates the area of the rectangle."""
        return self.length * self.width

    def perimeter(self):
        """Calculates the perimeter of the rectangle."""
        return 2 * (self.length + self.width)

# Class to manage a collection of books in a library
class Library:
    """A simple Library class to manage book collections."""

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

```
    def add_book(self, book):
        """Adds a new book to the collection."""
        self.books.append(book)
        return f"Book '{book}' added to the library."

    def remove_book(self, book):
        """Removes a book from the collection."""
        if book in self.books:
            self.books.remove(book)
            return f"Book '{book}' removed from the library."
        return "Book not found."

    def list_books(self):
        """Lists all the books in the collection."""
        return f"Available books: {', '.join(self.books)}" if self.books else "No books available."

# Testing the functions and classes
if __name__ == "__main__":
    # Function tests
    print(calculate_circle_area(5))
    print(factorial(5))
    print(is_prime(11))
    print(gcd(54, 24))

    # BankAccount class test
    account = BankAccount("John Doe", 100)
    print(account.deposit(50))
    print(account.withdraw(30))
```

```
# Rectangle class test
rect = Rectangle(10, 5)
print(f"Area: {rect.area()}")
print(f"Perimeter: {rect.perimeter()}")

# Library class test
library = Library()
print(library.add_book("1984"))
print(library.add_book("To Kill a Mockingbird"))
print(library.list_books())
print(library.remove_book("1984"))
print(library.list_books())
```