class BookStall:

def \_\_init\_\_(self, book\_name, customer\_id, price):

self.book\_name = book\_name

self.customer\_id = customer\_id

self.price = price

def calculate\_discount(self):

if 1 <= self.customer\_id <= 100:

discount\_percentage = 50

elif 101 <= self.customer\_id <= 300:

discount\_percentage = 30

elif 301 <= self.customer\_id <= 400:

discount\_percentage = 20

elif 401 <= self.customer\_id <= 500:

discount\_percentage = 10

else:

discount\_percentage = 0

# Calculate discounted price

discounted\_price = self.price - (self.price \* discount\_percentage / 100)

return discount\_percentage, discounted\_price

def display\_details(self):

discount\_percentage, discounted\_price = self.calculate\_discount()

print(f"Price of the Book is: {self.price} MRP")

print(f"You are eligible for a discount of: {discount\_percentage}%")

print(f"Your discounted price for the book is: {discounted\_price:.2f} MRP")

def main():

try:

# Input details from the user

book\_name = input("Enter your Book Name: ").strip()

if not book\_name:

raise ValueError("Book name cannot be empty.")

customer\_id = int(input("Enter your Customer ID: "))

if customer\_id <= 0:

raise ValueError("Customer ID must be a positive number.")

price = float(input("Enter the Price of the Book: "))

if price <= 0:

raise ValueError("Price must be a positive number.")

# Create an instance of BookStall and process details

book\_stall = BookStall(book\_name, customer\_id, price)

book\_stall.display\_details()

continue\_choice = input("Do you want to continue? (yes/no): ").strip().lower()

if continue\_choice == "yes":

main()

else:

print("Thank you for using the Book Stall system!")

except ValueError as e:

print(f"Error: {e}")

print("Please try again.")

main()

main()