```
class BankAccount:
    def __init__(self, account_number, account_holder_name, initial_balance=0):
        self._account_number = account_number
        self._account_holder_name = account_holder_name
        self._account_balance = initial_balance

    def deposit(self, amount):
        if amount > 0:
            self._account_balance += amount
            print(f"Deposited ${amount}. New balance: ${self._account_balance}")
        else:
            print("Invalid deposit amount. Please enter a positive value.")

    def withdraw(self, amount):
        if amount > 0 and amount <= self._account_balance:
            self._account_balance -= amount
            print(f"Withdrew ${amount}. New balance: ${self._account_balance}")
        elif amount <= 0:
            print("Invalid withdrawal amount. Please enter a positive value.")
        else:
            print("Insufficient funds for withdrawal.")

    def display_balance(self):
        print(f"Account Balance for {self._account_holder_name}: ${self._account_balance}")


# Create an instance of BankAccount
account = BankAccount("12345", "John Doe", 1000)
```

```
# Test deposit and withdrawal
account.display_balance()
account.deposit(500)
account.withdraw(200)
account.display_balance()

class Player:
    def play(self):
        print("The player is playing cricket.")

class Batsman(Player):
    def play(self):
        print("The batsman is batting.")

class Bowler(Player):
    def play(self):
        print("The bowler is bowling.")

# Creating objects of Batsman and Bowler classes
batsman = Batsman()
bowler = Bowler()

# Calling the play() method for each object
batsman.play()
bowler.play()

def sort_students(student_list):
    # Sort the student list based on CGPA in descending order
    sorted_students = sorted(student_list, key=lambda student: student.cgpa, reverse=True)
```

```
    return sorted_students

# Define a Student class
class Student:
    def __init__(self, name, roll_number, cgpa):
        self.name = name
        self.roll_number = roll_number
        self.cgpa = cgpa

# Test the function with a list of student objects
students = [
    Student("Alice", "S001", 3.9),
    Student("Bob", "S002", 3.7),
    Student("Charlie", "S003", 4.0),
    Student("David", "S004", 3.8),
]

sorted_students = sort_students(students)

# Print the sorted list
for student in sorted_students:
    print(f"Name: {student.name}, Roll Number: {student.roll_number}, CGPA: {student.cgpa}")

def linear_search_product(product_list, target_product):
    indices = []
    for index, product in enumerate(product_list):
        if product == target_product:
            indices.append(index)
    return indices
```