

```
'''
Implement a function called sort_students that takes a list of student objects as
input and sorts the
list based on their CGPA (Cumulative Grade Point Average) in descending order. Each
student object
has the following attributes: name (string), roll_number (string), and cgpa
(float). Test the function
with different input lists of students.
'''
```

```
class Student:
```

```
def __init__(self, name, roll_number, cgpa):
    self.name = name
    self.roll_number = roll_number
    self.cgpa = cgpa
```

```
def sort_students(student_list):
    # Sort the list of students in descending order of CGPA
    sorted_students = sorted(student_list,
                              key=lambda student: student.cgpa,
                              reverse=True)
    # Syntax - lambda arg:exp
    return sorted_students
```

```
# Example usage:
```

```
students = [
    Student("Hari", "A123", 7.8),
    Student("Srikanth", "A124", 8.9),
    Student("Saumya", "A125", 9.1),
    Student("Mahidhar", "A126", 9.9),
]
```

```
sorted_students = sort_students(students)
```

```
# Print the sorted list of students
```

[illegible]

```
"""
Write a function called linear_search_product that takes the list of products and a
target product
name as input. The function should perform a linear search to find the target
product in the list and
return a list of indices of all occurrences of the product if found, or an empty
list if the product is not
found.
"""
```

```
def linearSearchProduct(productList, targetProduct):
    indices = []

    for index, product in enumerate(productList):
        if product == targetProduct:
            indices.append(index)

    return indices
```

```
# Example usage:
products = ["shoes", "boot", "loafer", "shoes", "sandal", "shoes"]
target = "shoes"
target2 = 'apple'
result = linearSearchProduct(products, target)
print(result)
```