**JOBSHEET 6**

**Sorting (Bubble, Selection, Insertion Sort)**



**Name**

Sherly Lutfi Azkiah Sulistyawati

**NIM**

2341720241

**Class**

1I

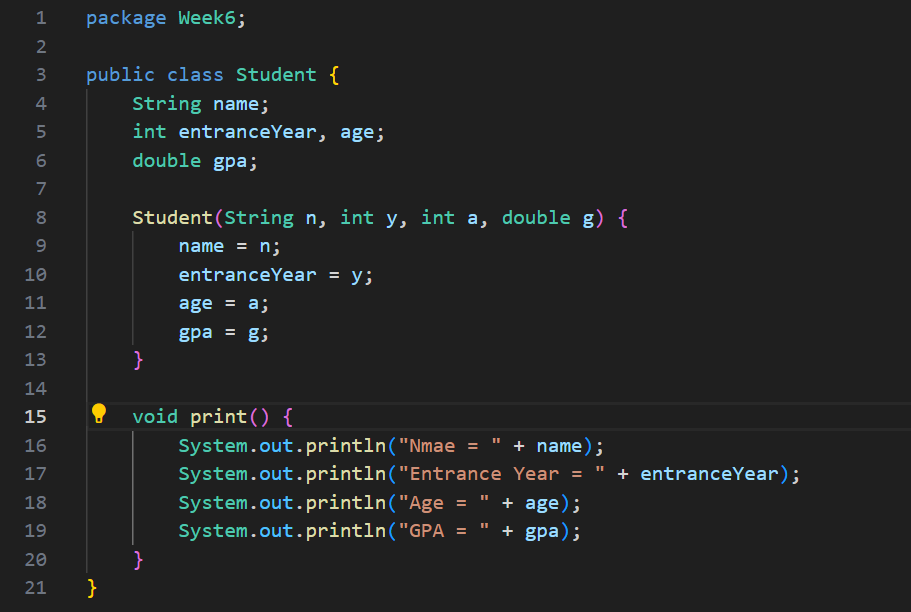
**Major**

Information Technology

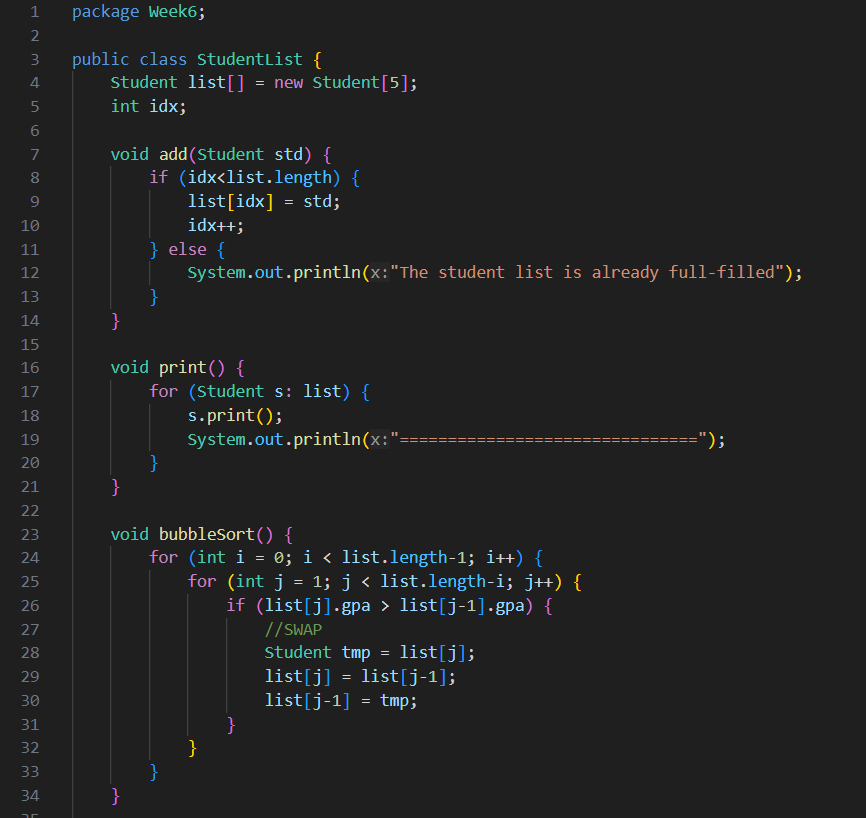
**Study Program**

D4 Informatics Engineering

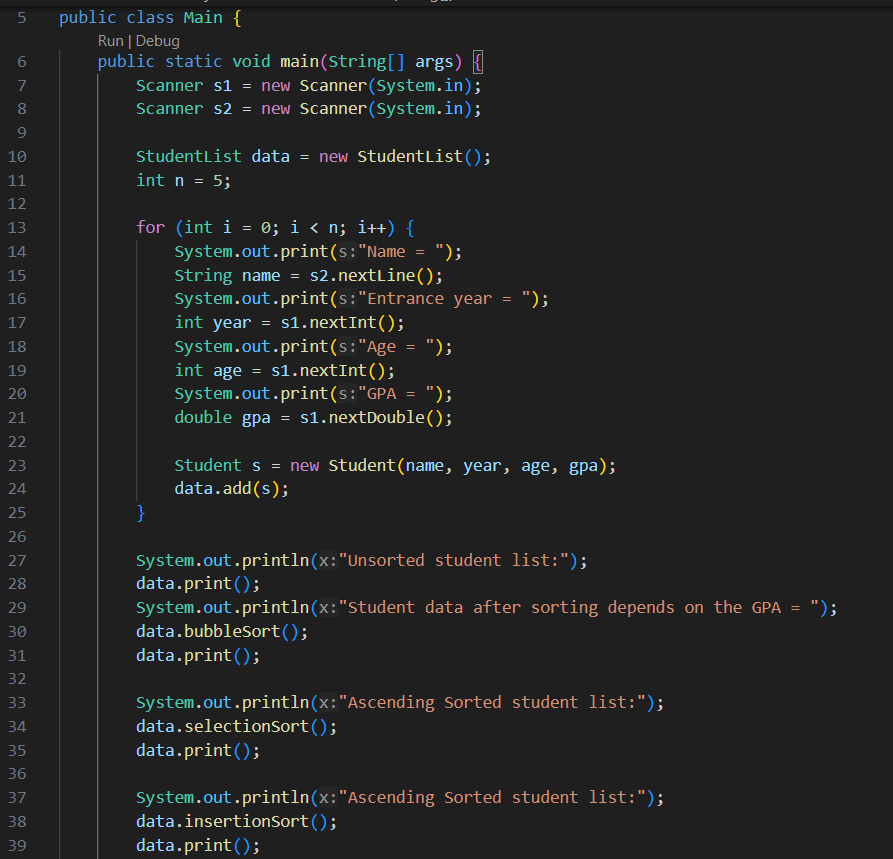
**Practicum 1: Create Student Class**

****

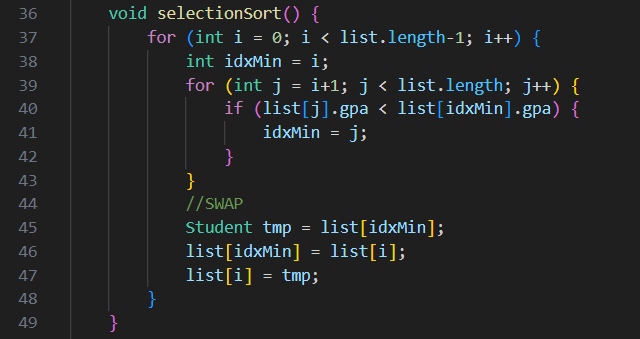
**Practicum 2: Create HighAchieverStudent Class**

****

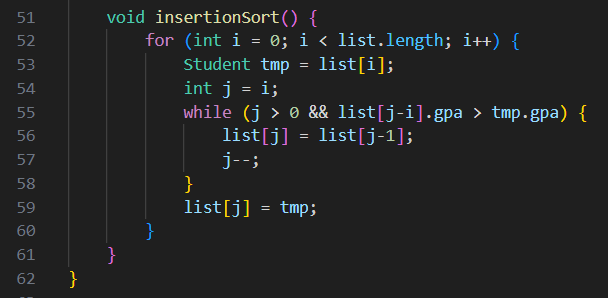
**Practicum 3: Create Main Class**

****

**Practicum 4: Add Selection Sort Process in HighAchieverStudent Class**

****

**Practicum 4: Add Insertion Sort Process in HighAchieverStudent Class**

****

**Question**

1. In which class we have a function to do sorting with bubble sort approach?

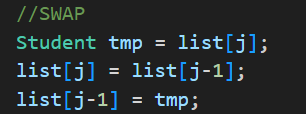
* The function to perform sorting with the bubble sort approach is located in the **HighAchieverStudent** class. The method is named **bubbleSort**.

1. In which class we have a function to do sorting with insertion sort approach?

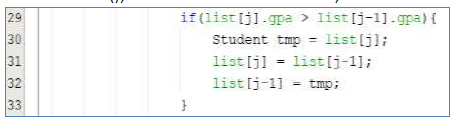
* The function to perform sorting with the insertion sort approach is also located in the **HighAchieverStudent** class. The method is named **insertionSort**.

1. What is the meaning of swapping process? Write the code to do the swapping process in the program above!

* The swapping process involves exchanging the positions of two elements in an array. In the provided program, the swapping process is used to correctly order the elements during sorting. Here's the code to perform the swapping process:



1. In bubbleSort(), there is these lines of code, what’s the function of it?



* In the bubbleSort method, the provided lines of code are responsible for comparing adjacent elements in the array and swapping them if they are in the wrong order. If the GPA of the current element (**list[j]**) is greater than the GPA of the previous element (**list[j-1]**), the two elements are swapped. This ensures that the higher GPA elements "bubble up" towards the end of the array.

1. Look at the loops inside the bubbleSort() method:



1. What’s the difference of loop *i* and loop *j*?

- The loop **i** iterates over each element of the array, starting from the first element and going up to the second-to-last element. It controls the number of passes through the array.

1. Why is the criteria of loop *i* is i<listStd.length-1?

- The condition **i < list.length - 1** ensures that the loop doesn't go beyond the last element of the array. Since each pass puts the largest unsorted element in its correct position, there is no need to compare the last element with any other elements.

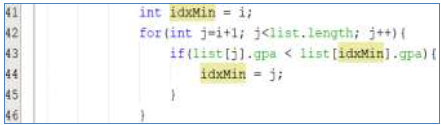
1. Why is the criteria of loop *j* is j<listStd.length-i?

- The loop **j** iterates over the unsorted portion of the array. It starts from index **1** (the second element) and goes up to **list.length - i**. This ensures that we only compare the unsorted portion of the array in each pass.

1. If the data in listStd is 50, how many loop *i* will happen? And how many bubble sort steps will be?

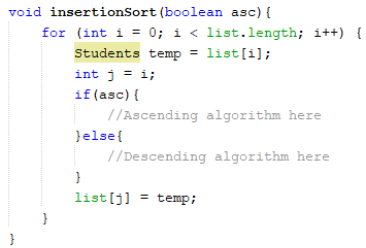
- If the data in **listStd** is 50, there will be 49 iterations of loop **i** and 49 bubble sort steps. This is because each iteration of loop **i** sorts one element to its correct position from the end.

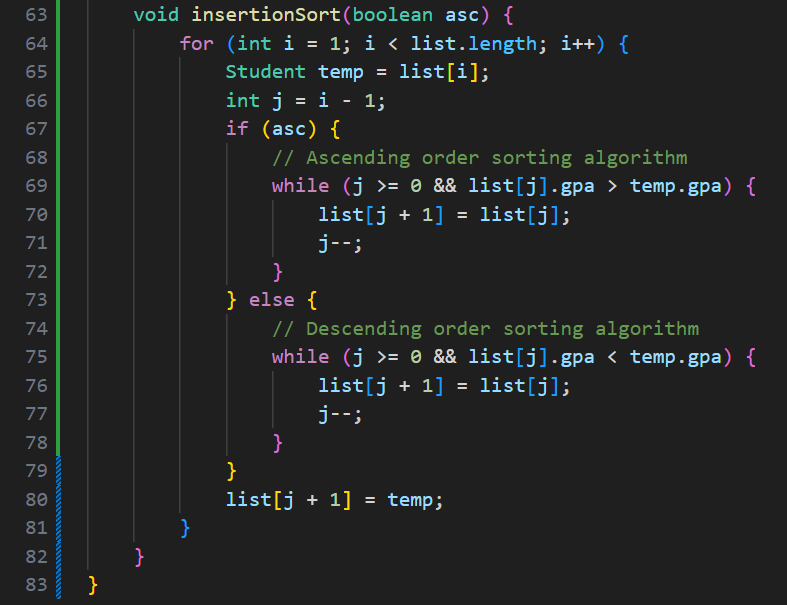
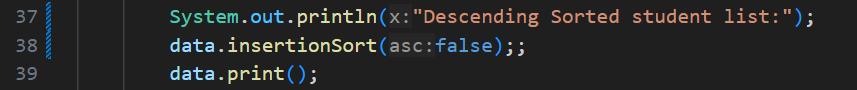
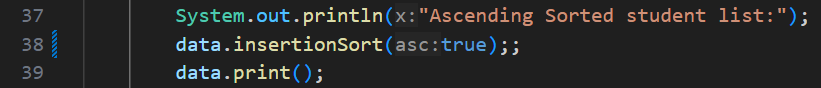
1. In selection sort method, there is these lines of code, what’s that for?



* In the selection sort method, the provided lines of code are responsible for finding the index of the minimum element in the unsorted portion of the array. This loop iterates over the unsorted portion of the array (starting from index **i+1**) and compares the GPA of each element with the GPA of the current minimum element (**list[idxMin]**). If a smaller GPA is found, the **idxMin** is updated to the index of the new minimum element.

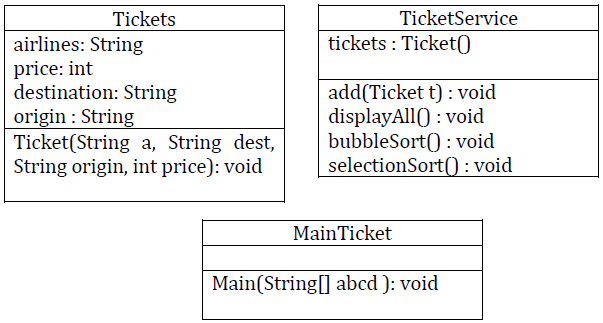
1. Change the insertionSort method so that the user has options to sort in either ascending or descending order. You can do it by adding a parameter, and this parameter’s value will be assigned through function calling in main class

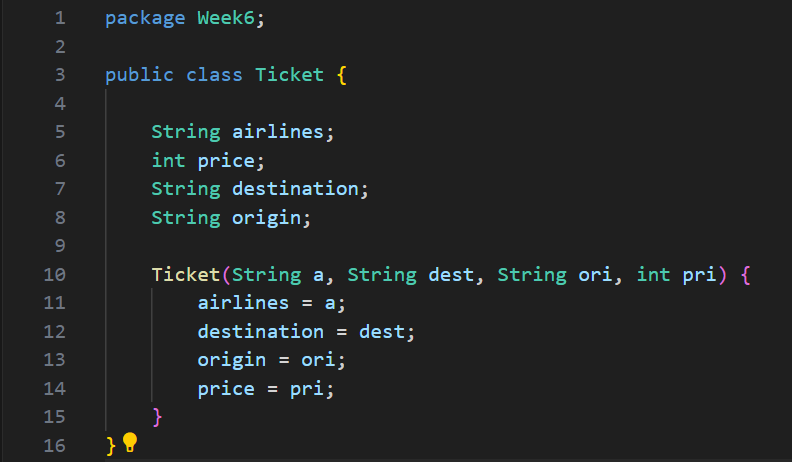


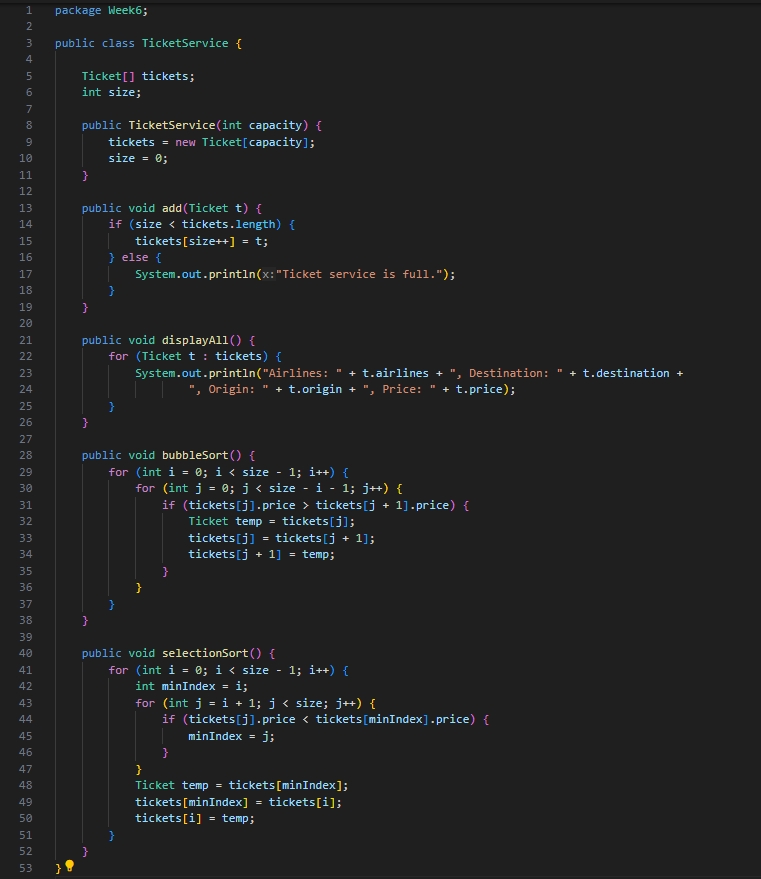
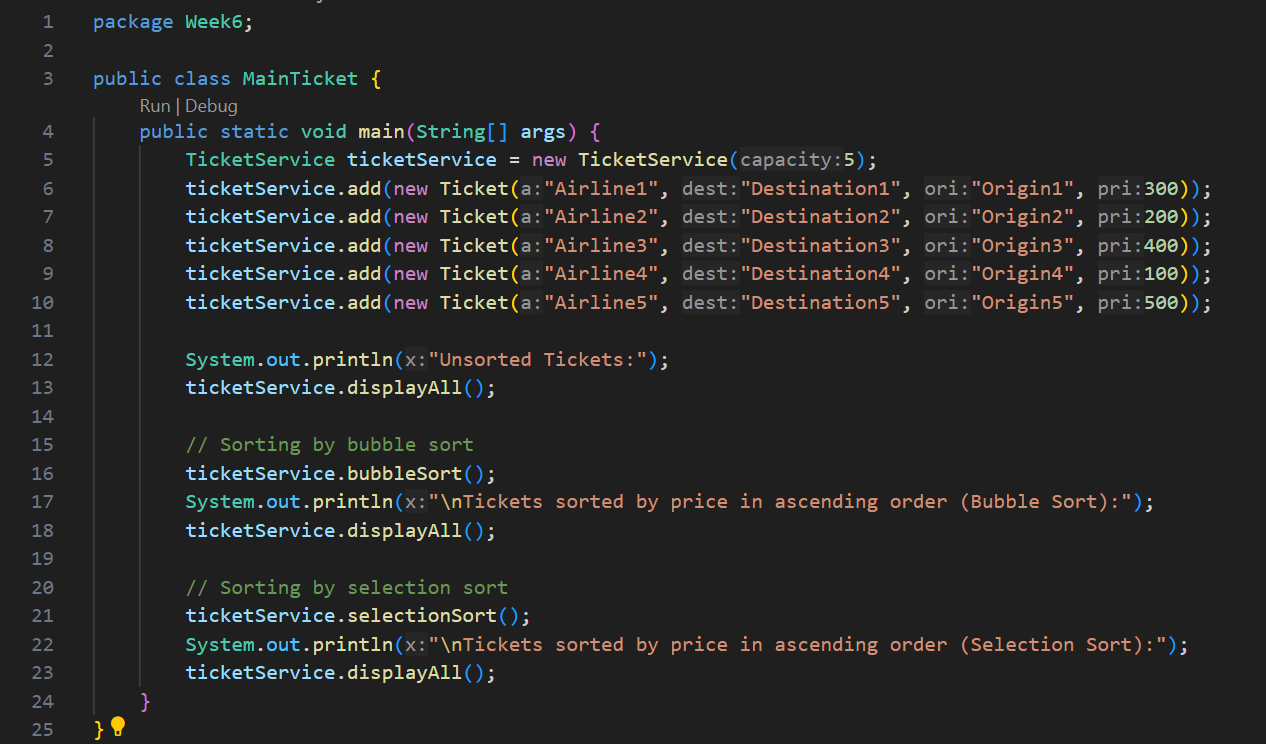
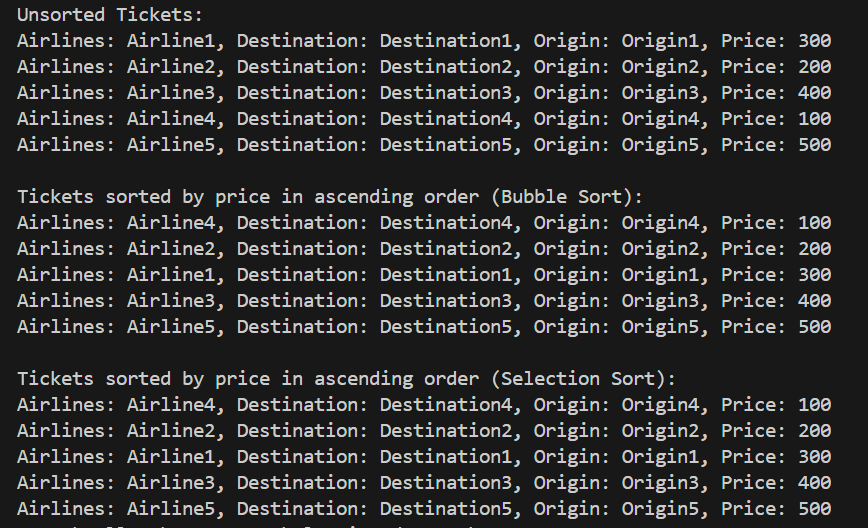
 

**Assignment**

1. There is a company that provide services in airplane ticket sales, they are developing a backend system for ticket reservation. One of its features is to display all available tickets based on filter from user. The ticket list must able to be sorted by the price in ascending and descending order. Implement these class diagrams in java program and create the sorting algorithm with **bubble sort** and **selection sort**





1. Premiere League in 2020 is already in half-season. In this season, Liverpool is the top of the list, the full list is displayed below



Change the standings list above to class diagram that has sorting club function based on highest to smallest points (in ascending order) with insertion sort algorithm. Take these following class diagrams as your reference:

