Part-1:

Part one is about checking in the seats in the plane.

At first, I create a Plane class. This class will assign a seat based on users’ input.

The key point of this class is I use an array to hold the whole seats which need to check in. Its data type is boolean. I initialize it to false. That means the seats are available. Then, after users check in, its status will change to true, unavailable.

And this one, checked will count the seats which are checked.

The method, checkIn does the work of checking in. When there are seats available, users can assign a seat. I use a for loop to realize it. And after each successfully completed section there is a break statement. If there are no seats available, they are presented with message, “no more seats are available for your selection.”

Ok, then we will see the main program. Firstly, I initialize an object, plan from the Plane class. Then, users can choose to input first class or economy class. According different input, we assign different data to the Plane class and perform different actions.

Part-2:

Part two is about sorting some invoices.

At first, I create an Invoice class. Private is PartNumber, Partdescription, Quantity and Price. I use tow constructors. One is default which initializes the private data. And the other is about these four arguments.

Actually, I use different function to finish different sorting. selectionSort, this function performs a selection sort on Invoice objects, arranging them in ascending PartDescription order. insertionSort, this function performs a insertion sort on Invoice objects, arranging them in descending Price order. bubbleSort, this function performs a bubble sort on Invoice objects, arranging them in ascending amount order.

In main program, firstly, I create and initialize the array of Invoice objects. Then, I sort them according different function.

Part-3:

Part three is a composition. There are two classes. Time2 is TimeCard’s supplier-class. Time2 is a class to setup the time. In the private, beside the data members, hour, minute and second, I use a function, normalizeTime which sets the hour value in the range 0 to 23, the minute value in the range 0 to 50 and the second value in the range 0 to 50. In the public, I use a method, writeTime to convert time in standard-time format. And duration method calculate the length of time.

In TimeCard class, I use two Time2 objects, punchInTime, punchOutTime to recored the employee’s checking in time and checking out time. Then in writeSalaryInfo method, I can use the duration method from Time2 class to calculate the work time.

In main program, users can input two employee’s data and get the worked hours and wages.