Assignment

Write a program that will track certain information about the flights of aircrafts. You will create your program by referencing the structure given in the diagram (1). Class definitions, access specifiers, data members and member functions in your program must be coherent with the diagram. Certain essential variables for operations such as loops, flag values, counts etc. are not given in the diagram therefore you can define them as your own extra variables in the program.

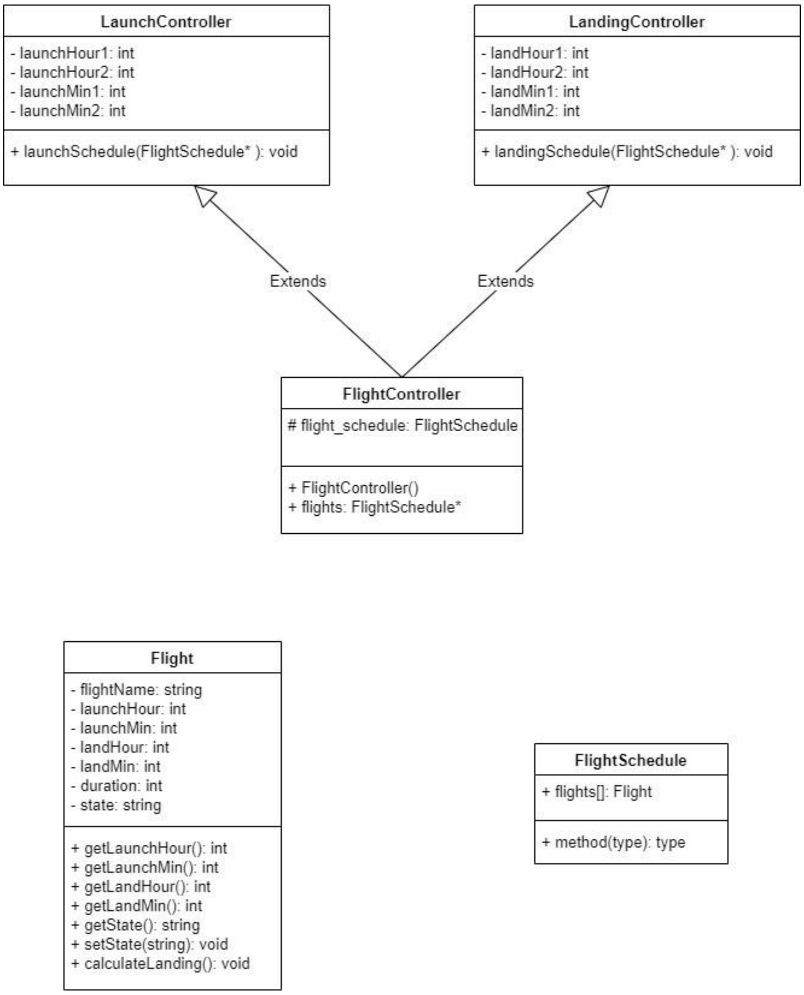


Diagram 1

Your program will have a class called FlightController that is going to be derived from two base classes which are called LaunchController and LandingController. The FlightController class inherits all accessible data fields and methods from the LandingController and LaunchController classes.

Inside the FlightController class, you will define an object of type FlightSchedule which will contain an array of 3 objects of type Flight. In your Flight objects, the constructor of these objects will ask from the user for the inputs of: flight name, launch hour, launch minutes, flight duration. The value of state of the Flight object will be assigned to “Idle” as a default value in the constructor also and you will call the member function within the Flight class which will calculate the landing time for your flight object by using the duration and the launch time. Definition of Flight class will also contain the necessary getters and setters that are given in the diagram and needed throughout your program.

In the constructor of your FlightController you will need to define a pointer that will point to the adress of the FlightSchedule object you have created. You will send that pointer into the functions launchSchedule and landingSchedule in your base classes and these functions will perform a sorting operation between these three flight objects.

The launchController will check the launch times as in hours and minutes of the array of objects that you have passed as a reference and will sort the array in ascending order. After the sorting you will output the launch order of these flights in order. Similar approach will be taken in the landingController and the landing order will be displayed as output after the sorting is complete. Outputs will consist of flight name, landing hour, landing minute, flight’s state and they will be printed from top down in order. Same thing will hold for both of the landing and launch controllers.

Class definitions for both the landingController and the launchController must also contain all of the necessary members and variables in correct access specifiers and expected types. After successfully displaying the ordered ouputs for launches and landing , the program will end.

Important Remarks:

- You can use the help of swap() function while ordering the landings and launches. This function is contained in the header file of #include<algorithm>

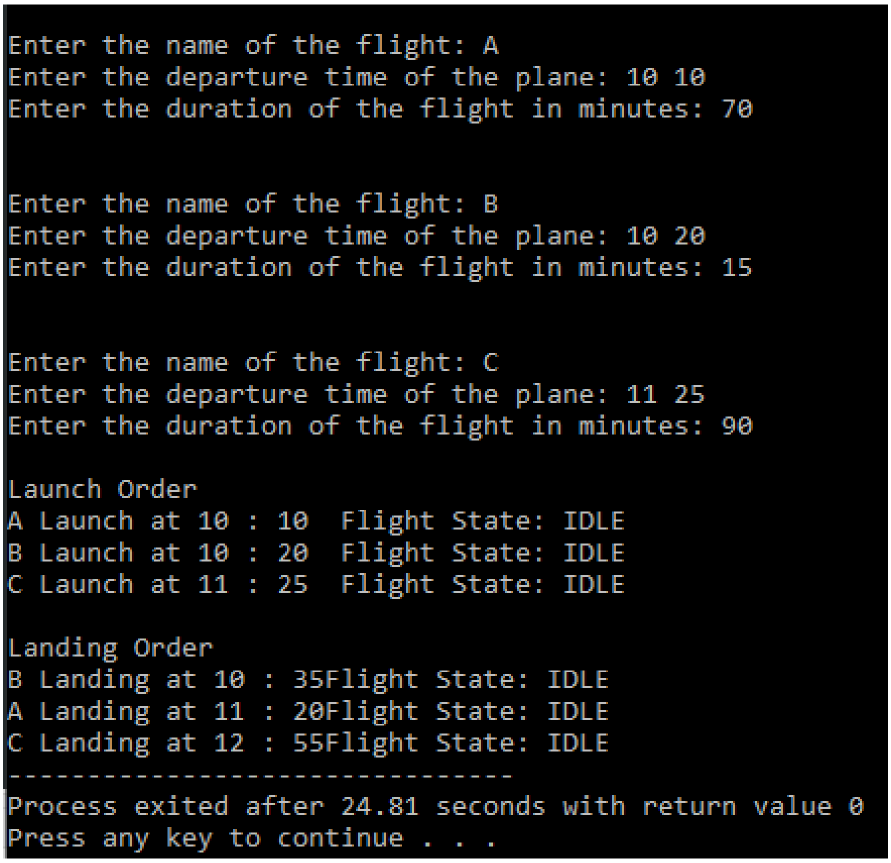
- You can use the help of any sorting algorithms your own sorting algorithms, bubble sort etc. To order the array of objects in the launch and flight controllers. Since these controllers are doing their own outputs within their own functions, you can call them one after another in your FlightController’s constructor and each of them will sort the array in their own needs and output regardingly.

- You are expected to use the help of passing objects as references, this operators, logical conditions, abilities of inheritance in OOP, any necessary extra calculations etc. during your program.

- If there are two or more flights that overlap during the launch or the landing, their states will be changed from the default value of “Idle” to “Delayed”, other values will be printed normally.

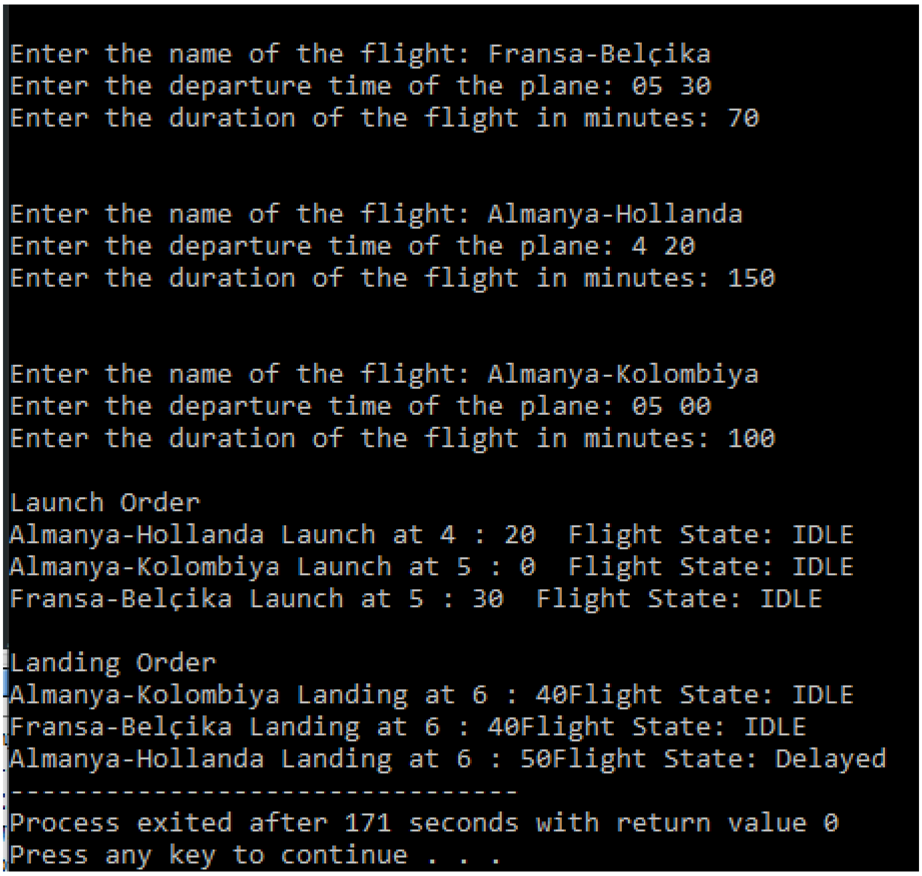
Example Outputs

Output 1



```
Enter the name of the flight: A
Enter the departure time of the plane: 10 10
Enter the duration of the flight in minutes: 70


Enter the name of the flight: B
Enter the departure time of the plane: 10 20
Enter the duration of the flight in minutes: 15


Enter the name of the flight: C
Enter the departure time of the plane: 11 25
Enter the duration of the flight in minutes: 90

Launch Order
A Launch at 10 : 10  Flight State: IDLE
B Launch at 10 : 20  Flight State: IDLE
C Launch at 11 : 25  Flight State: IDLE

Landing Order
B Landing at 10 : 35Flight State: IDLE
A Landing at 11 : 20Flight State: IDLE
C Landing at 12 : 55Flight State: IDLE
--------------------------------
Process exited after 24.81 seconds with return value 0
Press any key to continue . . .
```

Output 2



```

Enter the name of the flight: Fransa-Belçika
Enter the departure time of the plane: 05 30
Enter the duration of the flight in minutes: 70


Enter the name of the flight: Almanya-Hollanda
Enter the departure time of the plane: 4 20
Enter the duration of the flight in minutes: 150


Enter the name of the flight: Almanya-Kolombiya
Enter the departure time of the plane: 05 00
Enter the duration of the flight in minutes: 100

Launch Order
Almanya-Hollanda Launch at 4 : 20  Flight State: IDLE
Almanya-Kolombiya Launch at 5 : 0  Flight State: IDLE
Fransa-Belçika Launch at 5 : 30  Flight State: IDLE

Landing Order
Almanya-Kolombiya Landing at 6 : 40Flight State: IDLE
Fransa-Belçika Landing at 6 : 40Flight State: IDLE
Almanya-Hollanda Landing at 6 : 50Flight State: Delayed
--------------------------------
Process exited after 171 seconds with return value 0
Press any key to continue . . .
```