Documentation on Train Project

Abdulla M Ghosheh, Leah Ten Eyck, Alan Vuong, Stephanie Strain

Department of Science Saddleback College

Computer Science 1A

Prof. Soler

April 17, 2020

Abstract

The code created will simulate a system designed to manage a schedule for a train station located in Orange County and allow users to schedule a trip. The trains have two destinations, Los Angeles and San Diego. The code created will prompt the user with a menu with two options to pick from, either to depart to LA or SD, which the user types in 1 for LA or 2 for SD. When a number has been picked successfully the program will show the departure times schedule to each from Orange County. Orange County has a train departing to Los Angeles every hour from 5 am to 9 pm, and a train departing to San Diego every two hours from 5am to 9pm as well.

After that, the program will ask the user when he/she wants to leave to the chosen city. The user has to input a time between 5-21, which correlates to time interval 5 am to 9 pm, as whole numbers. Moreover, it will take the user one hour to arrive in LA and two hours to arrive in San Diego. For example, I want to head to Los Angeles at 5:30 am. I will enter 6 as asked from the program, then the program will output “The travel time is 1 hour. You will depart at 6:00 and arrive at 7:00...”. Or if I want to head to San Diego at 5:30 the next train will be at 7 thus the program will output “The travel time is 2 hours. You will depart at 7:00 and arrive at 9:00…”. After the program has prompted the program will ask the user if he/she wants to see the schedule again for the chosen city. The user will have to enter ‘y’ for yes and ‘n’ for no. If the user entered ‘n’ the program will ask the user “Would you like to quit?”. The user will have to enter ‘y’ for yes and ‘n’ for no again. If the user entered ‘y’ the program will return 0, or in other

words finish. If the user entered ‘n’ the program will prompt the menu again. The bases of the using the program from the user perspective are:

- The user enters the number correlating to the destination
- The user will be prompted with the departure times
- The user picks a departure time

When the user chooses one location an if statement will execute checking for 1 or 2. 1 is for the departure to Los Angeles and 2 is for the departure to San Diego as stated before. Let’s say the user chose 1 for LA, this will trigger a for loop that increases by 1 for each iteration; that loops inside a *departuresToLA* function which will eventually print out departure times from OC to LA. After this, as stated before, the program will ask the user when would the user like to leave. Then takes the input and outputs the statement “The travel time is X hours. You will depart at ‘*chosen time*’ and arrive at ‘*chosen time* + X’…”

In the case if the user chose 2 for San Diego, this will trigger a for loop that increases by 2 for each iteration; that loops inside a *departuresToSD* function which will eventually print out the departure times from OC to LA. Since the times for departure from OC to San Diego happen every two hours from 5:00 am to 9:00 pm all the time intervals that can be entered are odd numbers. Thus, a vector of integer type has been created and used in a nested for loop to minimize user error. This works by getting the user input of when he/she wants to leave. If the user enters a number which is not in the domain of the schedule the program will print “Invalid input. Please enter a time that is on schedule.”. This work with the nested for loop mentioned checks if the user input one of the even numbers in the list which is in the vector. If it does equal

one of the even numbers the statement above will be printed. This will keep iterating until the user enters a number on the train schedule.

In the end the user will be asked if he/she wants to see the schedule again. This will be executed two if statements at the end. If the user answers with ‘y’ which is yes, the first if statement will check if the user originally picked LA or SD thus outputting the respective schedule for each. Then prints to the user if he/she wants to quit ‘y’ which is yes to quit and ‘n’ which is no thus the program will rerun and show the menu again. If the user answers with ‘n’ to the question pertaining to seeing the schedule the second if statement executes and prints if he/she wants to quit the program answering with ‘y’ and ‘n’.

Bibliography

Class Vector<valuetype>, The Stanford C++ Libraries, www.web.stanford.edu/class/archive/cs/cs106b/cs106b.1126/materials/cppdoc/Vector-class.html. Accessed 12 April 2020

For Loops, Zybooks, ISBN: 978-1-394-02093-5. www.learn.zybooks.com/zybook/SADDLEBACKCS1ASolerSpring2020/chapter/4/section/4. Accessed 8 April 2020

Input/Output Library, Cplusplus, 2000-2020, www.cplusplus.com/reference/iolibrary/. Accessed 15 April 2020

Iterating Through Vectors, Zybooks, ISBN: 978-1-394-02093-5. www.learn.zybooks.com/zybook/SADDLEBACKCS1ASolerSpring2020/chapter/5/section/4. Accessed 8 April 2020

While Loops, Zybooks, ISBN: 978-1-394-02093-5. www.learn.zybooks.com/zybook/SADDLEBACKCS1ASolerSpring2020/chapter/4/section/2. Accessed 1 April 2020