

4.1a Create a `cancelTicket` function in the `MovieTheater` class that takes in a ticket number. If the ticket is not a valid ticket number (e.g. less than the starting ticket number or greater than or equal to the next ticket number), the function will display an error message. If the ticket number is valid, the function will find the ticket in the `soldTickets` vector and set the status of the ticket to “cancelled”.

4.1b Test the `cancelTicket` function in the main program. Try to cancel an invalid ticket number to see that you get the error message. Cancel a valid ticket, then print the ticket information using the `kioskPrintTicket` function to see that the status says “cancelled”.

4.2a Create a `ticketReport` function in the `MovieTheater` class that first prints out all the sold, “active” status ticket information, then prints out all the sold, “cancelled” status ticket information.

4.2b Test the `ticketReport` function in the main program. Make sure you have tickets in both “active” and “cancelled” status before running this function.

Submission: Attach your `MovieTheater.cpp`, `MovieTheater.h` and `main.cpp` files to the Blackboard assignment. Due: Thursday, February 21, 2019 by end of day.