Ticket Pseudocode

**PART 1**

Auditorium Constructor:

For (0 to num rows, increment rows)

For (0 to num columns, increment columns

Create node

Fill it with info (seat, column, reserved or not, type of ticket bought)

Link to next node

Input Validation:

Try-catch

Try{

Get row

If(not int and not in auditorium)

Throw exception 1

Get starting seat

If(not int and not in row)

Throw exception 2

Get num of adult tickets

If(not int or more than size of row)

Throw exception 3

Get num of child tickets

If(not int or more than size of row)

Throw exception 4

Get num of senior tickets

If(not int or more than size of row)

Throw exception 5

If(num total seats more than seats in row)

Throw exception 6

}

Catch(ex 1)

Loop until int and in auditorium

Catch(ex 2)

Loop until int and in row

Catch(ex 3)

Loop until int and less than size of row

Catch(ex 4)

Loop until int and less than size of row

Catch(ex 5)

Loop until int and less than size of row

Catch(ex 6)

Say no seats available bc total tickets is greater than size of row

Best Available:

(code already finds best seats within the row)

If(seats aren’t available in the desired row)

Go to one row below and check to see if seats are available there

Got to one row above and check to see if seats are available there

If(both have available seats)

Compare distance to middle of row on each row and select row that has seats closest to the middle

Else if(row below has available seats)

Select this row

Else if(row above has available seats)

Select this row

* Repeat for all rows until available seats are found
* If not return -1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PART 2**

Main:

Make a hash table and store info from userdb.dat into it

Ask user for username

Ask user for password

loop if not correct, x3

go to main menu if still not found

if (username is “admin”)

admin()

display main menu

if(choose to reserve)

reserve()

if(choose to view order)

viewOrders()

if(choose to update order)

updateOrder()

if(choose to display receipt)

displayReciept()

if(choose to log out)

return to starting point

admin()

parameters: auditorium 1,2, and 3

return type: void

logic:

print out admin menu

if(print report)

print column headers to console

print number of open seats for each auditorium

print total reserved seats for each auditorium

print adult, child and senior seats for each auditorium

print ticket sales for each auditorium

if(logout)

return to starting point

if (exit)

store auditoriums back to files

end program

reserve()

parameters: auditoriums 1,2, and 3

return type: orders

logic:

display submenu

display auditorium that is chosen

use reservation system from project (including input validation using exception handling)

viewOrders()

parameters: orders

return type: void

logic:

for(all of the orders)

display the auditoriums in which they have orders (complete with all orders)

displays the number of tickets and what type they are

displays the seats that are reserved by them

updateOrder()

parameters: orders

return type: void

logic:

display the orders descriptively with numbers

display submenu for action

if(add)

reserve() but not with best available

if(delete)

ask for seats to delete

if valid remove seats from order

if not valid display message and return to submenu

if (cancel)

delete order

update auditoriums

displayReciept()

parameters: orders

return type: void

logic:

for each auditorium

display auditorium, seats and number of tickets per ticket type

display amount per order

display total of all orders

return to main menu