30/30 points for Program 2

Very thorough solution, overall.
Documentation excellent.

```
/*******************************************************************************
 * This program inputs information simulates ordering tickets to attend a minor league
 * ball game.  The user is prompted about the type of tickets and the program will
 * calculate the price
 *
 * Program by Michael Clinesmith
 * CST 183 Programming Assignment 2
 *******************************************************************************/

/*******************************************************************************
 * Notes on error checking:
 * In addition to checking that the number of tickets must be above zero,
 * the number of games must be at least 1, but less than 100,
 * and that the Ticket types must be an L, S, or B,
 *
 * it also will check for data type mismatches, such as user inputting a double for an integer,
 * or hitting the cancel button, or entering unclear formatted strings
 * and give the user a total of MAX_TRIES attempts to enter correct data for each loop
 * before ending the program with an error message
 *
 * Additionally, the program will allow the user to use either upper and lower case
 * and allows the user to enter a variable length string for an answer and accept answers
 * Such as "YES", "Y", "yo", etc for yes, and
 * "LAWN", "l", "Lawn seats for me!", etc for lawn
 *******************************************************************************/

import javax.swing.JOptionPane;

public class TicketOrders
{
    public static void main(String args[])
    {
        // declarations
        final double    LAWN_PRICE = 8.0, SEAT_PRICE = 12.0, BOXSEAT_PRICE = 25.0,  MAIL_PRICE = 3.0,
                        LAWN_SEASON_PRICE = 800.0, SEAT_SEASON_PRICE = 1500.0, BOXSEAT_SEASON_PRICE = 3500.0;
        final double    DISCOUNT_1 = .05, DISCOUNT_2 = .10; // discounts for between 3 and 9 and more than 10 tickets
                                                            // the discount range is hard coded into the program


        final int MAX_TRIES = 3;        //  number of tries after a question to get proper input before ending program
        int inputTries;                 //  keeps track of input tries
        boolean endProgram = false;     //  flag to end program after too many errors
        boolean invalidInput;           //  flag to allow the program to loop to get proper input

        int numberOfTickets=0, numberOfGames=0; //  holds number of tickets and games ordered
        char    seatType='B';                   //  holds 'l' or 'S' or 'B' for seat type
        boolean seasonTickets = false;          //  stores if season tickets ordered
        boolean mailTickets = false;            //  stores if tickets are to be mailed
        String  messageString;                  //  used to create messages
        String inputString;                     //  receives input from user
        String finalOrderString;                //  used to create the final order
        String finalOrderStringFormatted;       //  the formatted final order

        double ticketCost=0.0;                  //  stores the total value of the tickets
        double discount;                        //  stores discount dollar amount
        double mailCost;                        //  stores the cost for mailing the tickets (or not)
        double totalBill;                       //  stores the total calculated bill

        // introductory messages
        //
```

Great use of constants.

```
JOptionPane.showMessageDialog(null, "Welcome to Ticket Order\n" +
                                                    "for your purchasing needs.");



messageString = "This program will prompt you for the following information:\n" +
                "Number of tickets\n" +
                "Type of seats: Lawn, Seat or BoxSeat\n" +
                "If you are buying season tickets\n" +
                "The number of games\n" +
                "If the tickets are to be mailed or picked up at will call\n" +
                "\n" +
                "Please have this information ready.";

JOptionPane.showMessageDialog(null, messageString);

//  get input for number of tickets
//

inputTries = 0;                         // priming to give user MAX_TRIES attempts to attempt input
invalidInput = true;                    // flag to exit loop when valid input is received
                                        // these preface every input loop
                                        // note the endProgram flag in each input loop that when
                                        // set to true because the user has exceeded the max number
                                        // of input attempts will make the program skip all the loops

while (!endProgram && invalidInput)     // loop to give user multiple tries to enter correct input
{

    /*-------------------------------------------------------------------------
     *   The messageString is first created with extra formatting symbols,
     *   then reformatted using String.format so that prices can be displayed
     *   in the proper format
     */

    messageString = "Please enter the number of tickets you wish to purchase:\n" +
                    "(This should be an integer value of at least 1.)\n" +
                    "\n" +
                    "There is a discount of %.1f%% for ordering\n" +
                    "from 3 to 9 tickets and,\n" +
                    "There is a discount of %.1f%% for ordering\n" +
                    "at least 10 tickets.";

                        // multiply discounts by 100 to display percentages
    messageString = String.format(messageString, DISCOUNT_1*100, DISCOUNT_2*100);

    try                 // used to catch bad input data exception
    {
        inputString = JOptionPane.showInputDialog(messageString);
        if (inputString == null)
        {
            JOptionPane.showMessageDialog(null, "Input cancelled", "ERROR",
                    JOptionPane.ERROR_MESSAGE);
            inputTries++;
        }
        else
        {
            numberOfTickets = Integer.parseInt(inputString);
            if (numberOfTickets<=0)
```

Very well-crafted code.

```
            {
                messageString = "Invalid number of tickets entered";
                JOptionPane.showMessageDialog(null, messageString, "ERROR",
                        JOptionPane.ERROR_MESSAGE);
                inputTries++;
            }
            else
            {
                messageString = "You selected " + numberOfTickets + " tickets.";
                JOptionPane.showMessageDialog(null, messageString);
                invalidInput = false;
            }
        }
    }
    catch (NumberFormatException e)     // catches exception where user inputs improperly formatted data
    {
        JOptionPane.showMessageDialog(null, "Invalid input", "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }

    if (inputTries>= MAX_TRIES)         // flag to end loop if too many errors
    {
        endProgram = true;
    }
}

//  get input for type of seats
//

inputTries = 0;
invalidInput = true;
while (!endProgram && invalidInput)     // loop to give user multiple tries to enter correct input
{
    /*-------------------------------------------------------------------------------
     *   The messageString is first created with extra formatting symbols,
     *   then reformatted using String.format so that prices can be displayed
     *   in the proper format
     */

    messageString = "Please enter the type of seats of tickets you wish to purchase:\n" +
                    "Enter L for (L)awn\n" +
                    "Enter S for (S)eat\n" +
                    "Enter B for (B)oxSeat\n" +
                    "\n" +
                    "Current Prices:\n" +
                    "----------------------\n" +
                    "Lawn (1 game): $%,.2f\n" +
                    "Seat (1 game): $%,.2f\n" +
                    "BoxSeat (1 game): $%,.2f\n" +
                    "Lawn (season): $%,.2f\n" +
                    "Seat (season): $%,.2f\n" +
                    "BoxSeat (season): $%,.2f\n";

    messageString = String.format(messageString, LAWN_PRICE, SEAT_PRICE, BOXSEAT_PRICE,
            LAWN_SEASON_PRICE, SEAT_SEASON_PRICE, BOXSEAT_SEASON_PRICE);

    inputString = JOptionPane.showInputDialog(messageString);
    if (inputString == null)
    {
```

```
        JOptionPane.showMessageDialog(null, "Input cancelled", "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else if (inputString.length()==0)
    {
        messageString = "No seat type entered";
        JOptionPane.showMessageDialog(null, messageString, "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else
    {
        inputString = inputString.toUpperCase();        // converts string to uppercase for easier checking
        seatType = inputString.charAt(0);

        switch(seatType)
        {
            case 'L':
                messageString = "You requested Lawn tickets";
                JOptionPane.showMessageDialog(null, messageString);
                invalidInput = false;
                break;
            case 'S':
                messageString = "You requested Seat tickets";
                JOptionPane.showMessageDialog(null, messageString);
                invalidInput = false;
                break;
            case 'B':
                messageString = "You requested BoxSeat tickets";
                JOptionPane.showMessageDialog(null, messageString);
                invalidInput = false;
                break;
            default:
                messageString = "Invalid seat type entered";
                JOptionPane.showMessageDialog(null, messageString, "ERROR",
                        JOptionPane.ERROR_MESSAGE);
                inputTries++;
                break;
        }
    }

    if (inputTries>= MAX_TRIES)         // flag to end loop if too many errors
    {
        endProgram = true;
    }
}

//  get input for season tickets
//

inputTries = 0;
invalidInput = true;
while (!endProgram && invalidInput)     // loop to give user multiple tries to enter correct input
{
    messageString = "Is this purchase for season tickets?\n" +
            "Enter Y for Yes\n" +
            "Enter N for No\n";

    inputString = JOptionPane.showInputDialog(messageString);
```

```
    if (inputString == null)
    {
        JOptionPane.showMessageDialog(null, "Input cancelled", "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else if (inputString.length() == 0)
    {
        messageString = "No answer entered";
        JOptionPane.showMessageDialog(null, messageString, "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else
    {
        inputString = inputString.toUpperCase();        // converts string to uppercase for easier checking
        if (inputString.charAt(0) == 'Y')
        {
            messageString = "You selected season tickets";
            JOptionPane.showMessageDialog(null, messageString);
            seasonTickets = true;
            invalidInput = false;
        }
        else if (inputString.charAt(0) == 'N')
        {
            messageString = "You selected ordering tickets by game";
            JOptionPane.showMessageDialog(null, messageString);
            seasonTickets = false;
            invalidInput = false;
        }
        else
        {
            messageString = "Invalid answer entered";
            JOptionPane.showMessageDialog(null, messageString, "ERROR",
                    JOptionPane.ERROR_MESSAGE);
            inputTries++;
        }

    }
    if (inputTries>= MAX_TRIES)         // flag to end loop if too many errors
    {
        endProgram = true;
    }

}

//  get input for number of games for non season tickets requests
//

inputTries = 0;
invalidInput = true;
while (!endProgram && invalidInput  && !seasonTickets)     // loop to give user multiple tries to enter correct input
{
    messageString = "Please enter the number of games you wish to purchase:\n" +
                    "(This should be an integer value from 1 to 99.)";
    try                 // used to catch bad input data exception
    {
        inputString = JOptionPane.showInputDialog(messageString);
        if (inputString == null)
        {
```

```
            JOptionPane.showMessageDialog(null, "Input cancelled", "ERROR",
                    JOptionPane.ERROR_MESSAGE);
            inputTries++;
        }
        else
        {
            numberOfGames = Integer.parseInt(inputString);
            if (numberOfGames <= 0 || numberOfGames >= 100)
            {
                messageString = "Invalid number of games entered";
                JOptionPane.showMessageDialog(null, messageString, "ERROR",
                        JOptionPane.ERROR_MESSAGE);
                inputTries++;
            }
            else
            {
                messageString = "You selected " + numberOfGames + " games.";
                JOptionPane.showMessageDialog(null, messageString);
                invalidInput = false;
            }

        }
    }
    catch (NumberFormatException e)     // catches exception where user inputs improperly formatted data
    {
        JOptionPane.showMessageDialog(null, "Invalid input", "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }

    if (inputTries>= MAX_TRIES)         // flag to end loop if too many errors
    {
        endProgram = true;
    }

}

//  get input for mailing tickets
//

inputTries = 0;
invalidInput = true;
while (!endProgram && invalidInput)     // loop to give user multiple tries to enter correct input
{
    /*-------------------------------------------------------------------------------
     *   The messageString is first created with extra formatting symbols,
     *   then reformatted using String.format so that prices can be displayed
     *   in the proper format
     */

    messageString = "Do you want the tickets mailed to you?\n" +
            "Otherwise they will be available at will call\n" +
            "at the ball park.\n" +
            "Enter Y for Yes\n" +
            "Enter N for No\n" +
            "\n" +
            "The cost for mailing the tickets is $%.2f\n" +
            "\n" +
            "There is no charge for picking up the tickets\n" +
            "at will call";
```

Very thorough flow through the program. We'll cover methods soon that will allow more of a break down into sub-programs.

```
    messageString = String.format(messageString, MAIL_PRICE);

    inputString = JOptionPane.showInputDialog(messageString);
    if (inputString == null)
    {
        JOptionPane.showMessageDialog(null, "Input cancelled", "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else if (inputString.length() == 0)
    {
        messageString = "No answer entered";
        JOptionPane.showMessageDialog(null, messageString, "ERROR",
                JOptionPane.ERROR_MESSAGE);
        inputTries++;
    }
    else
    {
        inputString = inputString.toUpperCase();        // converts string to uppercase for easier checking
        if (inputString.charAt(0) == 'Y')
        {
            messageString = "You selected to have the tickets mailed to you";
            JOptionPane.showMessageDialog(null, messageString);
            mailTickets = true;
            invalidInput = false;
        }
        else if (inputString.charAt(0) == 'N')
        {
            messageString = "You chose to pick up your tickets at the ball park.";
            JOptionPane.showMessageDialog(null, messageString);
            mailTickets = false;
            invalidInput = false;
        }
        else
        {
            messageString = "Invalid answer entered";
            JOptionPane.showMessageDialog(null, messageString, "ERROR",
                    JOptionPane.ERROR_MESSAGE);
            inputTries++;
        }
    }
    if (inputTries>= MAX_TRIES)         // flag to end loop if too many errors
    {
        endProgram = true;
    }
}

//  finish processing to determine price and prepare order summary
//

if (!endProgram)
{
    totalBill = 0;
    finalOrderString = "Order Summary:\n";

    /*-------------------------------------------------------------------------
     *   The finalOrderString is put together in pieces based on the choices of the user,
     *   then reformatted using String.format so that prices can be displayed
     *   in the proper format
```

```
 */

if(seasonTickets)           // determine cost for season tickets
{
    switch (seatType)
    {
        case 'L':
            finalOrderString += "%d season Lawn tickets ordered:  $%,.2f\n";
            ticketCost = numberOfTickets * LAWN_SEASON_PRICE;
            totalBill += ticketCost;
            break;
        case 'S':
            finalOrderString += "%d season Seat tickets ordered:  $%,.2f\n";
            ticketCost = numberOfTickets * SEAT_SEASON_PRICE;
            totalBill += ticketCost;
            break;
        case 'B':
            finalOrderString += "%d season BoxSeat tickets ordered:  $%,.2f\n";
            ticketCost = numberOfTickets * BOXSEAT_SEASON_PRICE;
            totalBill += ticketCost;
            break;
    }
}
else                        // determine cost of regular tickets
{
    switch (seatType)
    {
        case 'L':
            finalOrderString += "%d Lawn tickets ordered for %d games:  $%,.2f\n";
            ticketCost = numberOfTickets * numberOfGames * LAWN_PRICE;
            totalBill += ticketCost;
            break;
        case 'S':
            finalOrderString += "%d Seat tickets ordered for %d games:  $%,.2f\n";
            ticketCost = numberOfTickets * numberOfGames * SEAT_PRICE;
            totalBill += ticketCost;
            break;
        case 'B':
            finalOrderString += "%d BoxSeat tickets ordered for %d games:  $%,.2f\n";
            ticketCost = numberOfTickets * numberOfGames * BOXSEAT_PRICE;
            totalBill += ticketCost;
            break;
    }
}

if (numberOfTickets<3)          // determine discount
{
    finalOrderString += "No discount for bulk purchase of tickets: $%,.2f\n";
    discount = 0.0;
}
else if (numberOfTickets<10)
{
    finalOrderString += "Discount for purchasing from 3 to 9 tickets: -$%,.2f\n";
    discount = totalBill * DISCOUNT_1;
}
else
{
    finalOrderString += "Discount for purchasing 10 or more tickets: -$%,.2f\n";
    discount = totalBill * DISCOUNT_2;
}
```

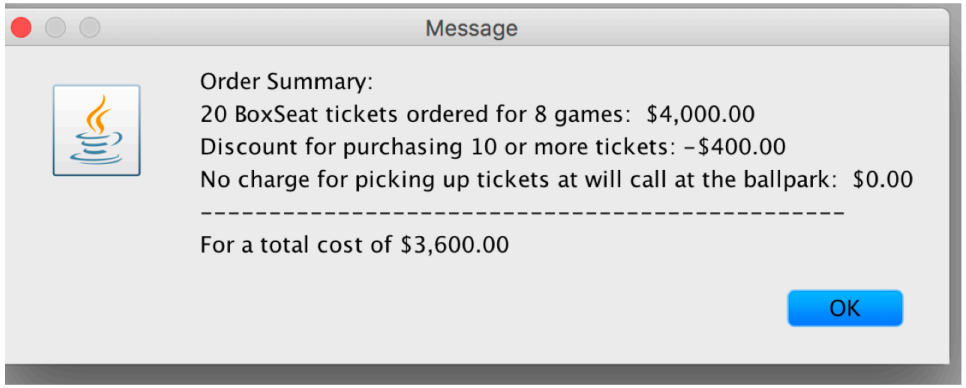
```
            if (mailTickets)            // determine cost of mailing
            {
                finalOrderString += "Cost for mailing tickets:  $%,.2f\n";
                mailCost = MAIL_PRICE;
            }
            else
            {
                finalOrderString += "No charge for picking up tickets at will call at the ballpark:  $%,.2f\n";
                mailCost = 0.0;
            }
            totalBill = totalBill - discount + mailCost;
            finalOrderString += "-----------------------------------------------\n" +
                                "For a total cost of $%,.2f\n";

            // print order
            //

            if (seasonTickets)
            {
                finalOrderStringFormatted = String.format(finalOrderString, numberOfTickets, ticketCost,
                                                          discount, mailCost, totalBill);
            }
            else
            {
                finalOrderStringFormatted = String.format(finalOrderString, numberOfTickets, numberOfGames,
                                                          ticketCost, discount, mailCost, totalBill);
            }
            JOptionPane.showMessageDialog(null, finalOrderStringFormatted);

            messageString = "Thank you for using Ticket Order\n" +
                            "for your ticket purchasing needs!\n";

            JOptionPane.showMessageDialog(null, messageString);

        }
        else                            // This code runs when the user entered too many errors
        {
            messageString = "Program aborted due to input errors";
            JOptionPane.showMessageDialog(null, messageString, "ERROR",
                    JOptionPane.ERROR_MESSAGE);
        }

    }
}
```

CST 183 Program 2 Testing

TEST CASES
1) 5 lawn tickets; 2 games; delivered
   ==> Output: $75.00
2) 20 box tickets; 8 games; pick-up
   ==> Output: $3600
3) 12 regular seat tickets; 15 games; delivered
   ==> Output: $1947
4) 5 season, box tickets; delivered
   ==> Output: $16,628

==> Various error-checking tests added

ALSO CHECKED ...
- Clarity and readability of if-logic
- Minimization of repetitive code
- Structure of code including indentation of program blocks
- Documentation/comments



Output looks great. Nicely formatted report.