CIS 200: Project 2 (30 + 10 points)
Due Friday, February 5th by midnight

Reminder: Programs submitted after the due date/time will be penalized 10% for each day the project is late (not accepted after 3 days, i.e. Midnight, Mon, Feb 8th)

** For all remaining projects, you may use either ***Console-based I/O*** or **GUI-based I/O**. (Sample screenshot of executed program will always be shown as Console-based I/O.) **

Reminder: As the projects become more challenging, it is a good idea to remind all students about the *Academic Integrity Policy* for this course. **ALL projects are intended to be done *individually*. Do NOT share your code with anyone.** If it is not your work, don't submit it as yours! Refer to the policy within the course syllabus which states, ***"If two (or more) students are involved in ANY violation of this policy,* at a minimum, *ALL* students involved receive a *zero*** for the assignment and the offense is officially reported to the *KSU Honor Council*. The second offense results in a failing grade for the course and possible suspension from the university (this decision is made by the *K-State Honor Council*)."

I strongly recommend that you do some 'pre-planning' before you code (i.e. I-P-O, algorithm, etc.) This program contains only a lengthy series of decision making steps. The difficulty lies in the logic of how to sequence these steps.

Assignment Description:
FYI: Loops are NOT required to complete this project, but are part of the extra credit.

It's almost baseball season and the return of the *World Champion Kansas City Royals*! Due to their miraculous win in the World Series last season, the front office has decided to completely redo their current season ticketing software and need your help on a software application for the upcoming season.

Create a file called ***Proj2.java*** then develop a java application that meets the following specifications:

3 plans are available for the 2016 season: ***Full Season plans*** that include tickets to all 81 regular season games; ***Half Season plans*** (40 regular season games); and ***Quarter Season plans*** (20 regular season games). Listed below are the 8 different seating options with the actual prices for 2016:

Seating	**Per Game**
BATS Crown Club Seats	$235
Diamond Club Box	$95
Dugout Box	$51
Dugout Plaza	$40
Field Box	$36
Field Plaza	$28
Outfield Box	$26
Hy-Vee Box	$16

A single parking pass can also be purchased for a discounted rate of $8 *per game* for *Full* Season plans and $9 per game for *Half* and *Quarter* Season plans (regularly $10), but must be purchased for all games of a given package (i.e. 81, 40, or 20 games).

- Develop a program that first displays an opening menu with the 3 ticket plans and allows the user to choose one.
- If the user makes an invalid ***numeric*** selection (e.g. 33), display the error message "*Not a valid Plan selected. Enter 1-3 only*" and simply *exit the program*. (Don't worry about *character* input.)

- **If they choose a valid plan option**, display the 8 seating options available, numbered 1-8.
- Again, **if the user makes an invalid selection on available seating options** (e.g. 33), display the error message "*Not a valid Seating option. 1-8 only*") and *exit the program*. (Don't worry about *char* input.)
- **If the user enters a *valid* Seating option**, ask how many tickets they would like (assume all will be the *same* Seating option) then ask if they would like to purchase parking ('Y' or 'N').
- Tax is charged on the **ticket** portion of the order only. Use a tax rate of ***5.75%*** (declare as a *constant*).
- Calculate and display the Plan and the number of games purchased, the total number and type of tickets requested, followed by the total cost for all tickets, tax, total cost for parking (if parking is requested), followed by a grand total for all ($ with 2 decimals on each). ***See sample runs below***
- User will be required to confirm the order. ***If confirmed***, display that the order was confirmed and the amount charged to the account on file ($ with 2 decimals). **If *not* confirmed**, display the message "*Purchase cancelled. Re-run the application to reselect tickets.*"

Sample run of the program ***with*** parking and the **order *confirmed*** (*use spaces/tabs to get output as close as possible to that shown below / user input is shown below in red*):

```
***Welcome to the World Champion Royals 2016 Season Ticketing Application! ***
                (Application developed by <Your Name>)
                           ----CROWNED K.C.! -----

Please select one of the Season Ticket Plans listed below:
        1)  Full Season Plan (tickets to all 81 regular season games!)
        2)  Half Season Plan (tickets to 40 regular season games)
        3)  Quarter Season Plan (tickets to 20 regular season games)

        Selection: 1

Please select one of the Seating Options listed below:
              Seating                   Per Game
    1)  BATS Crown Club Seats          $235
    2)  Diamond Club Box                $95
    3)  Dugout Box                      $51
    4)  Dugout Plaza                    $40
    5)  Field Box                       $36
    6)  Field Plaza                     $28
    7)  Outfield Box                    $26
    8)  Hy-Vee Box                      $16

    Selection: 3

    How many seats would you like to purchase? 2

    A single parking pass is available for purchase at a discounted rate of $8 per game (regularly $10).
    (You will be charge for all games of a given package.)
    Would you like to include parking? (Y or N): Y
```

```
You purchased the Full Season 81-game plan / 2 Dugout Box tickets with Parking
    Ticket Total: $8,262.00
    Tax: $475.06
    Parking: $648.00
    Grand Total: $9,385.06

Confirm Order (Y or N)? Y

Order was confirmed. $9,385.06 will be charged to the account on file.
```
(Not concerned if you are off my $0.01…)

Sample run of the program with a Half Season plan, 2 Dugout Box ***without*** parking and order ***NOT*** confirmed: ….(*See above – user selects option 2 then option 3)*

```
How many seats would you like to purchase? 2

A single parking pass is available for purchase at a discounted rate of $9 per game (regularly $10).
(You will be charge for all games of a given package.)
Would you like to include parking? (Y or N): N

You purchased the HALF Season 40-game plan / 2 Dugout Box tickets without Parking
    Ticket Total: $4,080.00
    Tax: $234.60
    Grand Total: $4,314.60

Confirm Order (Y or N)? N

Purchase cancelled. Re-run the application to reselect tickets.
```

OPTIONAL Extra Credit: (Up to 10 points – partial credit is available)

Important: **To be considered for FULL extra credit**, your project must be submitted **before the deadline** and the project must work ***EXACTLY*** as stated (i.e., *be an A-level program earning at least 27/30 of the non-extra credit points*). Otherwise, a maximum of ½ credit (i.e. 5 pts) is available for the extra credit.

Please update your documentation heading at the top of the program to let the GTA know that you completed the extra credit. If only submitting the project for ***partial*** *extra credit*, include in your documentation heading what works and what doesn't work, to help the GTA easily assess the amount of extra credit to be awarded.

Add the following error checking to your application (***do NOT write a separate application***):

1) If the user enters an invalid number for *EITHER* menu, display an error message and **LOOP** until a valid number is entered before continuing in your program. (3 pts.)
2) If the user does *NOT* enter a 'Y' or 'N' (upper or lower case) to the parking question, display an error message and **LOOP** until a valid answer (Y or N) is entered before continuing in your program. (3 pts.)
3) Add a Loop to ask if they would like to process another order. If so, display the opening menu again and repeat the program until they answer that they do not want to process another order. (Each order is separate) (4 pts.)

Requirements

This project will contain ONE file (called *Proj2*) containing a single class and a main method. Your program must compile (by command-line) with the statement: **javac Proj2.java**

It must then run with the command: **java Proj2**

Documentation:

At the top of *EACH* class, add the following comment block, filling in the needed information:

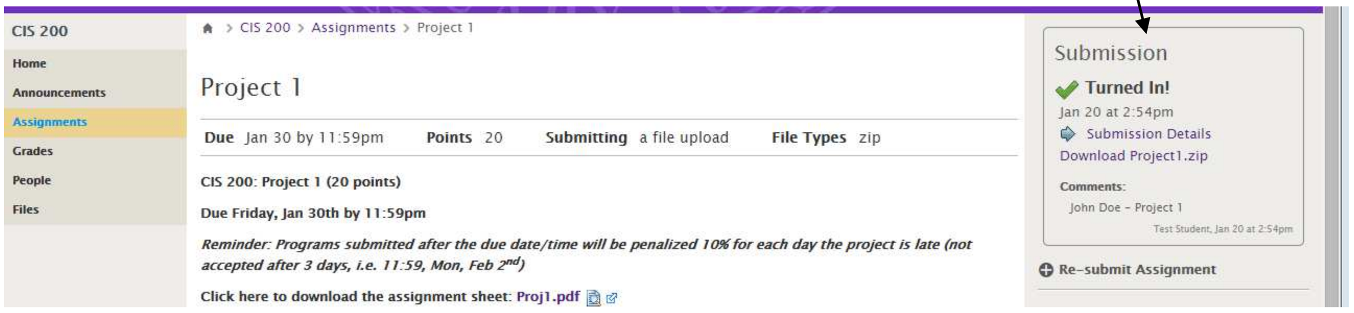
```
/**
 * <Full Filename>
 * <Student Name / Section Day/Time>
 *
 * <description of the project – i.e. What does the program do?>
*/
```

Submission – *read these instructions carefully or you may lose points*

To submit your project, first create a folder called ***proj2*** and copy or move your *.java* file (**Proj2.java)** into that folder. Then, right-click on that folder and select "***Send To → Compressed (zipped) folder***". This will create the file ***proj2.zip***

Log-in to Canvas and upload your ***proj2.zip*** file. **Only a .zip file will be accepted for this assignment in Canvas. Put your full name and Project 1 in the comments box.**

Important: It is the ***student's responsibility*** to verify that the ***correct*** file is ***properly*** submitted. If you don't properly submit the *correct* file, it will not be accepted after the 3-day late period. No exceptions.



Project 1

Due Jan 30 by 11:59pm Points 20 Submitting a file upload File Types zip

CIS 200: Project 1 (20 points)

Due Friday, Jan 30th by 11:59pm

Reminder: Programs submitted after the due date/time will be penalized 10% for each day the project is late (not accepted after 3 days, i.e. 11:59, Mon, Feb 2nd)

Click here to download the assignment sheet: Proj1.pdf

Jan 20 at 2:54pm

Comments:
John Doe - Project 1
Test Student, Jan 20 at 2:54pm

Grading:

Programs that do not compile will receive a grade of 0, so make sure you submit the *correct* file that *properly compiles*. Programs that *do* compile will be graded according to the following rubric:

Make sure and test your solution with ALL possibilities – the GTAs will use multiple scenarios to test the correctness of your output

Requirement	**Points**
Proj2	**-30-**
Specified Documentation included / Tax rate declared as a constant	**2**
Initial menu displayed correct (3 lines at the top followed by 3 Season Ticket Plans)	**2**
INVALID number on initial menu – program properly exits with correct error message	**2**
VALID number on initial menu – displays second menu correctly	**2**
INVALID number on second menu – program properly exits with correct error message	**2**
VALID number on second menu – prompts for number of seats	**1**
Correctly asks user if they want parking	**1**
Output correct if parking is ***NOT*** chosen	**6**
Output correct if parking ***IS*** chosen	**6**
Correct message if order ***IS*** confirmed	**2**
Correct message if order is ***NOT*** confirmed	**2**
Project Submission (zip file with .java file, submitted to correct folder with *Name* and *Project 2* in the description box)	**2**
OPTIONAL Extra Credit (Must score at least 27/30 on the project to be considered for full credit else ½ credit for extra credit)	**+10**
Properly error check Menu Number on EITHER menu – Loops instead of Exits	**3**
Properly error check *Parking Request*	**3**
Process Orders until user chooses to quit	**4**
Minus Late Penalty (10% per day)	
Total	**30 + 10**