# Assignment#3

**Description**

This lab is based on the material studied in weeks 1 to 6, including the exercises done in class.

This assignment is to be done individually; you are not allowed to work on this assignment with anyone.

An underground parking garage requires a program that allows clerks to calculate and display a bill for customer parking charges. A customer is charged when they leave the garage, and the amount charged is based on a few different factors. Some customers own a special membership card if they work in the building above the garage, so they are charged a special hourly rate of only $2 per hour. Customers without the card are charged $3 per hour, unless they park for more than 12 hours, in which case they are charged only $2.50 per hour. Partial hours are rounded up to the next hour, so if a customer parks for 5.2 hours, the charge is calculated for 6 hours. HINT: chargeable hours = Math.ceil(actual hours). The maximum charge allowed is $100, so if a customer's total charges are more than $100, they will be charged exactly $100.

Your program should use an input dialog to retrieve the number of hours parked. (**Bonus Marks:** If the user presses Cancel or enters nothing and clicks OK, your program should display an error message such as "Invalid data entered. Program terminating.")

Your program should use a confirm dialog to ask the user whether or not the customer has a special membership card.

All literal numeric values in your program must be defined as constants. Be sure to use well-defined and appropriate names for all of your constants.

Here is an IPO chart to help you understand the program's requirements:

|  |  |
| --- | --- |
| **Inputs:** | - number of hours parked - whether or not the customer has a membership card |
| **Processing:** | - determine hourly rate based on membership status and hours parked - chargeable hours = ceiling of actual hours parked - total charges = hourly rate \* chargeable hours |
| **Outputs:** | - hourly rate - chargeable hours - total charges |

Your program should also include the following methods:

* **getHourlyRate()** -- figures out the hourly rate based on the customer's membership status and hours parked. If a customer has a membership card, they are only charged $2 per hour. Customers without a membership card are charged $3 per hour, unless they park for more than 12 hours, in which case they are charged $2.50 per hour.
* Method Inputs:
  + an integer for membership status (comes from the confirm dialog -- yes or no button)
  + the chargeable hours (the ceiling of actual hours parked)
* The method returns a double value, the hourly rate that applies to the situation described above.
* **getCharges()** -- calculates and returns the total amount to charge the customer. Charges are calculated as the chargeable hours multiplied by the hourly rate. If the charges are more than $100, then a flat amount of $100 is charged.
* Method Inputs:
  + hourly rate
  + chargeable hours
* The method returns a double value, the amount of the charges.

Output of your program should be formatted appropriately and should appear as follows **in a message dialog** (values shown are examples only):

**Hourly Rate: $3.00**

**Number of Hours Charged: 6.0**

**Total Charge: $18.00**

**Evaluation**

Your submission will be evaluated based on the following criteria:

**Efficient Code:** Program uses variables where and only when necessary; program doesn't define variables that are never used, nor does it use too many variables for unnecessary tasks; program logic is written concisely and is not cluttered with unnecessary tasks. Methods are written concisely and using the standards and techniques discussed in class.

**Functionality:** Program functions according to specifications.

**Programming Style:** Proper indentation and spacing, use of comments/documentation; all identifiers (variables, methods, constants, class) are descriptive and valid; varibles are defined with appropriate types and converted when required. Method names follow the rules and standards discussed in class and have been given self-documenting names.

**Other:** All instructions regarding submissions and program specifications have been followed; submission was completed and submitted as requested in a timely fashion; techniques discussed in class have been used.