Purpose: Using one and two-dimensional arrays.

Topic: Morgan’s Department Store would like an employee bonus calculator. Bonuses to employees are based on their monthly paycheck amount, and whether they are full time or part time. Management would like a system that allows them to enter and display employee information, including the amount of bonus the employee earned.

* Create a console application that will use arrays to store and retrieve information about employees, and to reference the bonus rates to be used in calculating employee bonuses. Limit system to a maximum of 10 employees.
* Use a menu-type approach for user options.
* Create a two-dimensional String array (10 rows by 3 columns) that will be used to store employee first name, last name and status.
* Create a one-dimensional double array (10 rows) that will be used to store the employee paycheck amount.
* Option 1 - data entry:
  + Prompt user for first name and last name, separately. Store the first and last name in the first two columns of the two dimensional string array. Use a loop for each prompt to ensure the entry is not null.
  + Prompt user for paycheck amount. Store amount into the one dimensional double array. Use a loop to ensure the entry is numeric and a range of .01 through 9999.99.
  + Prompt user for full time or part time status (FT or PT). Use a loop to ensure the entry is valid. Store the status in the third column of the two-dimensional array.
  + After an employee is entered, prompt user to enter another employee, or return to Menu
  + Be sure to not allow user to enter more than ten employees.
* Option 2 – display all employees:
  + Be sure to handle the situation where no employees have been entered yet.
  + Display a report to the console in column format showing the employee name, paycheck amount, status and bonus earned. Display employee name in last name, first name order (example: Smith, Joe). Display paycheck and bonus amounts in formatted decimal format (example: 12,000.57). Display status as “Full Time” or “Part Time”. Follow standard alignment based on type of data (strings left justified and numeric right justified). Employee bonus amount is determined by searching the bonus rate chart below and performing a calculation.
  + At the bottom of the report, display the total amount of bonuses paid, aligned with the decimal point under the bonus column.
  + Return to Menu
* Option 3 – display a single employee
  + Be sure to handle the situation where no employees have been entered yet.
  + Allow user to enter the employee’s first and last name, then search the two-dimensional string array for a matching employee. If found, display the employee name, status, paycheck amount and bonus amount in vertical format, with appropriate labels identifying each field.
  + If not found, display an error message.
  + Return to Menu
* Option 4 – exit program
  + Show a message to user that they’ve chosen to end the application.
* The bonus rates are stored in an external data file and are to be read in and stored in a two-dimensional double array at the start of the program. **Be sure this file is located in the project directory.**
* Limit the use of static (class) variables when possible. Practice the use of passing and returning values to/from your methods where feasible.
* The arrays can be defined as static global variables since they may need accessed in multiple methods.
* Be sure to modularize your program. Consider separate methods for options 1, 2 and 3, input/validation of each field, populating the array of bonus rates from the data file, and initializing accumulators.
* Example calculation: employee is full-time with a paycheck of $450.00. Based on the chart, the full-time bonus rate is 6% (.06). The bonus would be 450.00 \* .06 = 27.00.

Bonus rate chart (need to create a .dat file – Do Not put column headings in your .dat file):



Sample Menu screen:

Option 1: Enter Employee Data

Option 2: Display All Employees

Option 3: Display Single Employee

Option 4: Exit Application

Enter your option: X

**Project is worth 50 points.**