***CSC 3020***

***Java Programming***

**Project 01**

**40 points**

**Due 11/15/2023 (11:45 A.M.)**

**The goal of this Project is:**

1. Being able to Analyze, Design, implement, and test a practical real-world application.
2. Being able to use selection and repetition structures.
3. Being able to deal with methods.
4. Being able to manipulate strings.
5. Being able to manipulate 2D arrays.

**Requirements:**

* Using java IDE software, write a java program for each problem.
* Test each program and verify that the program works.

**Restrictions:**

You must work individually. Use only material from class or from the textbook (**chapters 1- 8**). All code must be the work of the individual. Do not share your code or copy from external resources.

**Submission**

For problem 1, name the class MorseCode. For problem 2, name the class SwimSchedules.

Convert .java files to text files. Upload two text files to the Canvas by the due date. DO NOT Email your files.

All assignments must be submitted by the Canvas. **No email or hard copy** is accepted. You must follow the following format:

1. Convert the source file to text file.
2. Submit your file to the Canvas. You must submit your assignment on time; otherwise, you will receive zero. In addition, you cannot submit your file more than one time.
3. There will be several folders on the Canvas. You need to upload your file(s) using the correct folder on the Canvas.
4. To upload your file(s):

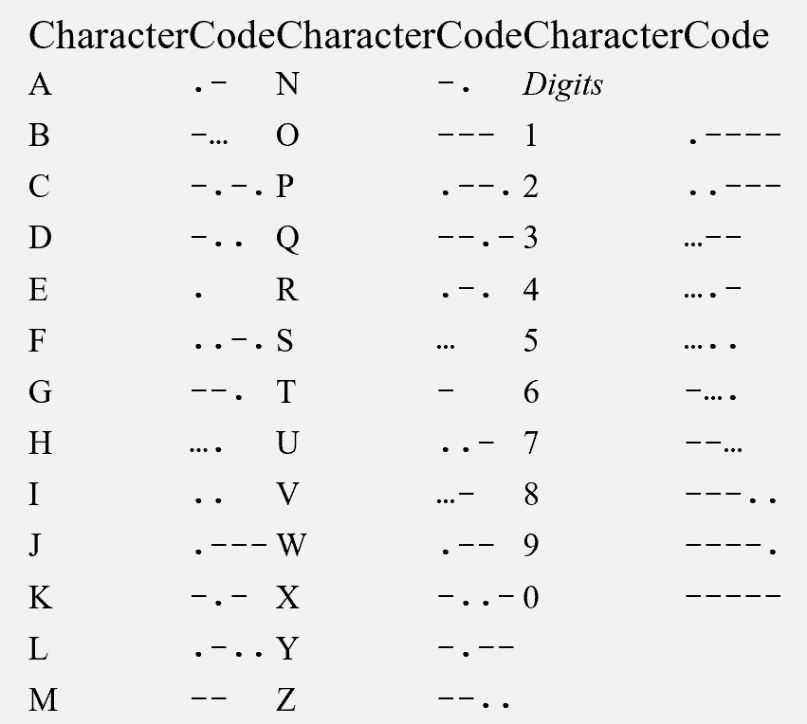
* In Course Navigation, click the Assignments link.
* Click the title of the assignment.
* Click the **Submit** Assignment button.
* Add **File**. ...
* Add Another **File**. ...
* **Submit** Assignment. ...
* View **Submission**.

**It is your responsibility to make sure that each file is uploaded correctly. If you uploaded a wrong file, you receive zero; files will not be accepted after due date even if you have a prove that the file is created before the due date.**

**Make sure you review the Cheating & Plagiarism policy on Canvas.**

**Question 01 - *Morse Code* (20 points)**

Perhaps the most famous of all coding schemes is the Morse code, developed by Samuel Morse in 1832 for use with the telegraph system. The Morse code assigns a series of dots and dashes to each letter of the alphabet and each digit. The international version of the Morse code appears in the below figure.



Write a menu driven program that reads an English-language phrase and encodes it into Morse code. Also, the program reads a phrase in Morse code and converts it into the English-language equivalent. Use ***one blank between each Morse-coded letter*** and ***three blanks between each Morse-coded word***.

In addition to the main method, your program must have at least a ***method for the menu****,* ***a method to convert normal text to Morse code*** and ***a method to decode Morse code letter***.

The program’s menu prints three options: t for encoding text, m for decoding morse code, or e to exit a program. The program should process unlimited number of user requests and validate user input.

*Question 01 Grading Criteria:*

* A function to print a menu (2 points).
* A function to convert normal text to Morse code (6 points).
* A function to convert Morse code to English-language (6 points).
* A function for the main part of the program (4 points).
* The program should validate user input (2 points).

|  |
| --- |
| **Question 01 Sample run**  Hello, this program allows you to translate text to morse code or translate morse code to text.  Please, select one of the below options:  \*\*\* Enter 't' for encoding text  \*\*\* Enter 'm' for decoding morse code  \*\*\* Enter 'e' to exit the program.  x  \*\*\*invalid option\*\*\*  Please enter a valid option:  y  \*\*\*invalid option\*\*\*  Please enter a valid option:  t  Please enter a phrase:  to be or not to be  MorseCode:  - --- -... . --- .-. -. --- - - --- -... .  Hello, this program allows you to translate text to morse code or translate morse code to text.  Please, select one of the below options:  \*\*\* Enter 't' for encoding text  \*\*\* Enter 'm' for decoding morse code  \*\*\* Enter 'e' to exit the program.  m  Please enter a Morse code:  - --- -... . --- .-. -. --- - - --- -... .  Normal Text:  TO BE OR NOT TO BE  Hello, this program allows you to translate text to morse code or translate morse code to text.  Please, select one of the below options:  \*\*\* Enter 't' for encoding text  \*\*\* Enter 'm' for decoding morse code  \*\*\* Enter 'e' to exit the program.  t  Please enter a phrase:  TO be OR not TO be  MorseCode:  - --- -... . --- .-. -. --- - - --- -... .  Hello, this program allows you to translate text to morse code or translate morse code to text.  Please, select one of the below options:  \*\*\* Enter 't' for encoding text  \*\*\* Enter 'm' for decoding morse code  \*\*\* Enter 'e' to exit the program.  m  Please enter a Morse code:  - --- -... . --- .-. -. --- - - --- -... .  Normal Text:  TO BE OR NOT TO BE  Hello, this program allows you to translate text to morse code or translate morse code to text.  Please, select one of the below options:  \*\*\* Enter 't' for encoding text  \*\*\* Enter 'm' for decoding morse code  \*\*\* Enter 'e' to exit the program.  e  Thanks for using this program! |

**Question 02 - Swim Scheduling (20 points)**

A swim school has two swimming instructors, Jeff and Anna. Their current schedules are shown below. An “X” denotes a 1-hour time slot that is occupied with a lesson.

A screenshot of a computer program

Description automatically generated

Write a program with array(s) capable of storing the schedules. Create a main menu that allows the user to mark a time slot as busy or free for either instructor. Also, add an option to output the schedules to the screen. Next, add an option to output all time slots available for individual lessons (slots when at least one instructor is free). Finally, add an option to output all time slots available for group lessons (when both instructors are free).

The program should process unlimited number of user requests and validate user input.

Your program must use two 2-D arrays for the schedules. In addition, you must use at least **four methods** to:

1. Prints a menu to the screen.
2. Prints Anna, Jeff, individual lessons, and group lessons schedules to the screen.
3. Prompts the user to select an instructor, day, and slot.
4. Schedule or free one of the instructor's slots.

*Question 02 Grading Criteria:*

* A method to print a menu (2 points).
* A method to print schedules (4 points).
* Input method to select instructor, day, and slot (4 points).
* A method to schedule or to free instructor’s slot (6 points).
* Validating user input (4 points).

|  |
| --- |
| **Question 02 Sample run**  Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: p  Jeff:  Mon Tue Wed Thu  11-12 X X \_ \_  12-1 \_ X X X  1-2 \_ X X \_  2-3 X X X \_  Anna:  Mon Tue Wed Thu  11-12 X X \_ X  12-1 \_ X \_ X  1-2 X X \_ \_  2-3 X \_ X X  Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: s  Select instructor (1 - Jeff, 2 - Anna): 1  Select Day (1 - Mon, 2 - Tue, 3 - Wed, 4 - Thu): 4  Select Slot (1 - 11-12, 2 - 12-1, 3 - 1-2, 4 - 2-3): 1  Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: p  Jeff:  Mon Tue Wed Thu  11-12 X X \_ X  12-1 \_ X X X  1-2 \_ X X \_  2-3 X X X \_  Anna:  Mon Tue Wed Thu  11-12 X X \_ X  12-1 \_ X \_ X  1-2 X X \_ \_  2-3 X \_ X X    Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: f  Select instructor (1 - Jeff, 2 - Anna): 2  Select Day (1 - Mon, 2 - Tue, 3 - Wed, 4 - Thu): 1  Select Slot (1 - 11-12, 2 - 12-1, 3 - 1-2, 4 - 2-3): 1  Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: p  Jeff:  Mon Tue Wed Thu  11-12 X X \_ X  12-1 \_ X X X  1-2 \_ X X \_  2-3 X X X \_  Anna:  Mon Tue Wed Thu  11-12 \_ X \_ X  12-1 \_ X \_ X  1-2 X X \_ \_  2-3 X \_ X X    Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: i  Slots marked with an 'I' are available for individual lessons:  Mon Tue Wed Thu  11-12 I \_ I \_  12-1 I \_ I \_  1-2 I \_ I I  2-3 \_ I \_ I  Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: g  Slots marked with a 'G' are available for individual lessons:  Mon Tue Wed Thu  11-12 \_ \_ G \_  12-1 G \_ \_ \_  1-2 \_ \_ \_ G  2-3 \_ \_ \_ \_    Enter one of the following commands:  p - Print schedules  s - Schedule a slot  f - Free a slot  i - Show slots available for individual lessons  g - Show slots available for group lessons  q - Quit  Command: q |

**Extra Credit - Coffee Shop (20 points)**

***Points earned in this part can be added to exams or assignments.***

Jason opened a coffee shop at the beach and sells coffee in three sizes: small (9oz), medium (12oz), and large (15oz). The cost of one small cup is $1.75, one medium cup is $1.90, and one large cup is $2.00.

Write a menu-driven program that will make the coffee shop operational. Your program should allow the user to do the following:

* Buy coffee in any size and in any number of cups.
* At any time show the total number of cups of each size sold.
* At any time show the total amount of coffee sold.
* At any time show the total money made.

Your program should consist of at least the following functions:

1. a function to print the menu
2. a function to order the coffee (option 1)
3. a function to check the total money made up (option 2),
4. a function to check the total amount of coffee sold up (option 3)
5. a function to check the number of cups of coffee of each size sold (option 4)
6. a function that print the data (option 5).

Special values such as coffee cup sizes and cost of a coffee cup must be declared as named constants.

***Extra Credit Grading Criteria:***

* *2 points for each of the functions a to f excluding part b (total 10 points)*
* *5 points Main function with Menu driven*
* *5 points for part b (order the coffee)*

|  |
| --- |
| **Extra Credit Sample output**  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **1**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **1**  Enter the number of cups: **2**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **9**  Please pay $3.5  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **1**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **3**  Enter the number of cups: **1**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **2**  Enter the number of cups: **3**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **9**  Please pay $7.70  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **2**  Total money made: $11.2  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **3**  Total amount of coffee sold: 69 oz  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **4**  Small cup count: 2  Medium cup count: 3  Large cup count: 1  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **5**  Small cup count: 2  Medium cup count: 3  Large cup count: 1  Total amount of coffee sold: 69 oz  Total money made: $11.2  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **1**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **1**  Enter the number of cups: **1**  1: Enter 1 to buy coffee in a small cup size (9 oz)  2: Enter 2 to buy coffee in a medium cup size (12 oz)  3: Enter 3 to buy coffee in a large cup size (15 oz)  9: Enter 9 to exit.  **9**  Please pay $1.75  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **5**  Small cup count: 3  Medium cup count: 3  Large cup count: 1  Total amount of coffee sold: 78 oz  Total money made: $12.95  1: Enter 1 to order coffee.  2: Enter 2 to check the total money made up to this time.  3: Enter 3 to check the total amount of coffee sold up to this time.  4: Enter 4 to check the number of cups of coffee of each size sold.  5: Enter 5 to print the data.  9: Enter 9 to exit the program.  **9** |