Faculty of Engineering and Technology
Computer Engineering Department

Object-Oriented Problem Solving Lab
Fall 2020 - Midterm Exam
Eng. Asma Abdel Karim

❖ **Write a Java program that implements the following classes:**

1. **The following class *Menu* which represents a menu (قائمة طعام) in a restaurant:**

Menu
- meals: String[] - prices: double [] - timesOrdered: int [] - numOfMenusCreated: int
+ Menu( ) + Menu(m: Menu) + getMeals ():String[] + setPrice(p:double, index:int): void + getPrices():double[] + incrementTimesOrdered(int index): void + getTimesOrdered(index:int): int + getNumOfMenusCreated (): int + suggestAMeal(): void + getMostOrderedMeal(): String + ptintMenu(): void

   a. **The no-arg constructor must:**
      - Read the number of meals from the user.
      - Initialize the arrays *meals*, *prices*, and *timesOrdered* by creating them with the entered size.
      - Read the *meals* (names) and *prices* from the user.
      - The *timesOrdered* array elements should be initialized to 0.
      - Increment the data field *numOfMenusCreated*.

   b. **The constructor *Menu(m:Menu)* must:**
      - Initialize the arrays *meals*, *prices*, and *timesOrdered* by creating them with the same size as the arrays *meals*, *prices*, and *timesOrdered* of the passed Menu object.
      - Copy the *meals* (names) and *prices* from the *meals* and *prices* arrays of the passed object.
      - The *timesOrdered* array elements should be initialized to 0.
      - Increment the data field *numOfMenusCreated*.

   c. **The method *setPrice* must set the element whose index is passed, of the array *prices*, with the passed double value *p*. The method should only allow the price to be modified if the value of *p* is positive and nonzero.**

   d. **The method *getPrices* must return the prices, but not by returning the reference of the array *prices*, it must return a reference to a copy of the *prices* array.**

e. **The method *incrementTimesOrdered* must increment the element in the array *timesOrdered* with the passed index.**

f. **The method *suggestAMeal* must randomly select one of the meals, and print its name from the *meals* array and its price from the *prices* array. (for example: Chicken Burger – 3.5)**

g. **The method *getMostOrderedMeal* must return the name of the meal which was most ordered based on the *timesOrdered* array.**

h. **The method *printMenu* must print the meals index-name-price each on a new line, then print the name of the most ordered meal by invoking the *getMostOrderedMeal* method. An example output:**

0 – Chicken Burger – 3.5
1 –Vegetarian Pizza – 6
.
.
Most ordered meal: ….

2. In your main class:

a. **Define the method *addElement* which takes an array of integers and an integer, and returns a new array that is formed by adding the integer to the end of the passed array.**

b. **Define the void method *makeAnOrder* which takes an object of type *Menu* and performs the following:**
   - **Print the passed menu by invoking the *printMenu* method for the passed object.**
   - **Ask the user to enter the index of the meal he wants to order, and then create an array of integers that consists initially of the index he entered. This array will include the indices of the meals the user will order.**
   - **The method must then ask the user if he wants to add more meals. If he wants to add more meals, it must ask him to enter the index of the meal and add the index to the array of orders by invoking the *addElement* method. The method should keep on asking the user if he wants more meals and reading the indices of the meal until he enters that he does not want to add more meals.**
   - **The method must compute the total payment of the order by summing the prices of the ordered meal.**
   - **The method must invoke the *incrementNumOrdered* method for each ordered meal using its index.**
   - **At the end of the method, the order should be displayed by printing the ordered meals name-price each on a line, and print the total payment amount on the last line.**

c. **Define the method *isMealOrdered* which takes an array of integers (that represents an order of meal indices) and an integer (that represents a meal index). The method should return true if the meal is found in the order and false otherwise. This can be done by searching for the integer in the array of integers. Note that the integers in the passed array are not sorted. You must use suitable methods of the Arrays class.**

d. In your main method, perform the following in order:

- **Print the number of menus created.**
- **Create an object of type *Menu* named menu1 using the no-arg constructor.**
- **Create an object of type *Menu* named *menu2* using the second constructor and pass *menu1* as an argument.**
- **Iterate over the prices of *menu2* and change them to random double values in the range [0.5,10). (0.5 included, 10 excluded).**
- **Invoke the method *makeAnOrder* twice by passing *menu1* in the first time and *menu2* in the second.**

GOOD LUCK ☺