Mehrnaz zhian

Seneca College

Feb 04, 2020

Applied Arts & Technology

SCHOOL OF COMPUTER STUDIES

JAC444 **Demo Due date: Feb 07, 2020**

Final Code Submission Date: Feb 08, 2020

Workshop 4

Notes:

i. One task should be demo during the class.
ii. In order to get the full mark, all the tasks should be submitted through the blackboard.
iii. Make sure you have all security and check measures in place, like wrong data types etc., implement the proper Exception Handling in any tasks needed.
iv. Make your project in proper hierarchy; introduce proper class coherence in your project. Proper packages and **any tasks that uses inheritance, should be handled by only one main method which should be in a TesterClass**.
v. Given output structure is just for student to have a glimpse what the output can look, students are free to make the output better in any way.

Other inputs can be given during demo, so make sure you test your program properly

1. Write a program that meets the following requirements:

   Creates an array with 100 randomly chosen integers.
   Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds.

2. Design a class named Person and its two subclasses named Student and Employee. Make Faculty and Staff subclasses of Employee. A person has a name, address, phone number, and email address. A student has a class status (freshman, sophomore, junior, or senior). Define the status as a constant. An employee has an office, salary, and date hired. Create and use MyDate class as defined below to create an object for date hired. A faculty member has office hours and a rank. A staff member has a title. Override the toString method in each class to display the class name and the person's name.

   Write a test program that creates a Person, Student, Employee, Faculty, and Staff, and invokes their toString() methods.

*** Here is what MyDate class contains of:

- The data fields year, month, and day that represent a date. month is 0-based, i.e., 0 is for January.
- A no-arg constructor that creates a MyDate object for the current date.
- A constructor that constructs a MyDate object with a specified elapsed time

- since midnight, January 1, 1970, in milliseconds.
- A constructor that constructs a MyDate object with the specified year,
- month, and day.
- Three getter methods for the data fields year, month, and day, respectively.
- A method named setDate(long elapsedTime) that sets a new date for
- the object using the elapsed time.

3. Write a nested for loop that prints the following output:

```
                    1
                  1   2   1
                1   2   4   2   1
              1   2   4   8   4   2   1
            1   2   4   8  16   8   4   2   1
          1   2   4   8  16  32  16   8   4   2   1
        1   2   4   8  16  32  64  32  16   8   4   2   1
      1   2   4   8  16  32  64 128  64  32  16   8   4   2   1
```

4. Write a program to show the following table:

```
            Multiplication Table
        1   2   3   4   5   6   7   8   9
-----------------------------------------
1 |     1   2   3   4   5   6   7   8   9
2 |     2   4   6   8  10  12  14  16  18
3 |     3   6   9  12  15  18  21  24  27
4 |     4   8  12  16  20  24  28  32  36
5 |     5  10  15  20  25  30  35  40  45
6 |     6  12  18  24  30  36  42  48  54
7 |     7  14  21  28  35  42  49  56  63
8 |     8  16  24  32  40  48  56  64  72
9 |     9  18  27  36  45  54  63  72  81
```