Ira A. Fulton Schools of Engineering

school of computing, informatics, & decision systems engineering

CSE 110 – Assignment #8

Maximum points: 20 pts

Topics

- 2D Array (Chapter 6)
  - Object instance as an element of array.
- File I/O (Chapter 7)
  - Extract text information from a .txt file
  - File object, and Exception Handlings

Your programming assignments require individual work and effort to be of any benefit. Every student must work independently on his or her assignments. This means that every student must ensure that neither a soft copy nor a hard copy of their work gets into the hands of another student. Sharing your assignments with others in any way is **NOT** permitted. Violations of the University Academic Integrity policy will not be ignored. The university academic integrity policy is found at http://www.asu.edu/studentlife/judicial/integrity.html

Use the following Guidelines:

- Give identifiers semantic meaning and make them easy to read (examples numStudents, grossPay, etc).
- Keep identifiers to a reasonably short length.
- User upper case for constants. Use title case (first letter is upper case) for classes. Use lower case with uppercase word separators for all other identifiers (variables, methods, objects).
- Use tabs or spaces to indent code within blocks (code surrounded by braces). This includes classes, methods, and code associated with ifs, switches and loops. Be consistent with the number of spaces or tabs that you use to indent.
- Use white space to make your program more readable.

Important Note:

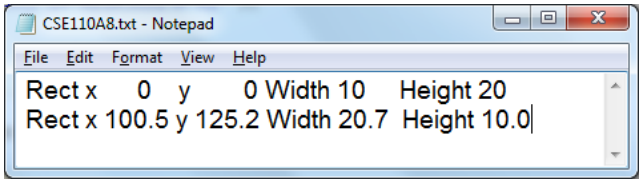
All submitted assignments must begin with the descriptive comment block. To avoid losing trivial points, make sure this comment header is included in every assignment you submit, and that it is updated accordingly from assignment to assignment. **(If not, -1 Pt)**

```
//*************************************************************
// Name: your name
// Title: title of the source file
// Description: Write the description in your words.
// Date: the date you programmed
//*************************************************************
```

Part 1: Writing Exercise (5 pts)

Consider the following code snippet. It uses `"[^A-Za-z]+"` for the input of `useDelimiter` method. Write the output when using each regular expression, a) – d) below. The regular expression is explained in out Textbook Chapter 7.

```
File f = new File("CSE110A8.txt");
Scanner in = new Scanner(f);
in.useDelimiter("[^A-Za-z]+");
while ( in.hasNext() ){
      String s = in.next();
      System.out.print(s + " ");
}
in.close();
```



a. `in.useDelimiter ("[^A-Za-z]+")`
b. `in.useDelimiter ("[^0-9]+")`
c. `in.useDelimiter ("[^0-9]*")`
d. `in.useDelimiter ("[0-9]+")`
e. Write the regular expression in the delimiter method to generate the following output :

```
in.useDelimiter ("???")

0 0 10 20 100.5 125.2 20.7 10.0
```

Note: The answers to the 5 questions (a through e) above should be typed in the block of comments in the `Dungeon`.java file such as;

```
/*
a. XXX XX XX XXX
…
e. in.useDelimiter ("XXXXX")
*/
```

Part 2: Programming (15 pts)

The goal of this assignment is to develop a program that generates a dungeon, which is a two-dimensional grid space (x, y). By reading monster's information from a file monsters are allocated to the grid cells. The user can shift and shuffle the monsters.

Download the following files and use them for this assignment.
***) Do not change any content of the following files.**

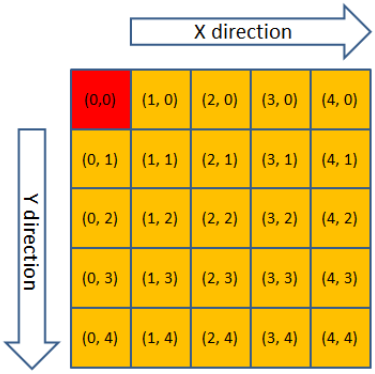
- Assignment8.java
- Monster.java
- Data1.txt
- Data2.txt

The last two files are input files (text) that will be read from Assignment8 class. Save all files in the same folder. You will be creating a class called **Dungeon**. This class should be defined in a file named "**Dungeon.java**". The class will contain a two-dimensional array called "`monsters`" of `Monster` objects. It also has two static variables, `gridWidth` and `gridHeight`. **(3 pt)**

Dungeon
- monsters: Monster[][] + gridWidth: int + gridHeight: int
+ Dungeon (int width, int height) + updateMonster(int x, int y, Monster m) : void - swap(int x1, int y1, int x2, int y2) : void + shuffle() : void + shift(char command) : void + printInfo() : String

As shown in the image below, the x direction is from left to right, and the y direction is from top to down. The left top is represented as (x, y) = (0, 0). The image shows a case in which the grid size is 5 by 5. All monster can move in four directions by reading a corresponding key; **up ('w'), left ('a'), down ('s') and right ('d').**

Before starting the Dungeon class, read the code of Assignment8.java and Monster.java carefully. Without the knowledge of methods of them or tasks in the Assignment8, it is impossible to complete this assignment.



The class Dungeon **MUST** include the following constructor and methods. (If your class does not contain any of the following methods, points will be deducted.)

- `public Dungeon(int width, int height)`
  It instantiates a two-dimensional array `monsters`. The column and row are defined by the input first and second parameters respectively. Each `Monster` object of `monsters` is constructed by using the Monster's default constructor, `Monster()`. Two static variables, `gridWidth` and `gridHeight` are initialized using the first and second inputs also. **(2 pts)**

- `public void updateMonster(int x, int y, Monster m)`
  It reads two integers and one Monster to replace the Monster at (x, y) position, `monsters[y][x] (* not monsters[x][y])` with the input one, **only if it equals to a default one (no name and -1 level)**. When **the input x and y are more than the dungeon's size**, do nothing. Without this validation, you will get error messages in INPUT2 when you submit to Gradescope.**(2 pts)**

- `public void shift(char command )`
  It reads a command and shifts all monsters to one cell in one direction corresponding to the command: **up ('w'), left ('a'), down ('s') and right ('d')** **(2 pts)**

- `public void shuffle()`
  It shuffles the monsters randomly. Use the `private swap(int x1, int y1, int x2, int y2)` method. **(2 pts)**

- `private void swap (int x1, int y1, int x2, int y2)`
  It reads two locations (x1, y1) and (x2, y2), and swap the monsters at the positions. This method is used in the `shuffle()` method. **(2 pts)**

- `public String printInfo()`
  It displays the initials of Monster's name and level at the allocated positions. Use the Monster's getInfo() method. **(2 pts)**

```
V-5     *       *       V-6     Y-2
Y-1     *       D-9     *       *
*       *       *       *       D-7
*       *       *       *       *
O-5     *       *       *       O-3
```

Use only the Java statements that have been covered in class to date. **DO NOT** use any other items out of the Chapter1- 8. If in doubt, ask. If you use them, then you lose the points of task. Don't copy any code developed by others. Don't **show** your code to others. Don't use any algorithm, which you cannot understand. Your assignment file is checked by the MOSS (by Stanford Univ.), which is a program to detect cheatings.

Testing Results: Input in red (Not seen at online-submission)

```
----------------
INPUT 1
----------------
a
5
5
e
b
Data1.txt
c
w
c
a
c
s
c
d
d
?
q
----------------
YOUR OUTPUT 1
----------------
*** Start of Program ***

Command Options
-----------------------------------
a: create a new dungeon
b: set the monsters
c: shift the monsters
d: shuffle the monsters
e: display the dungeon
```

```
?: display the menu again
q: quit this program


Please enter a command or type ? a
    a [Create a new dungeon]
    [Input the width of dungeon]: 5
    [Input the heigh of dungeon]: 5
Please enter a command or type ? e
    e [display the dungeon]
    *       *       *       *       *
    *       *       *       *       *
    *       *       *       *       *
    *       *       *       *       *
    *       *       *       *       *

Please enter a command or type ? b
    b [set the monsters]
    [Type the file name]: Data1.txt
    V-5     *       *       V-6     Y-2
    Y-1     *       D-9     *       *
    *       *       *       *       D-7
    *       *       *       *       *
    O-5     *       *       *       O-3


Please enter a command or type ? c
    c [shift the monsters]
    [Input the shift command ('w', 'a', 's', or 'd')]:w
    Y-1     *       D-9     *       *
    *       *       *       *       D-7
    *       *       *       *       *
    O-5     *       *       *       O-3
    V-5     *       *       V-6     Y-2

Please enter a command or type ? c
    c [shift the monsters]
    [Input the shift command ('w', 'a', 's', or 'd')]:a
    *       D-9     *       *       Y-1
    *       *       *       D-7     *
    *       *       *       *       *
    *       *       *       O-3     O-5
    *       *       V-6     Y-2     V-5

Please enter a command or type ? c
    c [shift the monsters]
    [Input the shift command ('w', 'a', 's', or 'd')]:s
    *       *       V-6     Y-2     V-5
    *       D-9     *       *       Y-1
    *       *       *       D-7     *
    *       *       *       *       *
    *       *       *       O-3     O-5
```

```
Please enter a command or type ? c
    c [shift the monsters]
    [Input the shift command ('w', 'a', 's', or 'd')]:d
    V-5     *       *       V-6     Y-2
    Y-1     *       D-9     *       *
    *       *       *       *       D-7
    *       *       *       *       *
    O-5     *       *       *       O-3

Please enter a command or type ? d
    d [shuffle the monsters]
    *       *       *       *       V-6
    *       D-9     *       *       *
    *       Y-2     *       *       *
    D-7     *       O-5     *       V-5
    *       *       *       O-3     Y-1

Please enter a command or type ? ?
Command Options
-----------------------------------
a: create a new dungeon
b: set the monsters
c: shift the monsters
d: shuffle the monsters
e: display the dungeon
?: display the menu again
q: quit this program


Please enter a command or type ? q
*** End of Program ***
```

/

Submit your homework by following the instructions below:

***/**

• Go to the course web site, and then click on the GradeScope on CANVAS.
• Submit the downloaded **Assignment8.java, Monster.java, and your Dungeon.java** files together with **Data1.txt and Data2.txt** on-line.

• The **Dungeon.java** should have the following, in order:

- ✓ In comments, the assignment Header described in "Important Note".
- ✓ In comments, the answers to questions a-e presented in Part#1.
- ✓ The working Java code requested in Part #2.
- ✓ All **jave** files must compile and run as you submit them. You can confirm this by viewing your submission results.

Important Note: You may resubmit as many times as you like until the deadline, but we will only mark your last submission. **NO LATE ASSIGNMENTS WILL BE ACCEPTED.**