

Seneca College

Applied Arts & Technology
SCHOOL OF COMPUTER STUDIES

JAC444**Submission date:****March 21, 2020**

Workshop 7

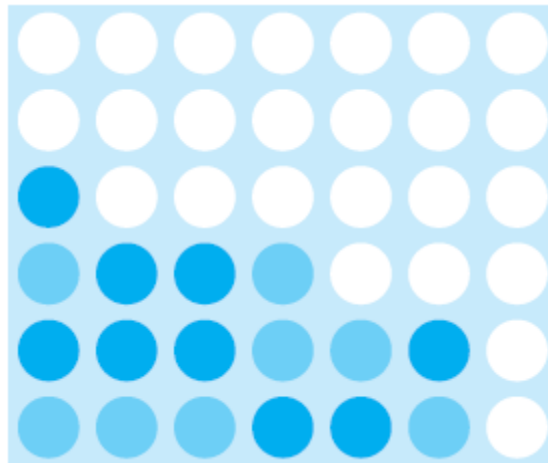
Description:

The following workshop lets you practice basic java coding techniques, creating classes, methods, using arrays, Java I/O, inheritance, polymorphism, Exceptional Handling, JavaFx (GUI), Lambda expressions, Functional Interface.

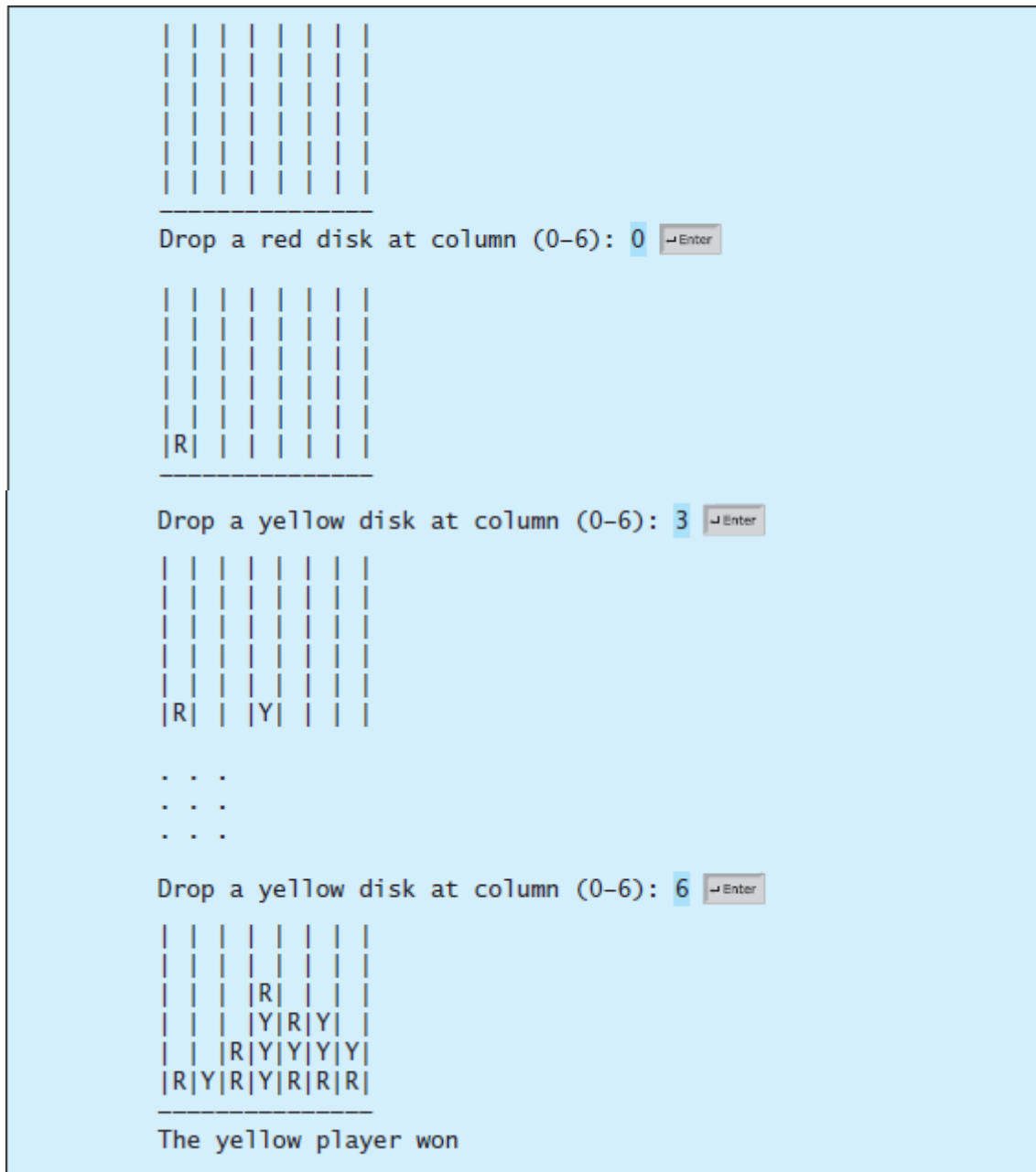
Task 1: (Game – Connect four)

(Note: Students have option the to use JavaFx or Console based for this task, the snap shot has be provided for both versions.)

Connect four is a two-player board game in which the players alternately drop colored disks into a seven-column, six-row vertically suspended grid, as shown below.



The objective of the game is to connect four same-colored disks in a row, a column, or a diagonal before your opponent can do likewise. The program prompts two players to drop a red or yellow disk alternately. In the preceding figure, the red disk is shown in a dark color and the yellow in a light color. Whenever a disk is dropped, the program redisplay the board on the console and determines the status of the game (win, draw, or continue). Here is a sample run:

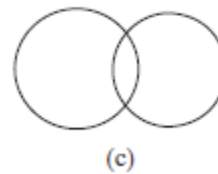
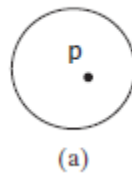


Task 2:

Define the Circle2D class that contains:

- Two double data fields named *x* and *y* that specify the center of the circle with getter methods.
- A data field *radius* with a getter method.
- A no-arg constructor that creates a default circle with (0, 0) for (*x*, *y*) and 1 for radius.
- A constructor that creates a circle with the specified *x*, *y*, and radius.
- A method *getArea()* that returns the area of the circle.

- A method `getPerimeter()` that returns the perimeter of the circle.
- A method `contains(double x, double y)` that returns true if the specified point (x, y) is inside this circle (Figure - a).
- A method `contains(Circle2D circle)` that returns true if the specified circle is inside this circle (Figure b).
- A method `overlaps(Circle2D circle)` that returns true if the specified circle overlaps with this circle (Figure c).



Write a test program that creates a **Circle2D** object **c1** (**`new Circle2D(2, 2, 5.5)`**), displays its area and perimeter, and displays the result of **`c1.contains(3,3)`**
`c1.contains(new Circle2D(4, 5, 10.5))`
`c1.overlaps(new Circle2D(3, 5, 2.3))`.

Workshop Header

/*****

Workshop #

Course:<subject type> - Semester

Last Name:<student last name>

First Name:<student first name>

ID:<student ID>

Section:<section name>

This assignment represents my own work in accordance with Seneca Academic Policy.

Signature

Date:<submission date>

*****/

Code Submission Criteria:

Please note that you should have:

- Appropriate indentation.
- Proper file structure
- Follow java naming convention
- Document all the classes properly
- Do Not have any debug/ useless code and/ or files in the assignment
- Do not have everything in the *main method*.
- Have a separate TestClass with the main method in it.
- Check your inputs if the user is not entering garbage inputs.
- Use exceptional handling or other methods to let the user know if the inputs are incorrect.

Deliverables and Important Notes:

All these deliverables are supposed to be uploaded on the blackboard once done.

- You are supposed to create video/ record voice/ detailed document of your running solution. **(40%)**
 - Screen Video captured file should state your last name and id, like Ali_123456.mp4 (or whatever the extension of the file is)
 - Record voice clip should also include a separate word file with the screen shots of your program's output, state your last name and id, like Ali_123456.mp3 (or whatever the extension of the file is)

- Detailed document should include screen shots of your output, have your name and id on the top of the file and save the file with your last name and id, like Ali_123456.docx (or whatever the extension of the file is)
- A word/ text file which will reflect on learning of your concepts in this workshop. (Also include the instructions on how to run your code. Which is only required if you have any special instructions for me on how to run your code.) **(30%)**
 - Should state your Full name and Id on the top of the file and save the file with your last name and id, like Ali_123456.txt
- Submission of working code. **(30%)**
 - Make sure your follow the “**Code Submission Criteria**” mentioned above.
 - You should zip your whole working project to a file named after your Last Name followed by the first 3 digits of your student ID. For example, **Ali123.zip**.
- Your marks will be deducted according to what is missing from the above-mentioned submission details.
- Late submissions would result in additional 10% penalties for each day or part of it.
- Remember that you are encouraged to talk to each other, to the instructor, or to anyone else about any of the assignments, but the final solution may not be copied from any source.