Task Sheet

General Instructions

You will have the afternoon to complete these tasks.

Create a new project for all of the following tasks. As a suggestion, you can call your main project “tasks” but you can call it whatever suits you.

Each task will go into its own separate package and will be run individually. Each task will require its own `main` method. This would mean the following structure with each `App.java` file having its own `main`.

```
|- com.sparta
        |
        |- day1
        |     |- debug
        |     |     |- DebugApp.java
        |     |
        |     |- calculator
        |           |- CalculatorApp.java
        |
        |- day2
```

For all tasks you should be able to explain and justify the solution that you arrive at.

Tasks

You Sank My Battleship!

Package:	com.sparta.day7.destroyer
Class:	DestroyerApp

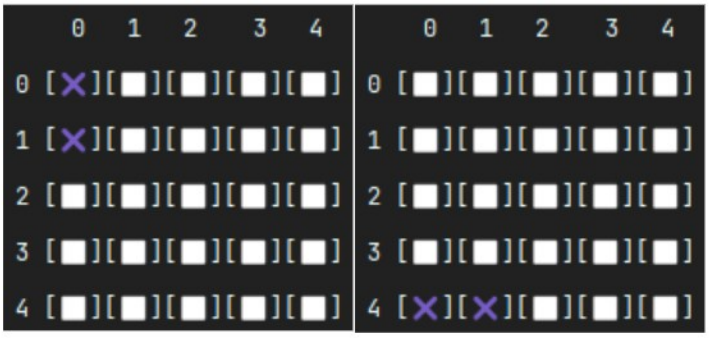
You may or may not be familiar with the old children's game Battleships (if not see https://en.wikipedia.org/wiki/Battleship_(game)).

You are going to create a simplified version.

In your program you will have a 5 x 5 game grid on which a Destroyer (which will take up two adjacent squares either vertically or horizontally) will be automatically generated.

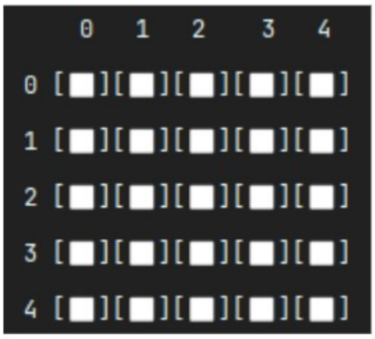
Your program will then display a representation of the gameboard (see Fig. 2) and prompt the user to enter a coordinate for a "shot", for example row 1 column 3. Using a function it will check to see if the shot is a hit or a miss and display an appropriate message (Fig. 3.).

The game will continue until the player "sinks" the destroyer by guessing the two positions of the ship (Fig. 4.).



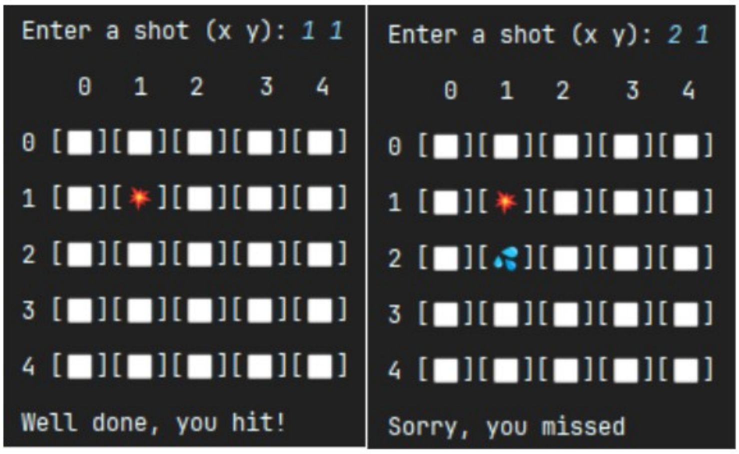
```
   0  1  2  3  4        0  1  2  3  4
0 [X][□][□][□][□]    0 [□][□][□][□][□]
1 [X][□][□][□][□]    1 [□][□][□][□][□]
2 [□][□][□][□][□]    2 [□][□][□][□][□]
3 [□][□][□][□][□]    3 [□][□][□][□][□]
4 [□][□][□][□][□]    4 [X][X][□][□][□]
```

Fig. 1. Examples of ship placement on the gameboard (for illustration only)



```
   0  1  2  3  4
0 [□][□][□][□][□]
1 [□][□][□][□][□]
2 [□][□][□][□][□]
3 [□][□][□][□][□]
4 [□][□][□][□][□]
```

Fig. 2. Example of the initial displayed gameboard.



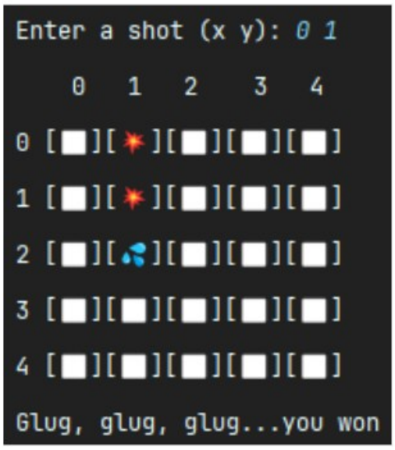
```
Enter a shot (x y): 1 1

    0  1  2  3  4
0 [⬜][⬜][⬜][⬜][⬜]
1 [⬜][💥][⬜][⬜][⬜]
2 [⬜][⬜][⬜][⬜][⬜]
3 [⬜][⬜][⬜][⬜][⬜]
4 [⬜][⬜][⬜][⬜][⬜]
Well done, you hit!
```

```
Enter a shot (x y): 2 1

    0  1  2  3  4
0 [⬜][⬜][⬜][⬜][⬜]
1 [⬜][💥][⬜][⬜][⬜]
2 [⬜][💦][⬜][⬜][⬜]
3 [⬜][⬜][⬜][⬜][⬜]
4 [⬜][⬜][⬜][⬜][⬜]
Sorry, you missed
```

Fig. 3. Example of the displayed gameboard after player has taken a shot.



```
Enter a shot (x y): 0 1

    0  1  2  3  4
0 [⬜][💥][⬜][⬜][⬜]
1 [⬜][💥][⬜][⬜][⬜]
2 [⬜][💦][⬜][⬜][⬜]
3 [⬜][⬜][⬜][⬜][⬜]
4 [⬜][⬜][⬜][⬜][⬜]
Glug, glug, glug...you won
```

Fig. 4. Example of the displayed gameboard after the player has won.