

Assignment Due Date and Time

24 November 2019 (Sun) 11:55 p.m.

Instructions to Students

1. This assignment is weighted 40% of the overall Continuous Assessment of this module.
2. This assignment is an individual assignment and should be done by you only. **Plagiarism will be treated seriously.** Any submitted assignment is found that involved wholly or partly in plagiarism (no matter the assignments are from the original authors or from the plagiarists) **will be scored Zero mark** and the students involved will be received discipline penalty set by the institute accordingly.
3. Only Java programming language is allowed to develop any required program.
4. Your programs must be well structured. A comment should be included in each class and method to state its main function. Explanation of each variable is also required. The source code must be properly indented. The first few lines in a source file must be a comment stating the name of the source file, your name, class, student number as well as the description of the purposes of the program. Marks will be deducted if any of the above mentioned comment is not included.
5. Grading of your programs will be based on correctness, quality, style, efficiency and the advanced features.
6. You are required to hand in a softcopy of the program source codes uploaded to Moodle.
7. Remember to backup all your programs.
8. Late submission will **NOT** be accepted.

Assignment Specification

You are asked to write a Battleship game in Java.

Information on the Battleship Game

The game is played on four grids, two for each player. The grids are typically square – usually 10×10 – and the individual squares in the grid are identified by letter and number. On one grid the player arranges ships and records the shots by the opponent. On the other grid the player records their own shots.

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Before play begins, each player secretly arranges their ships on their primary grid. Each ship occupies a number of consecutive squares on the grid, arranged either horizontally or vertically. The number of squares for each ship is determined by the type of the ship. The ships cannot overlap (i.e., only one ship can occupy any given square in the grid). The types and numbers of ships allowed are the same for each player. These may vary depending on the rules.

There are five kinds of ships in this game:

No.	Class of ship	Size
1	Carrier	5
2	Battleship	4
3	Destroyer	3
4	Submarine	3
5	Patrol Boat	2

Requirements of the Assignment

Game Flow

The Battleship game will be run in console mode. For simplicity, we only use two grids, one for Player1 to set the ships, and the other one is the hidden grid for Player2 to guess the position of the ships which are set by Player1.

For Player1, she can be a human to set the ships manually or can be a CPU to set the ships randomly.

So, at the beginning, you need to choose Player1 should be a human or CPU. If Player1 is "Human", then Player1 will start to set the battle ships:

```
Battleship Player1 (0 - Human, 1 - CPU, x - Exit): 0
```

```
      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 | . . . . . . . . . .
1 | . . . . . . . . . .
2 | . . . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^
Y
```

```
Player1: Set the ship: Carrier, ship size: 5
```

```
Orientation (0 - horizontal, 1 - vertical), x - Exit: 0
```

```
Position of Carrier [XY]: 21
```

```
      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 | . . . . . . . . . .
1 | . . S S S S S . . .
2 | . . . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^
Y
```

```
Set the ship: Battleship, ship size: 4
```

```
Orientation (0 - horizontal, 1 - vertical), x - Exit:
```

After Player1 to set all five battle ships, you need to clear the screen and then let Player2 start to guess the positions of five battle ships:

```
Set the ship: Patrol Boat, ship size: 2
Orientation (0 - horizontal, 1 - vertical), x - Exit: 0
Position of Patrol Boat [XY]: 68
    0 1 2 3 4 5 6 7 8 9 <--X
```

```
---+-----
0 | . . . . . . . . . .
1 | . . S S S S S . . .
2 | . . S . . . . . . .
3 | . . S . . . . . . .
4 | . . S . . . . . . .
5 | . . S S S S . . . .
6 | . . . . . S . . . .
7 | . . . . . S . . . .
8 | . . . . . S S S . .
9 | . . . . . . . . . .
^
```

Y

```
Press Enter for Player2 to start ...
(Clear screen with 100 System.out.println();)
```

```
    0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 | . . . . . . . . . .
1 | . . . . . . . . . .
2 | . . . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^
```

Y

```
Player2: Set your missile [XY], x - Exit:00
Missed.
Launched:1, Hit: 0
```

```
    0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 | o . . . . . . . . . .
1 | . . . . . . . . . .
2 | . . . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
```

```

7 | . . . . .
8 | . . . . .
9 | . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit:11
Missed.

Launched:2, Hit: 0

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . .
1 | . o . . . . .
2 | . . . . .
3 | . . . . .
4 | . . . . .
5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit:21
It's a hit!!

Launched:3, Hit: 1

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . .
1 | . o # . . . . .
2 | . . . . .
3 | . . . . .
4 | . . . . .
5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit:22
It's a hit!!

Launched:4, Hit: 2

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . .
1 | . o # . . . . .
2 | . . # . . . . .
3 | . . . . .
4 | . . . . .

```

```

5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit:

Error Control

For the input 0 or 1, your program should detect the following input errors:

Battleship Player1 (0 - Human, 1 - CPU, x - Exit): abc

Please input 0 or 1, x - Exit :123

Please input 0 or 1, x - Exit :9

Please input 0 or 1, x - Exit :

For the input [XY], your program should detect the following input errors:

Player2: Set your missile [XY], x - Exit:abc

Error in [XY]! Please input again, x - Exit: 123

Error in [XY]! Please input again, x - Exit: a

Error in [XY]! Please input again, x - Exit: 1

Error in [XY]! Please input again, x - Exit:

Player1 cannot set a ship over the boundary:

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 | . . . . .
1 | . . . . .
2 | . . . . .
3 | . . . . .
4 | . . . . .
5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^
Y

```

Player1: Set the ship: Carrier, ship size: 5

Orientation (0 - horizontal, 1 - vertical), x - Exit: 1

Position of Carrier [XY], x - Exit : 06

The ships cannot be over the boundary!

Position of Carrier [XY], x - Exit :

Player1 cannot set a ship to overlap other ships:

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . . .
1 | . S S S S S . . . .
2 | . . . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^

```

Y

Player1: Set the ship: Battleship, ship size: 4
 Orientation (0 - horizontal, 1 - vertical), x - Exit: 0
 Position of Battleship [XY]: 51
 The ships cannot overlap!
 Position of Battleship [XY]:

Player2 cannot fire on a repeated postion:

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | o . . . . . . . . . .
1 | . o # . . . . . . . .
2 | . . # . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : 22
 You have already fired this area.

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | o . . . . . . . . . .
1 | . o # . . . . . . . .
2 | . . # . . . . . . . .
3 | . . . . . . . . . .
4 | . . . . . . . . . .
5 | . . . . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .

```

^
Y
Player2: Set your missile [XY], x - Exit :

x - Exit

Your program MUST use following method to handle the "x - Exit" :

```
private static void xExit(String x){
    if (x.length() == 1 && x.charAt(0)=='x'){
        System.exit(0);
    }
}
```

End Game

When all ships have been hit, the game will be end:

Player2: Set your missile [XY], x - Exit:78
It's a hit!!
Launched:22, Hit: 17

```
      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # # . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # # # # o . . .
6 |  . . o o . # . . . .
7 |  . . . . . # . . . .
8 |  . . . . . # # # . .
9 |  . . . . . . . . . .
^
Y
You have hit all battleships!
```


Testing

You can ease the testing by using input redirection rather than inputting data manually. Prepare a text file, in the folder that contains your class file, which includes all user inputs in a game run. The following data.txt is an example.

abc
9

0
bb
8
0
ab
abc

21
1
22
0
35
1
56
0
68
abc
a

00
11
21
22
23
24
25
26
31
41
51
61
35
36
45

55
65
56
57
58
68
78

Then open a command prompt window and change to the folder that contains your class file. Type the command

`java Battleship < data.txt`

Or you can use “Copy & Paste” the above test case under the command prompt.

You can get the result automatically (without the input data echoed).

---- Battleship Game----

Battleship Player1 (0 - Human, 1 - CPU, x - Exit): abc

9

0

bb

8

0

ab

abc

60

21

1

22

0

25

35

1

56

0

68

abc

a

00

11

21

22

22

23

24

25

26

31

41
51
61
35
36
45
55
65
56
57
58
68
78

Please input 0 or 1, x - Exit : Please input 0 or 1, x - Exit : Please
input 0 or 1, x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```
--+-----
0 | . . . . .
1 | . . . . .
2 | . . . . .
3 | . . . . .
4 | . . . . .
5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^
```

Y

Player1: Set the ship: Carrier, ship size: 5

Orientation (0 - horizontal, 1 - vertical), x - Exit: Please input 0 or
1, x - Exit : Please input 0 or 1, x - Exit : Position of Carrier [XY],
x - Exit : Error in [XY]! Please input again, x - Exit : Error in [XY]!
Please input again, x - Exit: Error in [XY]! Please input again, x - Exit:
The ships cannot be over the boundary!

Position of Carrier [XY], x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```
--+-----
0 | . . . . .
1 | . . S S S S S . .
2 | . . . . .
3 | . . . . .
4 | . . . . .
5 | . . . . .
6 | . . . . .
7 | . . . . .
8 | . . . . .
9 | . . . . .
^
```

Y

Player1: Set the ship: Battleship, ship size: 4

Orientation (0 - horizontal, 1 - vertical), x - Exit: Position of Battleship [XY], x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . . .
1 | . . S S S S S . . .
2 | . . S . . . . . . .
3 | . . S . . . . . . .
4 | . . S . . . . . . .
5 | . . S . . . . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^

```

Y

Player1: Set the ship: Destroyer, ship size: 3

Orientation (0 - horizontal, 1 - vertical), x - Exit: Position of Destroyer [XY], x - Exit : The ships cannot overlap!

Position of Destroyer [XY], x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . . .
1 | . . S S S S S . . .
2 | . . S . . . . . . .
3 | . . S . . . . . . .
4 | . . S . . . . . . .
5 | . . S S S S . . . .
6 | . . . . . . . . . .
7 | . . . . . . . . . .
8 | . . . . . . . . . .
9 | . . . . . . . . . .
^

```

Y

Player1: Set the ship: Submarine, ship size: 3

Orientation (0 - horizontal, 1 - vertical), x - Exit: Position of Submarine [XY], x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . . .
1 | . . S S S S S . . .
2 | . . S . . . . . . .
3 | . . S . . . . . . .
4 | . . S . . . . . . .
5 | . . S S S S . . . .
6 | . . . . . S . . . .
7 | . . . . . S . . . .
8 | . . . . . S . . . .
9 | . . . . . . . . . .
^

```

Y

Player1: Set the ship: Patrol Boat, ship size: 2

Orientation (0 - horizontal, 1 - vertical), x - Exit: Position of Patrol Boat [XY], x - Exit :

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . .
1 | . . S S S S S . .
2 | . . S . . . . . .
3 | . . S . . . . . .
4 | . . S . . . . . .
5 | . . S S S S . . .
6 | . . . . . S . . .
7 | . . . . . S . . .
8 | . . . . . S S S .
9 | . . . . . . . . .
^

```

Y

Press Enter for Player2 to start ...

(100 System.out.println())

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | . . . . . . . . .
1 | . . . . . . . . .
2 | . . . . . . . . .
3 | . . . . . . . . .
4 | . . . . . . . . .
5 | . . . . . . . . .
6 | . . . . . . . . .
7 | . . . . . . . . .
8 | . . . . . . . . .
9 | . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : Error in [XY]! Please input again, x - Exit: Error in [XY]! Please input again, x - Exit: Missed.

Launched:1, Hit: 0

0 1 2 3 4 5 6 7 8 9 <--X

```

--+-----
0 | o . . . . . . . . .
1 | . . . . . . . . .
2 | . . . . . . . . .
3 | . . . . . . . . .
4 | . . . . . . . . .
5 | . . . . . . . . .
6 | . . . . . . . . .
7 | . . . . . . . . .
8 | . . . . . . . . .
9 | . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : Missed.
 Launched:2, Hit: 0

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o . . . . . . . . .
2 | . . . . . . . . . . .
3 | . . . . . . . . . . .
4 | . . . . . . . . . . .
5 | . . . . . . . . . . .
6 | . . . . . . . . . . .
7 | . . . . . . . . . . .
8 | . . . . . . . . . . .
9 | . . . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:3, Hit: 1

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o # . . . . . . . .
2 | . . . . . . . . . . .
3 | . . . . . . . . . . .
4 | . . . . . . . . . . .
5 | . . . . . . . . . . .
6 | . . . . . . . . . . .
7 | . . . . . . . . . . .
8 | . . . . . . . . . . .
9 | . . . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:4, Hit: 2

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o # . . . . . . . .
2 | . . # . . . . . . . .
3 | . . . . . . . . . . .
4 | . . . . . . . . . . .
5 | . . . . . . . . . . .
6 | . . . . . . . . . . .
7 | . . . . . . . . . . .
8 | . . . . . . . . . . .
9 | . . . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : You have already fired this area.

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o # . . . . . . .
2 | . . # . . . . . . .
3 | . . . . . . . . .
4 | . . . . . . . . .
5 | . . . . . . . . .
6 | . . . . . . . . .
7 | . . . . . . . . .
8 | . . . . . . . . .
9 | . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:5, Hit: 3

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o # . . . . . . .
2 | . . # . . . . . . .
3 | . . # . . . . . . .
4 | . . . . . . . . .
5 | . . . . . . . . .
6 | . . . . . . . . .
7 | . . . . . . . . .
8 | . . . . . . . . .
9 | . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:6, Hit: 4

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 | o . . . . . . . . .
1 | . o # . . . . . . .
2 | . . # . . . . . . .
3 | . . # . . . . . . .
4 | . . # . . . . . . .
5 | . . . . . . . . .
6 | . . . . . . . . .
7 | . . . . . . . . .
8 | . . . . . . . . .
9 | . . . . . . . . .
^
Y

```

Player2: Set your missile [XY], x - Exit : It's a hit!!

Launched:7, Hit: 5

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . . . . . .
1 |  . o # . . . . . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # . . . . . . .
6 |  . . . . . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : Missed.
Launched:8, Hit: 5

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . . . . . .
1 |  . o # . . . . . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # . . . . . . .
6 |  . . o . . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:9, Hit: 6

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . . . . . .
1 |  . o # # . . . . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # . . . . . . .
6 |  . . o . . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:10, Hit: 7


```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # . . . .
2 |  . . # . . . . . .
3 |  . . # . . . . . .
4 |  . . # . . . . . .
5 |  . . # . . . . . .
6 |  . . o . . . . . .
7 |  . . . . . . . . .
8 |  . . . . . . . . .
9 |  . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:11, Hit: 8

```

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # . . . .
2 |  . . # . . . . . .
3 |  . . # . . . . . .
4 |  . . # . . . . . .
5 |  . . # . . . . . .
6 |  . . o . . . . . .
7 |  . . . . . . . . .
8 |  . . . . . . . . .
9 |  . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:12, Hit: 9

```

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # # . . .
2 |  . . # . . . . . .
3 |  . . # . . . . . .
4 |  . . # . . . . . .
5 |  . . # . . . . . .
6 |  . . o . . . . . .
7 |  . . . . . . . . .
8 |  . . . . . . . . .
9 |  . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:13, Hit: 10

```

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # # . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # # . . . . . .
6 |  . . o . . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : Missed.
Launched:14, Hit: 10

```

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # # . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # # . . . . . .
6 |  . . o o . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:15, Hit: 11

```

```

      0 1 2 3 4 5 6 7 8 9 <--X
---+-----
0 |  o . . . . . . . . .
1 |  . o # # # # # . . .
2 |  . . # . . . . . . .
3 |  . . # . . . . . . .
4 |  . . # . . . . . . .
5 |  . . # # # . . . . .
6 |  . . o o . . . . . .
7 |  . . . . . . . . . .
8 |  . . . . . . . . . .
9 |  . . . . . . . . . .
^
Y
Player2: Set your missile [XY], x - Exit : It's a hit!!
Launched:16, Hit: 12

```

```

      0 1 2 3 4 5 6 7 8 9 <--X

```

```

--+-----
0 |  o . . . . .
1 |  . o # # # # . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # . .
6 |  . . o o . . . .
7 |  . . . . .
8 |  . . . . .
9 |  . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : Missed.
 Launched:17, Hit: 12

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . .
1 |  . o # # # # . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # o . .
6 |  . . o o . . . .
7 |  . . . . .
8 |  . . . . .
9 |  . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:18, Hit: 13

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . .
1 |  . o # # # # . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # o . .
6 |  . . o o . # . . .
7 |  . . . . .
8 |  . . . . .
9 |  . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!
 Launched:19, Hit: 14

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----

```

```

0 |  o . . . . .
1 |  . o # # # # . . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # o . . .
6 |  . . o o . # . . . .
7 |  . . . . . # . . . .
8 |  . . . . . . . . .
9 |  . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!

Launched:20, Hit: 15

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . .
1 |  . o # # # # . . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # o . . .
6 |  . . o o . # . . . .
7 |  . . . . . # . . . .
8 |  . . . . . # . . . .
9 |  . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!

Launched:21, Hit: 16

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . .
1 |  . o # # # # . . .
2 |  . . # . . . . .
3 |  . . # . . . . .
4 |  . . # . . . . .
5 |  . . # # # # o . . .
6 |  . . o o . # . . . .
7 |  . . . . . # . . . .
8 |  . . . . . # # . . .
9 |  . . . . . . . . .
^

```

Y

Player2: Set your missile [XY], x - Exit : It's a hit!!

Launched:22, Hit: 17

```

      0 1 2 3 4 5 6 7 8 9 <--X
--+-----
0 |  o . . . . .

```

```

1 | . o # # # # # . . .
2 | . . # . . . . . .
3 | . . # . . . . . .
4 | . . # . . . . . .
5 | . . # # # # o . . .
6 | . . o o . # . . . .
7 | . . . . . # . . . .
8 | . . . . . # # # . .
9 | . . . . . . . . . .
^

```

Y

You have hit all battleships!

Example codes for WRONG Scanner usage

```

// create new Scanner objects in loop
do {
    Scanner sc = new Scanner( System.in);
    choice = sc.nextInt();
} while (choice != 1);

```

You are NOT allowed to use GUI such as JOptionPane in your program.

Following is an example program to use a Global Scanner to do the input.

```

import java.util.Scanner;

public class Test {
    //Global declaration for Scanner
    public static Scanner sc = new Scanner(System.in);

    public static void main(String args[]) {
        int x;
        System.out.print("Enter x:");
        x = sc.nextInt();
    }

    public static void method2() {
        int y;
        System.out.print("Enter y:");
        y = sc.nextInt();
    }
}

```

Marking Scheme

Functions

Correctly display the game board	10
Player1 can manually set the ships	10
Player1 can automatically set the ships	10
Player2 can set the bomb on the game board	5
Correctly show the result of bomb (hit or miss)	5
Error Checking	
Input Error	10
Player1 set an overlap position	5
Player1 set a ship over the boundary	5
Player2 hit on a repeated position	5
Correctly show the statistics of the game	10
Game finishes appropriately	<u>5</u>
Style	10
Documentation (Comments)	10

Deduction

Cannot pass the Java compiler	-100
Do not follow the requirements in “ Requirements of the Assignment ”	-30
Cannot do the Test Case in the “ Testing ”	-30
More than one Scanner objects in your codes	-20
Do not use Scanner as the one and only one input method in your codes	-50
Cannot perform a demonstration to your lecturer	-50