

## Assignment Due Date and Time

21 Nov 2021 (Sun) 11:55 pm

## Instructions to Students

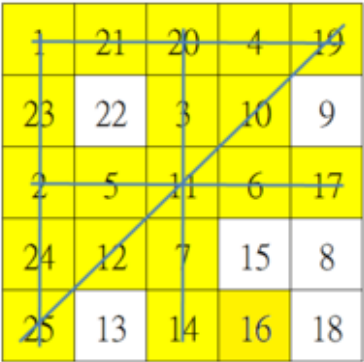
1. This assignment is weighted 40% of the overall Continuous Assessment of this module.
2. This assignment is an individual assignment and should be done by you only. Plagiarism will be treated seriously. Any submitted assignment is found that involved wholly or partly in plagiarism (no matter the assignments 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 program source codes.
9. Late submission will **NOT** be accepted.

## Assignment Specification

You are asked to write a BINGO game in Java.

### *Information on the BINGO*

**BINGO** is a game of chance in which each player matches numbers (from 1 to 25) printed in different arrangements on cards with the numbers the game host (caller) draws at random, marking the selected numbers with tiles. When a player finds the selected numbers are arranged on their card in a row, they call out "Bingo!" to alert all participants to a winning card, which prompts the game host (or an associate assisting the host) to examine the card for verification of the win.



1	21	20	4	19
23	22	3	10	9
2	5	11	6	17
24	12	7	15	8
25	13	14	16	18

You can watch the game play on the Youtube:

<https://youtu.be/ANwucID0vAo>

### *Requirements of the Assignment*

The BINGO game will be run in console mode. At the beginning, the program will display two players' cards on the screen. To simplify the program and the test procedure, the arrangement of 25 numbers on the cards are **fixed**.

```

Player1's Card:
24  2  8  1 25
12 16  7 17 15
 5  6 20 19 13
14 23 22  4  3
10 18 11 21  9

```

```

Player2's Card:
24 21 17 15  6
10  3  8 18 20
14  7 16 12  5
25 23 13 19 11
22  4  9  1  2

```

```
Game Host call (0 to exit): █
```

A Game Host will input a number to your program. According to the corresponding number on players' cards, your program will mark a "XX" to replace the number.

```
Game Host call (0 to exit): 16
```

```

Player1's Card:
24  2  8  1 25
12 XX  7 17 15
 5  6 20 19 13
14 23 22  4  3
10 18 11 21  9

```

```

Player2's Card:
24 21 17 15  6
10  3  8 18 20
14  7 XX 12  5
25 23 13 19 11
22  4  9  1  2

```

```
Game Host call (0 to exit): █
```

The program will show who is the winner if a player wins, and then end the game.

```
Game Host call (0 to exit): 23
```

```

Player1's Card:
24 XX  8  1 25
12 XX  7 17 15
 5 XX XX 19 13
14 XX XX  4  3
XX XX 11 21  9

```

```

Player2's Card:
24 21 17 15 XX
XX  3  8 XX XX
14  7 XX 12  5
25 XX 13 19 11
XX  4  9  1 XX

```

```
Player1 Bingo!
```

```
Player2 Bingo!
```

The program should also detect the following input errors:

```
Game Host call (0 to exit): 26
The number must be between 1 to 25, please call again!
Game Host call (0 to exit): -1
The number must be between 1 to 25, please call again!
```

```
Game Host call (0 to exit): 16

Player1's Card:
24  2  8  1 25
12 XX  7 17 15
 5  6 20 19 13
14 23 22  4  3
10 18 11 21  9

Player2's Card:
24 21 17 15  6
10  3  8 18 20
14  7 XX 12  5
25 23 13 19 11
22  4  9  1  2

Game Host call (0 to exit): 16
The number 16 is repeated, please call again!
Game Host call (0 to exit): █
```

Furthermore, your program MUST allows Game Host to input 0 to quit the game.

## Testing Method

You can ease the testing by using '**Copy and Paste**' rather than inputting data manually. Prepare a text file, which includes all user inputs in a game run. The following input is an example.

By using Copy and Paste, you can automatically input in the command prompt window and then get the result automatically (without the input data echoed).

### Test Case

28  
-2  
16  
10  
10  
22  
6  
20  
18  
2  
23

### Expected Output

Player1's Card:

```
24  2  8  1 25
12 16  7 17 15
 5  6 20 19 13
14 23 22  4  3
10 18 11 21  9
```

Player2's Card:

```
24 21 17 15  6
10  3  8 18 20
14  7 16 12  5
25 23 13 19 11
22  4  9  1  2
```

Game Host call (0 to exit): 28

-2  
16  
10  
10  
22  
6  
20  
18  
2  
23

The number must be between 1 to 25, please call again!

Game Host call (0 to exit): The number must be between 1 to 25, please call again!

Game Host call (0 to exit):

Player1's Card:

```
24  2  8  1 25
12 XX  7 17 15
 5  6 20 19 13
14 23 22  4  3
```

Data inputted by  
using copy and  
paste

---

```
10 18 11 21 9

Player2's Card:
24 21 17 15 6
10 3 8 18 20
14 7 XX 12 5
25 23 13 19 11
22 4 9 1 2

Game Host call (0 to exit):
Player1's Card:
24 2 8 1 25
12 XX 7 17 15
5 6 20 19 13
14 23 22 4 3
XX 18 11 21 9

Player2's Card:
24 21 17 15 6
XX 3 8 18 20
14 7 XX 12 5
25 23 13 19 11
22 4 9 1 2

Game Host call (0 to exit): The number 10 is repeated, please call again!
Game Host call (0 to exit):
Player1's Card:
24 2 8 1 25
12 XX 7 17 15
5 6 20 19 13
14 23 XX 4 3
XX 18 11 21 9

Player2's Card:
24 21 17 15 6
XX 3 8 18 20
14 7 XX 12 5
25 23 13 19 11
XX 4 9 1 2

Game Host call (0 to exit):
Player1's Card:
24 2 8 1 25
12 XX 7 17 15
5 XX 20 19 13
14 23 XX 4 3
XX 18 11 21 9

Player2's Card:
24 21 17 15 XX
XX 3 8 18 20
14 7 XX 12 5
25 23 13 19 11
XX 4 9 1 2

Game Host call (0 to exit):
Player1's Card:
24 2 8 1 25
12 XX 7 17 15
5 XX XX 19 13
14 23 XX 4 3
XX 18 11 21 9

Player2's Card:
24 21 17 15 XX
```

```
XX  3  8 18 XX
14  7 XX 12  5
25 23 13 19 11
XX  4  9  1  2
```

Game Host call (0 to exit):

Player1's Card:

```
24  2  8  1 25
12 XX  7 17 15
 5 XX XX 19 13
14 23 XX  4  3
XX XX 11 21  9
```

Player2's Card:

```
24 21 17 15 XX
XX  3  8 XX XX
14  7 XX 12  5
25 23 13 19 11
XX  4  9  1  2
```

Game Host call (0 to exit):

Player1's Card:

```
24 XX  8  1 25
12 XX  7 17 15
 5 XX XX 19 13
14 23 XX  4  3
XX XX 11 21  9
```

Player2's Card:

```
24 21 17 15 XX
XX  3  8 XX XX
25 23 13 19 11
XX  4  9  1 XX
```

Game Host call (0 to exit):

Player1's Card:

```
24 XX  8  1 25
12 XX  7 17 15
 5 XX XX 19 13
14 XX XX  4  3
XX XX 11 21  9
```

Player2's Card:

```
24 21 17 15 XX
XX  3  8 XX XX
14  7 XX 12  5
25 XX 13 19 11
XX  4  9  1 XX
```

Player1 Bingo!

Player2 Bingo!

## **Requirement for Scanner usage**

### **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**

20 marks

Correctly draw the players' card with marked numbers according to Game Host's input.

20 marks

Your program will detect two kinds of errors:  
Game Host input error: input number is not within 0 to 25.  
Game Host input error: input duplicated number.

Game Host input 0 to quit the game.

40 marks

Correct program logic - determine winner and then end game.

10 marks

Use good program structure (indentation), appropriate methods and meaningful variables' names.

10 marks

Documentation (Comments).

**Deduction**

Cannot pass the Java compiler	-100
Do not follow the requirement and corresponding output in "Testing Method"	-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