Project 1

Battleship Game

(Updated with pointers, structures, and files)

CIS-17A

Professor:
Dr. Mark E. Lehr

Name:
Jyro Jimenez

Date:
July 17 2021

Introduction

Title: Battleship game (Updated)

This game is played with two people, however this specific code focuses on one player and a computer as an opponent. The objective of the game is to be the first to sink all five boats of your opponent. Whoever sinks the other's boats first wins the game. The rules are simple, each player has their own field, with different orientations of boats. Both players do not know the pattern of each other's boats, as they have to guess where to shoot in order to sink a ship. Each player calls on a coordinate to initialize an attack, and will be prompted whether they hit a ship or not. With this prompt, the code will illustrate with an "X" if a player misses, or an "O" if a player hits a ship. The game will end if all 9 hittable slots are hit by a player or computer.

Summary

Project size: 460+ lines with comments

Number of Variables: 20

Within this project, I recycled an old game I created for a project from csc-5, where I utilized what I had already built. In this old project, it is a version of the game which utilized while loops, functions, linear search, and 2-dimensional arrays to mimic the game "Battleship." In this game, it mimics the basic rules of battleship, where the pattern of both the player and computer's boards are randomized. Through this, it allows the player and computer to guess where the player's ships are and are informed whether they hit an enemy ship or not.

Through this barebones project that I had already built, I updated the game to illustrate the different concepts we have learned so far in CIS-17A. These concepts include pointers, memory allocation, char arrays, structured data, and the use of binary files. These concepts have allowed me to optimize the game by including different approaches

to the code, while shortening the length itself. The previous chunk of code had been 500+ lines, whereis this chunk only amounted to 465+ lines. The project has lost 50 lines of code, but its substance includes more concepts than the previous, where it includes concepts from ch1-12 of the Gaddis book.

While the difference in length is a change within the project, the overall feel and rules aren't much different. I simply utilized a structure to hold all the data, and switched the arrays to a character array rather than a string, while making each array and structure dynamically allocated for. As for the use of binary files, I simply wrote the results of the game onto a binary and text file.

Project Description

The purpose of this project is as follows:

1. Ask the player If they would like to play Battleship.
2. Play Battleship with a player and the computer.
3. State who won depending on who destroyed all the other player's ships.

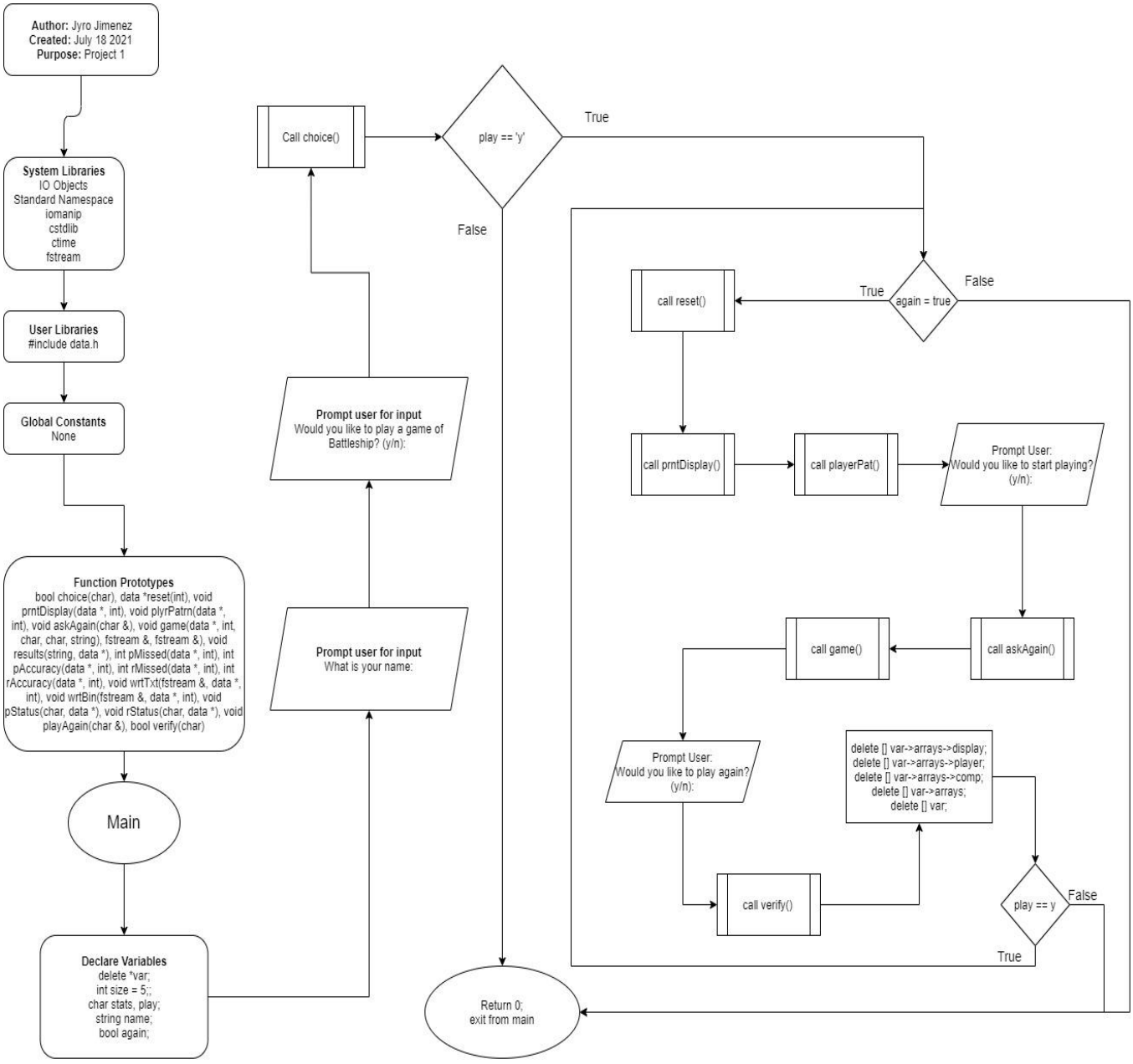
Pseudo Code

// System Libraries Here
// User Libraries Here
// Global Constants Only, No Global Variables
// Columns in the 2-D Arrays
// Like PI, e, Gravity, or conversions
// Program Execution Begins Here
// Set Random Number seed
// Declare Variables
// Local Constants
// Initialize structured data
// Initialize other variables
// Initialize Fstream operators
// Ask if player wants to play
// Initialize bool operator

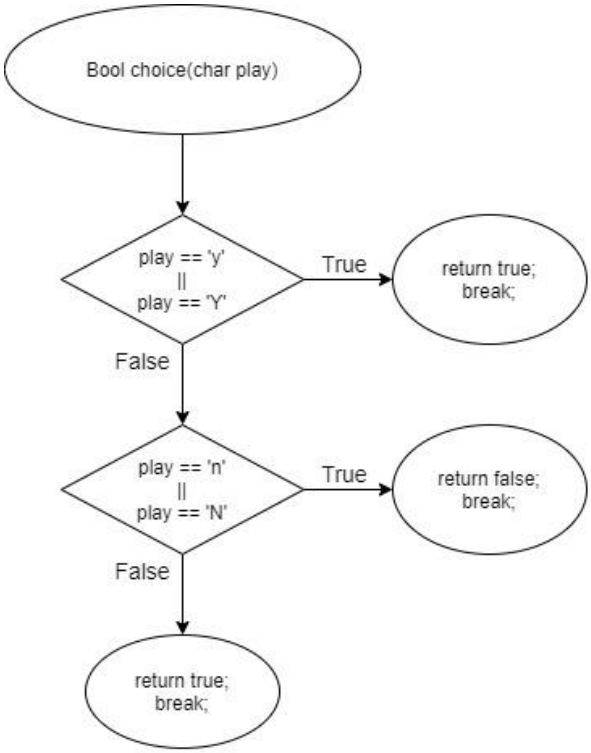
// Start game in loop with bool operator
// Reset player, display, and computer table
// Initialize each table with a different pattern
// Process/Map inputs to outputs
// Initialize count
// Pattern for player
// Assign player's pattern depending on random
// Pattern for Computer
// Pass By Reference
// Assign computer's pattern depending on random
// Output data
// Output player's display
// Output player's table
// Ask User if they want to start playing
// Loop just in case they say no
// Ask User where they want hit
// Display player's choices
// Ask player where they would like to hit
// See if player hit a ship or not
// Output that computer is choosing
// Use random and linear search to see if computer chose that number already;
// See if Computer hit a ship or not
// Output player's table
// Output Results
// Use linear search to determine how much you missed and how much you hit.
// Calculate player and computer's accuracy
// Ask if they would like to know their accuracy
// Ask if they would like to know computer's accuracy
// Give player choice to play again or not
// Verify player choice
// Exit

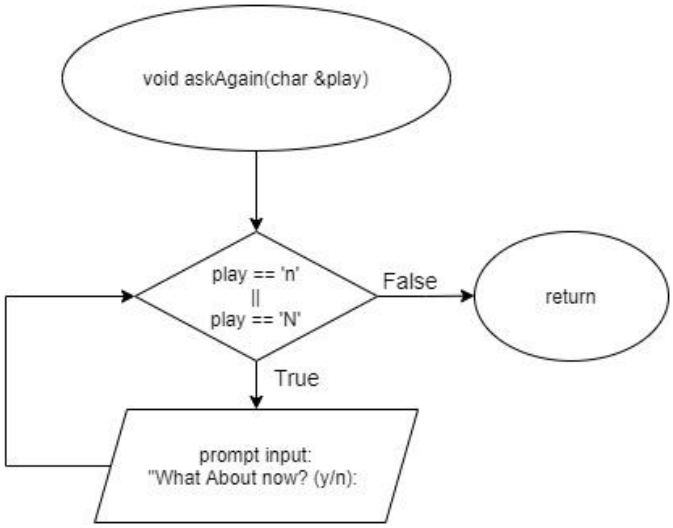
Flow Chart

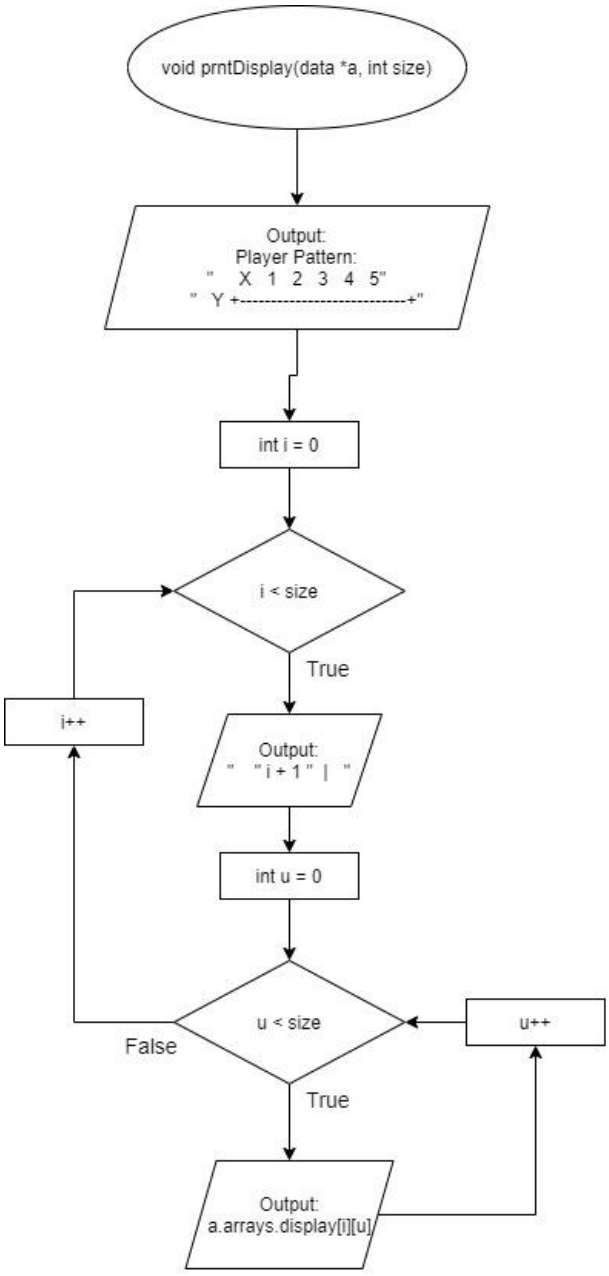
Full flow chart:

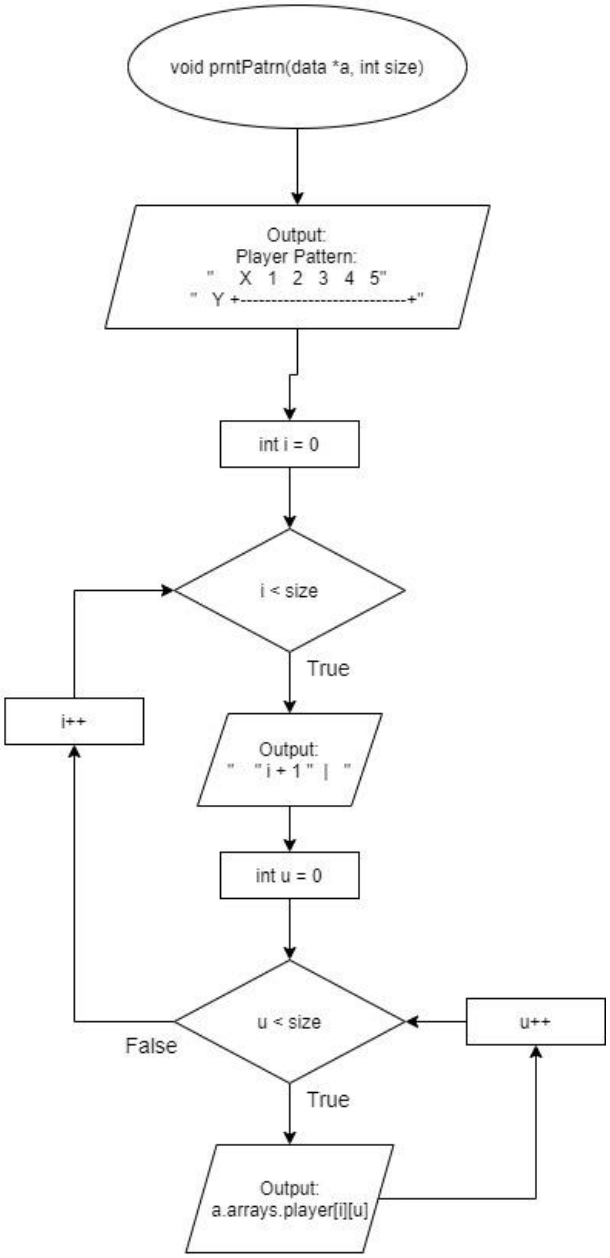


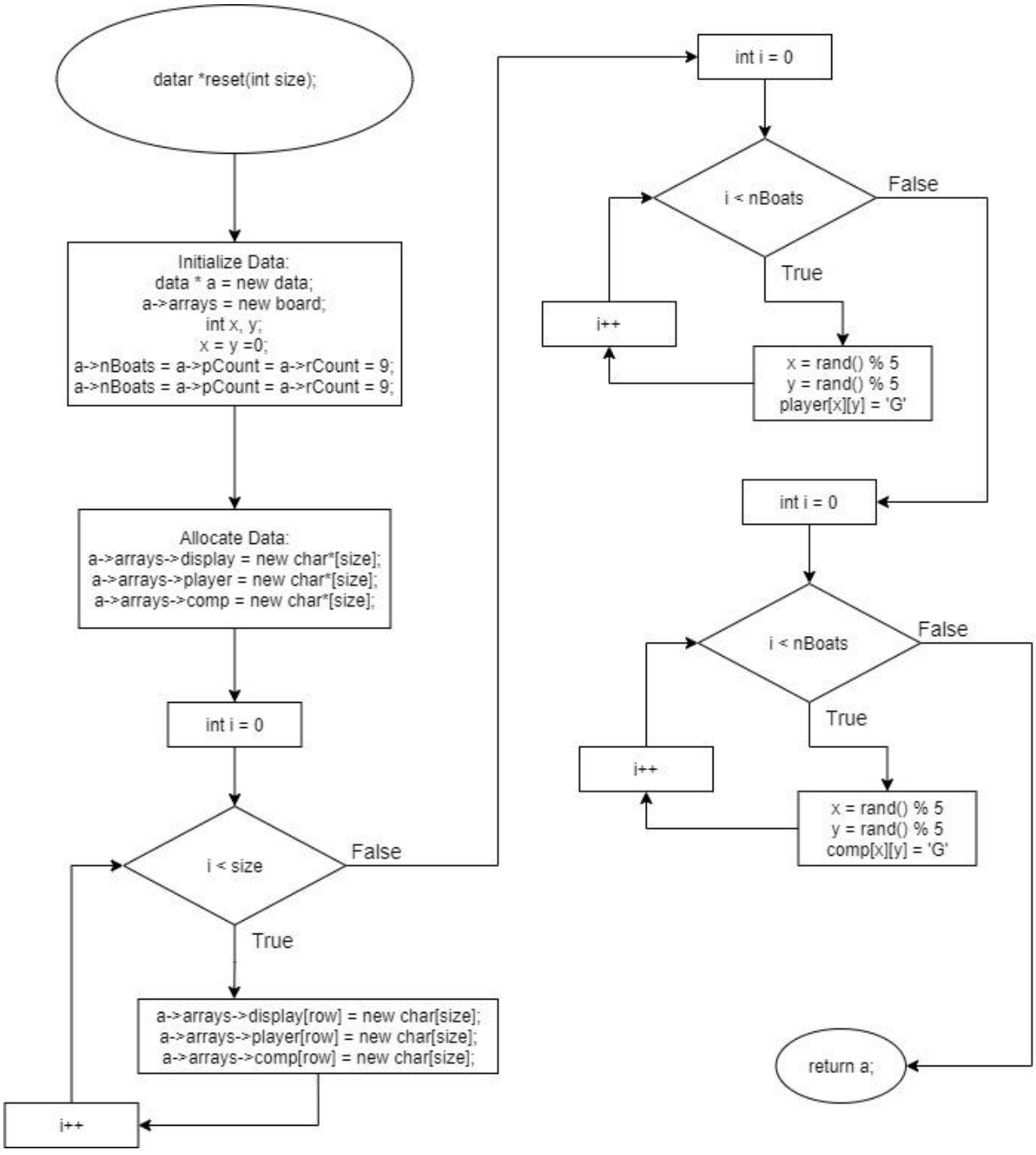
Flow chart functions:

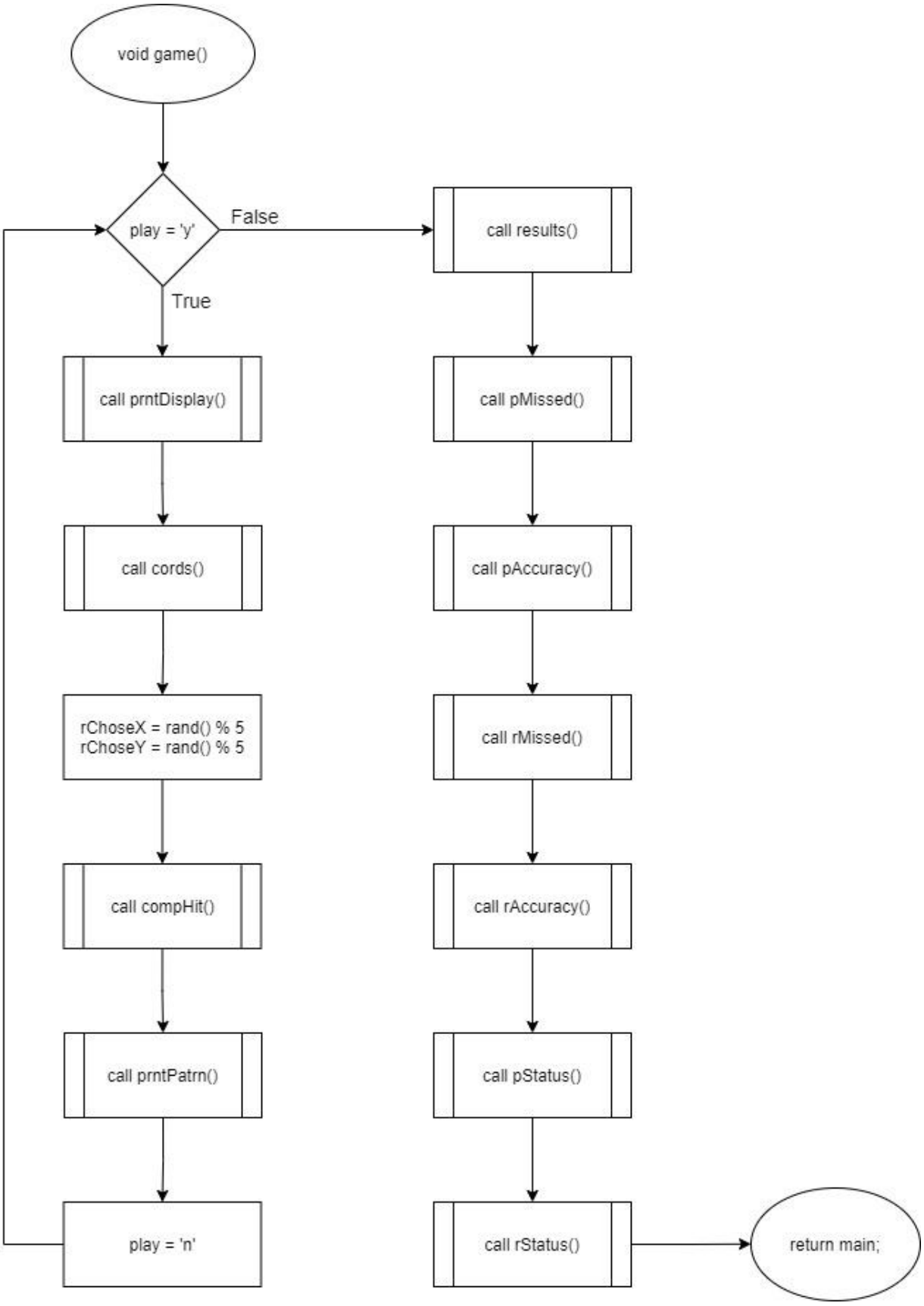





void prntDisplay(data *a, int size)
Output:
Player Pattern:
" X 1 2 3 4 5"
" Y +----------+"
int i = 0
i < size
True
i++
Output:
" " i + 1 " | "
int u = 0
u < size
u++
False
True
Output:
a.arrays.display[i][u]


void prntPatrn(data *a, int size)
Output:
Player Pattern:
" X 1 2 3 4 5"
" Y +----------+"
int i = 0
i < size
True
i++
Output:
" " i + 1 " | "
int u = 0
u < size
u++
False
True
Output:
a.arrays.player[i][u]


datar *reset(int size);
Initialize Data:
data * a = new data;
a->arrays = new board;
int x, y;
x = y =0;
a->nBoats = a->pCount = a->rCount = 9;
a->nBoats = a->pCount = a->rCount = 9;
Allocate Data:
a->arrays->display = new char*[size];
a->arrays->player = new char*[size];
a->arrays->comp = new char*[size];
int i = 0
i < size
False
True
a->arrays->display[row] = new char[size];
a->arrays->player[row] = new char[size];
a->arrays->comp[row] = new char[size];
i++
int i = 0
i < nBoats
False
True
i++
x = rand() % 5
y = rand() % 5
player[x][y] = 'G'
int i = 0
i < nBoats
False
True
i++
x = rand() % 5
y = rand() % 5
comp[x][y] = 'G'
return a;

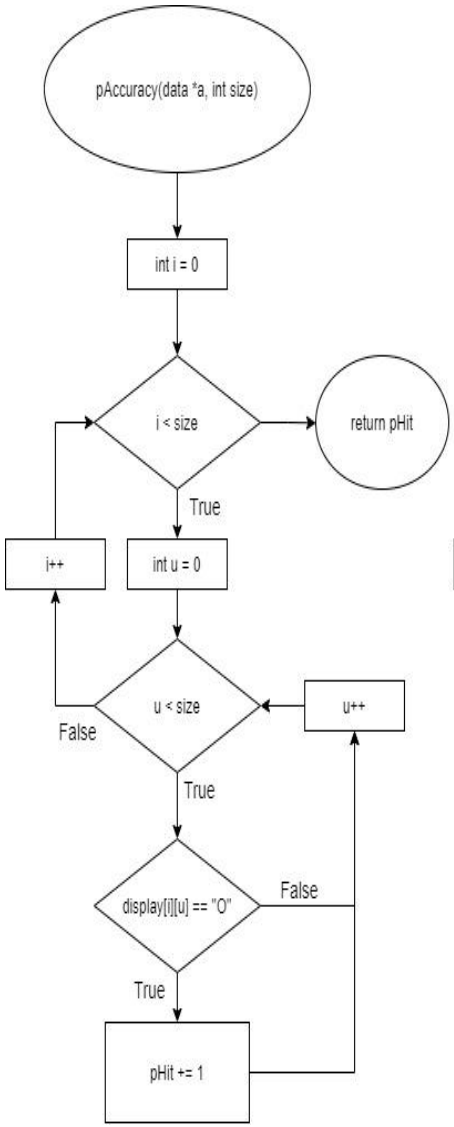

void game()
play = 'y'
False
True
call results()
call prntDisplay()
call pMissed()
call cords()
call pAccuracy()
rChoseX = rand() % 5
rChoseY = rand() % 5
call rMissed()
call compHit()
call rAccuracy()
call prntPatrn()
call pStatus()
play = 'n'
call rStatus()
return main;

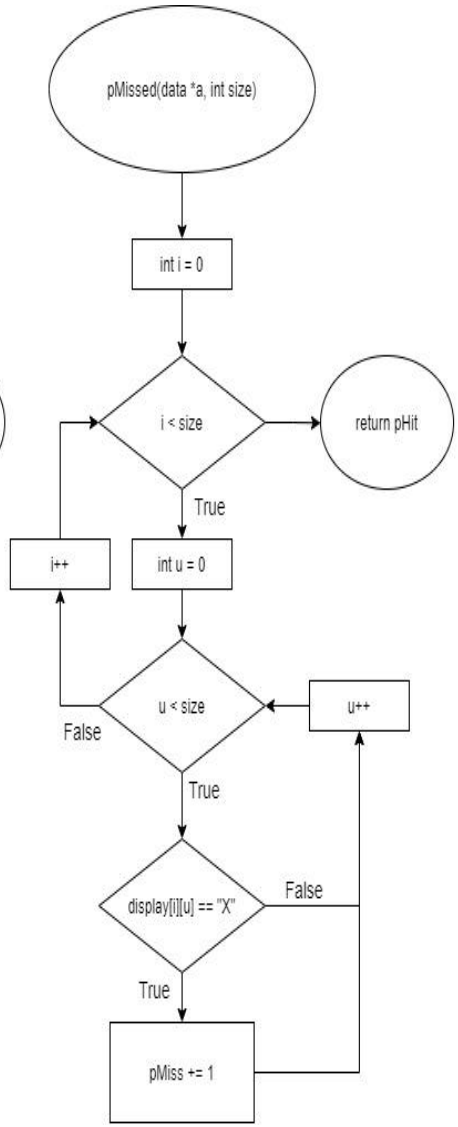
int cords(data *a)
prompt input
What coordinates would you like to shoot in?
X-Coordinate
Y-Coordinate
pChoseX > 8 || pChoseY > 8 ||
pChoseX <= 0 || pChoseY <= 0
True
prompt input
These numbers are invalid try again...
X-Coordinate
Y-Coordinate
True
False
return

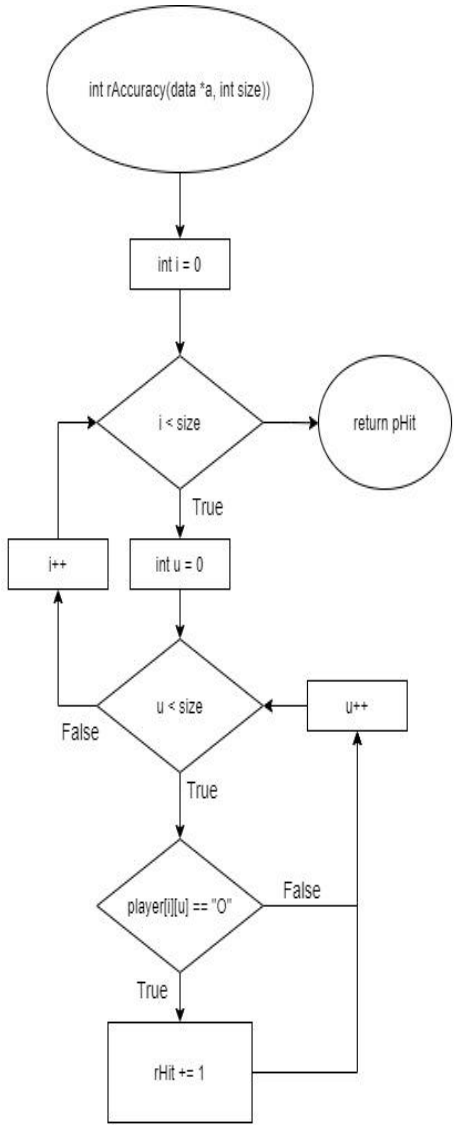
void displayHit(data *a)
compChoseY - 1]
[pChoseX - 1] != " "
True
Output
You have hit a ship
displayIpChoseY - 1][pChoseX - 1] = 'O';
rCount -= 1
False
compChoseY - 1]
[pChoseX - 1] == " "
True
Output
You missed...
displayIpChoseY - 1][pChoseX - 1] = 'X'
False
return

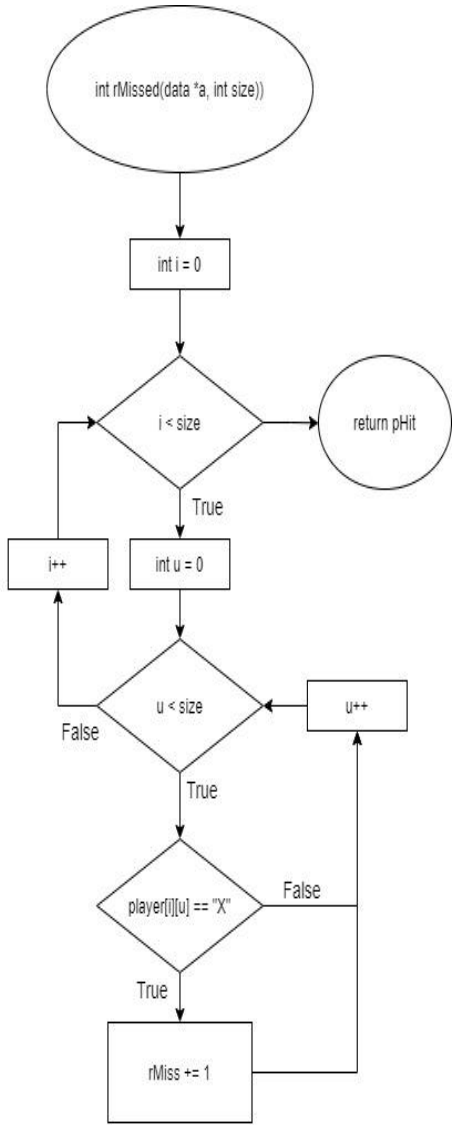
void compHit(data *a)
playerICloseY][ICloseX]
!= " "
True
Output
Computer has hit a ship
playerICloseY][ICloseX] = 'O';
pCount -= 1;
False
playerICloseY][ICloseX]
== " "
True
Output
Computer has missed...
playerICloseY][ICloseX] = 'X'
False
return

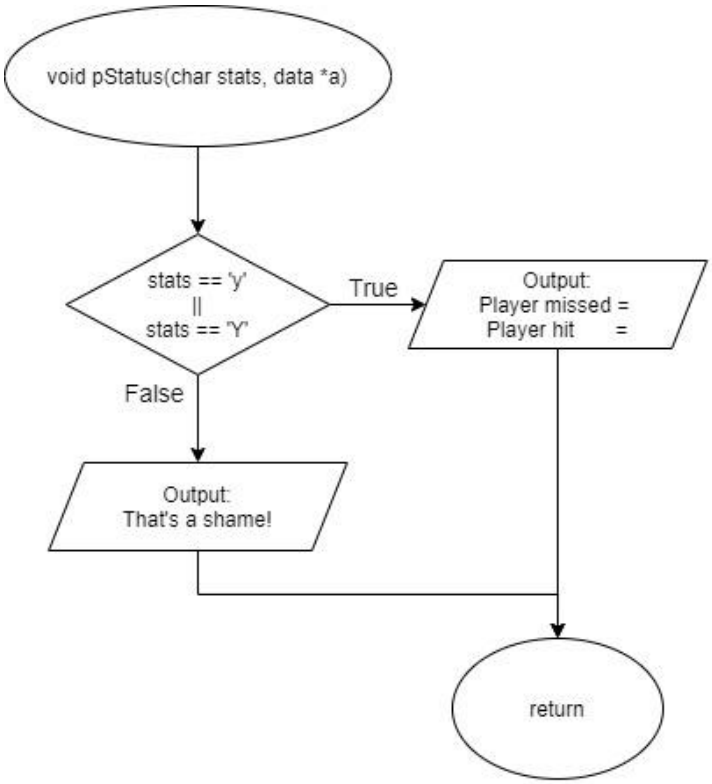
Bool choice(char play, data *a)
rCount == 0
True
Output
You have won! Congratulations
False
Output
You have lost! Better luck next time!
return true;
break;

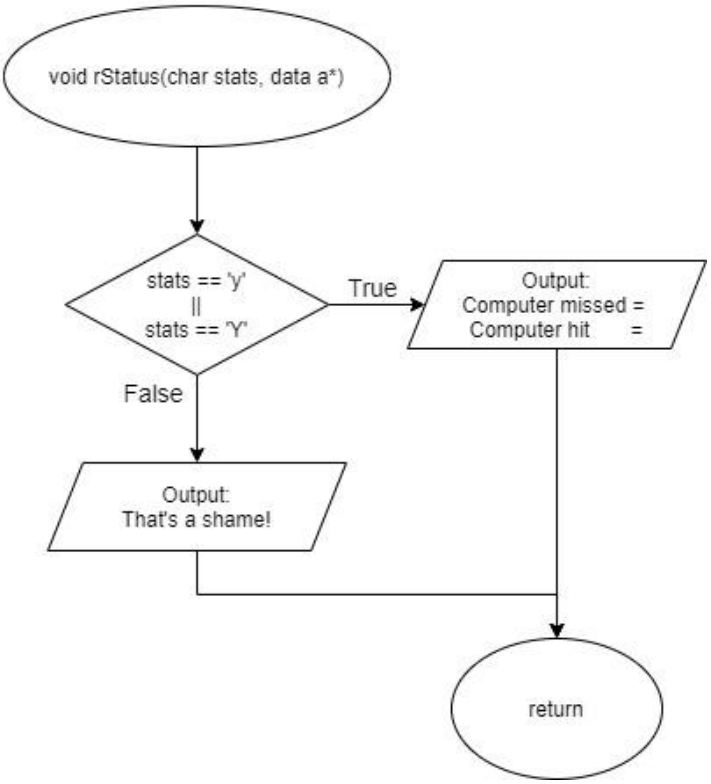

pAccuracy(data *a, int size)
int i = 0
i < size
return pHit
True
i++
int u = 0
u < size
u++
False
True
display[i][u] == "O"
False
True
pHit += 1


pMissed(data *a, int size)
int i = 0
i < size
return pHit
True
i++
int u = 0
u < size
u++
False
True
display[i][u] == "X"
False
True
pMiss += 1


int rAccuracy(data *a, int size))
int i = 0
i < size
return pHit
True
i++
int u = 0
u < size
u++
False
True
player[i][u] == "O"
False
True
rHit += 1


int rMissed(data *a, int size))
int i = 0
i < size
return pHit
True
i++
int u = 0
u < size
u++
False
True
player[i][u] == "X"
False
True
rMiss += 1


void pStatus(char stats, data *a)
stats == 'y'
||
stats == 'Y'
True
Output:
Player missed =
Player hit =
False
Output:
That's a shame!
return


void rStatus(char stats, data a*)
stats == 'y'
||
stats == 'Y'
True
Output:
Computer missed =
Computer hit =
False
Output:
That's a shame!
return

Check-Off Sheet for Project 1

Chapter 9:

Topic	Frequency	Location
Pointer Variables	1	Lines 56
Arrays/Pointers	1	Line 56
Function Parameters	13	Lines 33-46
Memory Allocation	9	Lines 56, 140-151
Return Parameters	1	Line 84

Chapter 10:

Topic	Frequency	Location
C-Strings	3	In data.h structure
Strings	1	Line 59

Chapter 11:

Topic	Frequency	Location
Structured Arrays	5	Line 56, and in data.h file
Nested Structures	1	data.h
Function Argument	13	Lines 33-46
Function Return	1	Line 84
Pointers	4	In data.h file

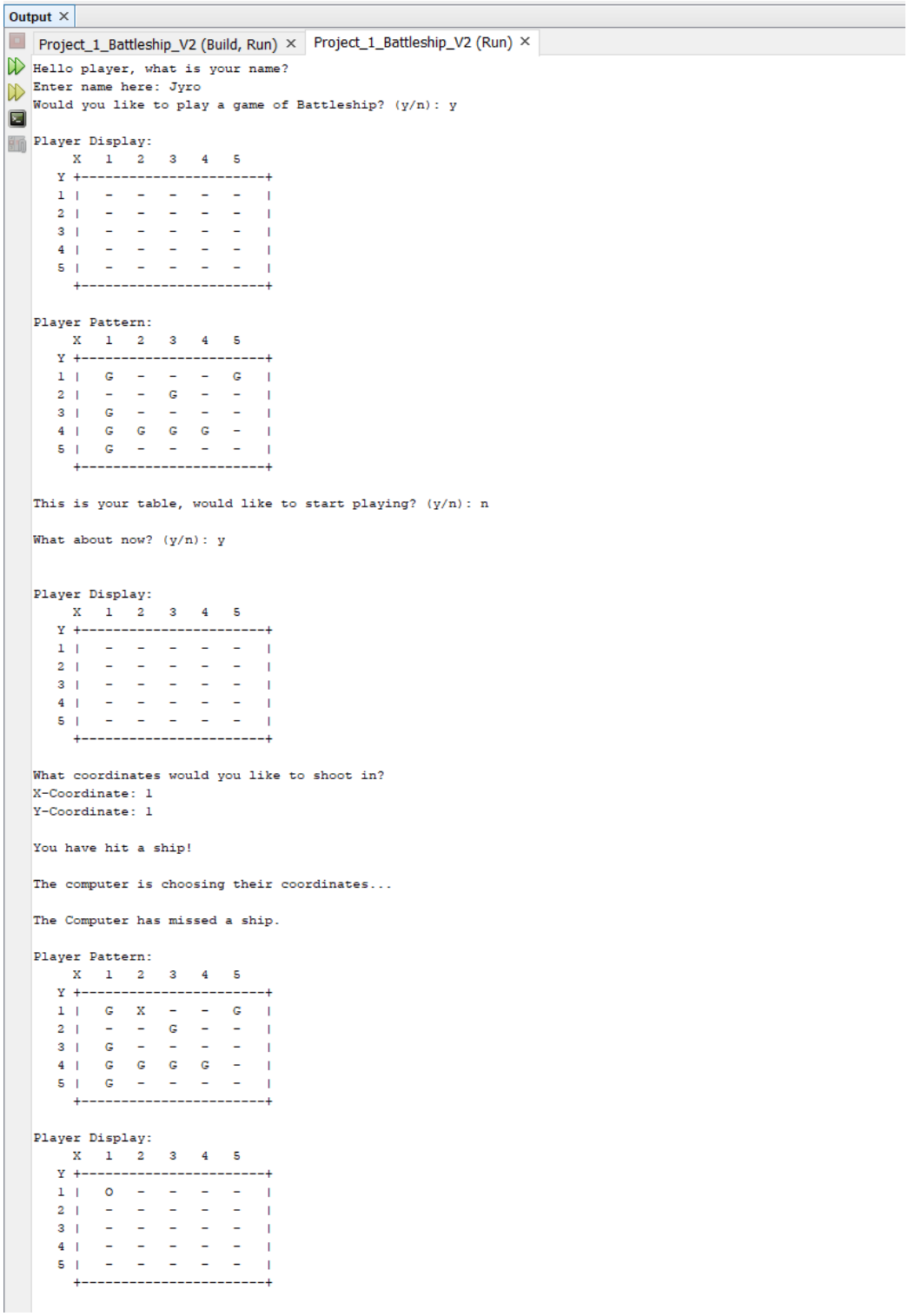
Chapter 12:

Topic	Frequency	Location
Formatting	1	Lines 16
Function Parameters	3	Lines 37, 43, 44
Member Functions	1	Line 71
Multiple Files	3	Lines 63-65
Binary Files	2	Lines 63 and 64
Records with Structures	5	Lines 63-65 and 392
Input to File	2	Lines 391-421

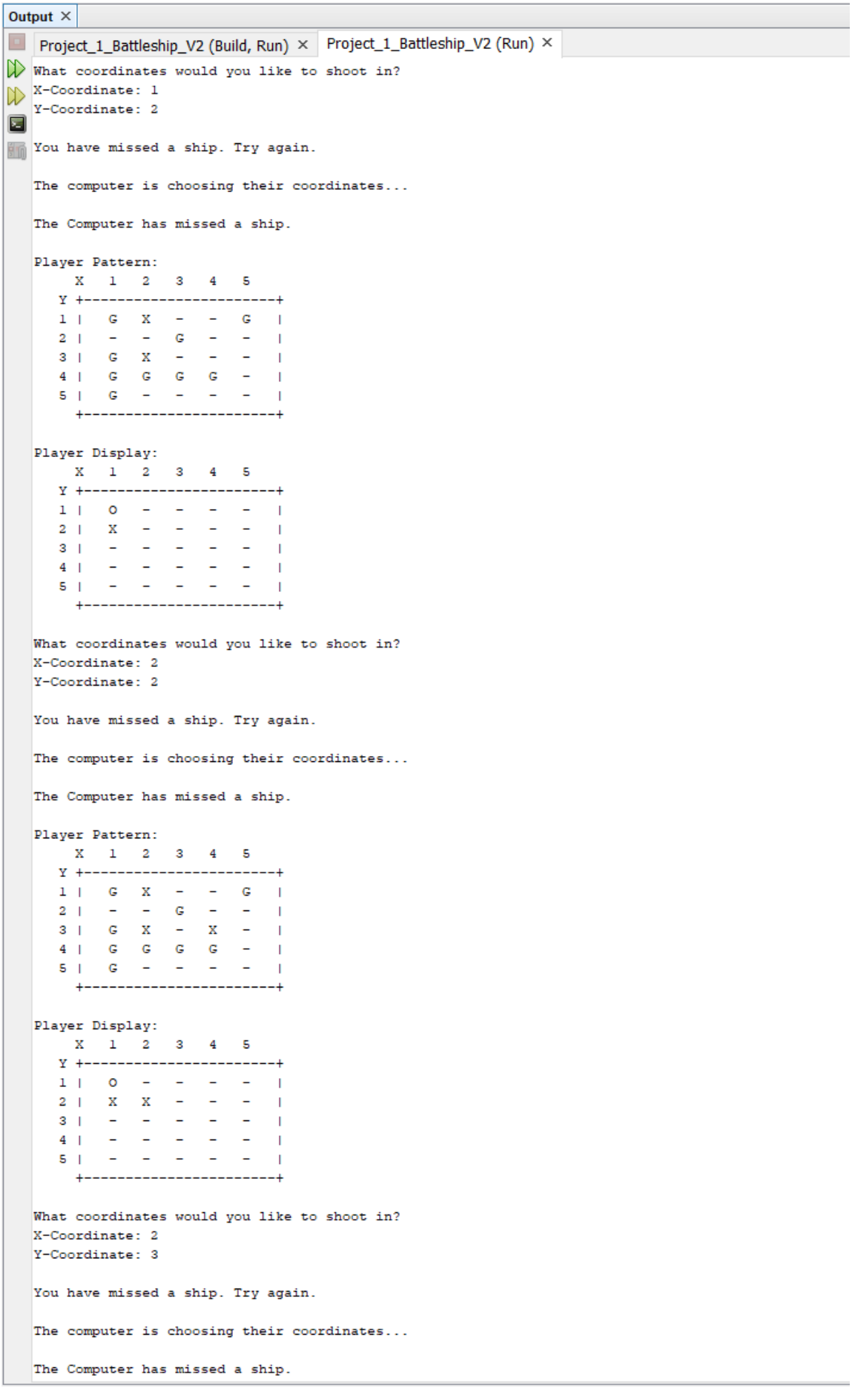
References

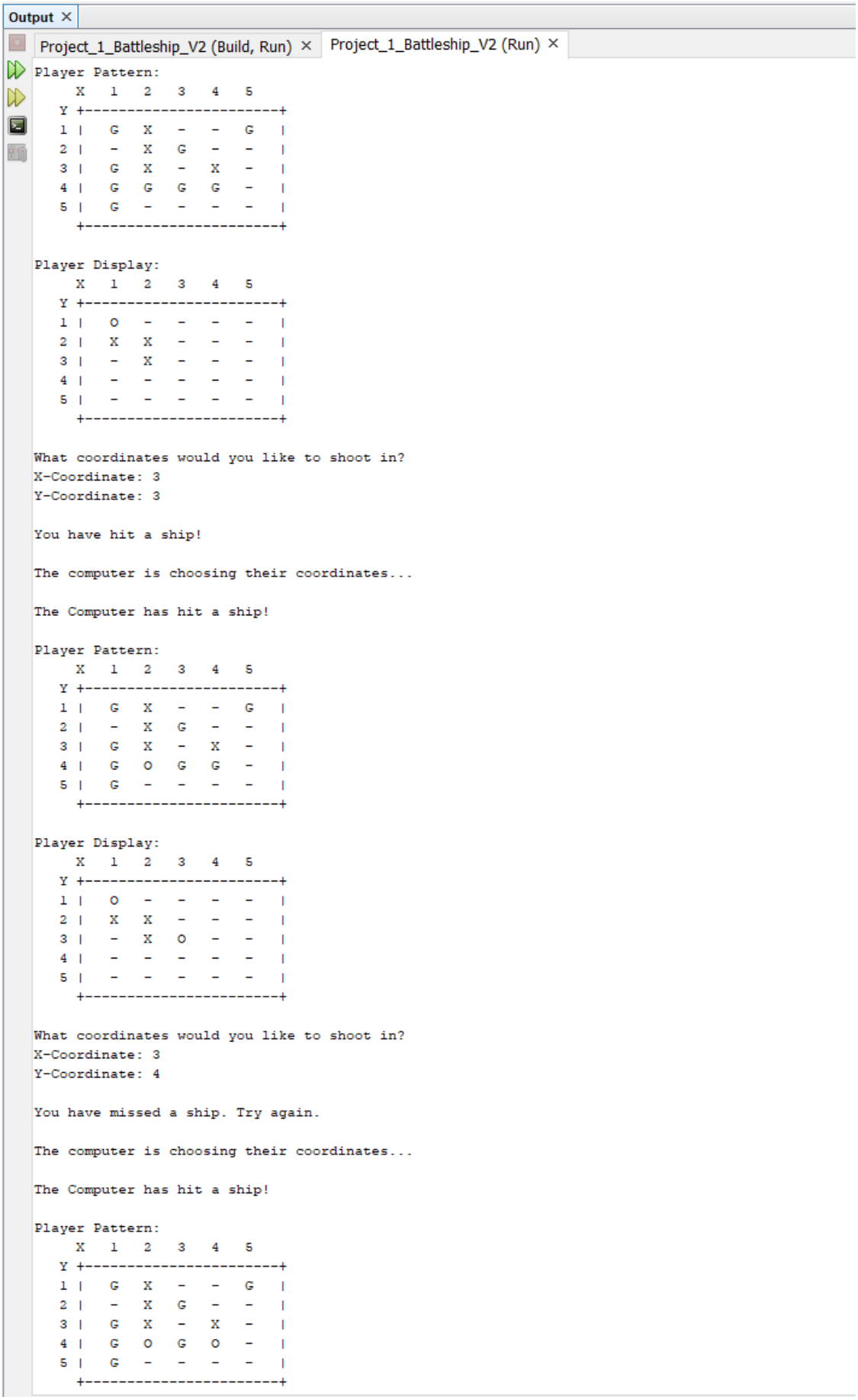
1. Dr. Lehr's Lectures
2. Class Homeworks
3. Class Github
4. Class books (Gaddis and Savitch)
5. https://www.w3schools.com/cpp/cpp_break.asp (Clarify the use of operators)

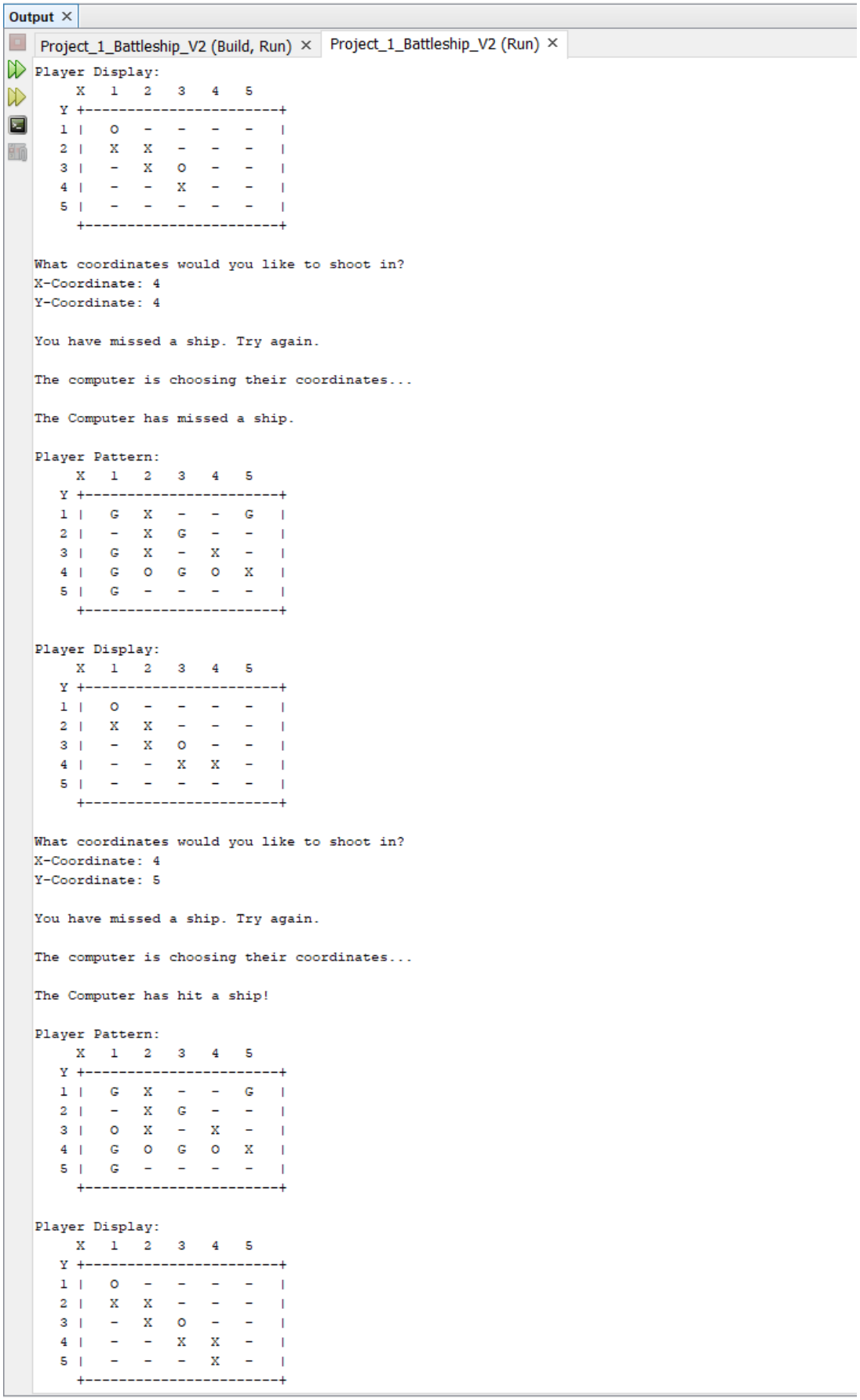
Example Output

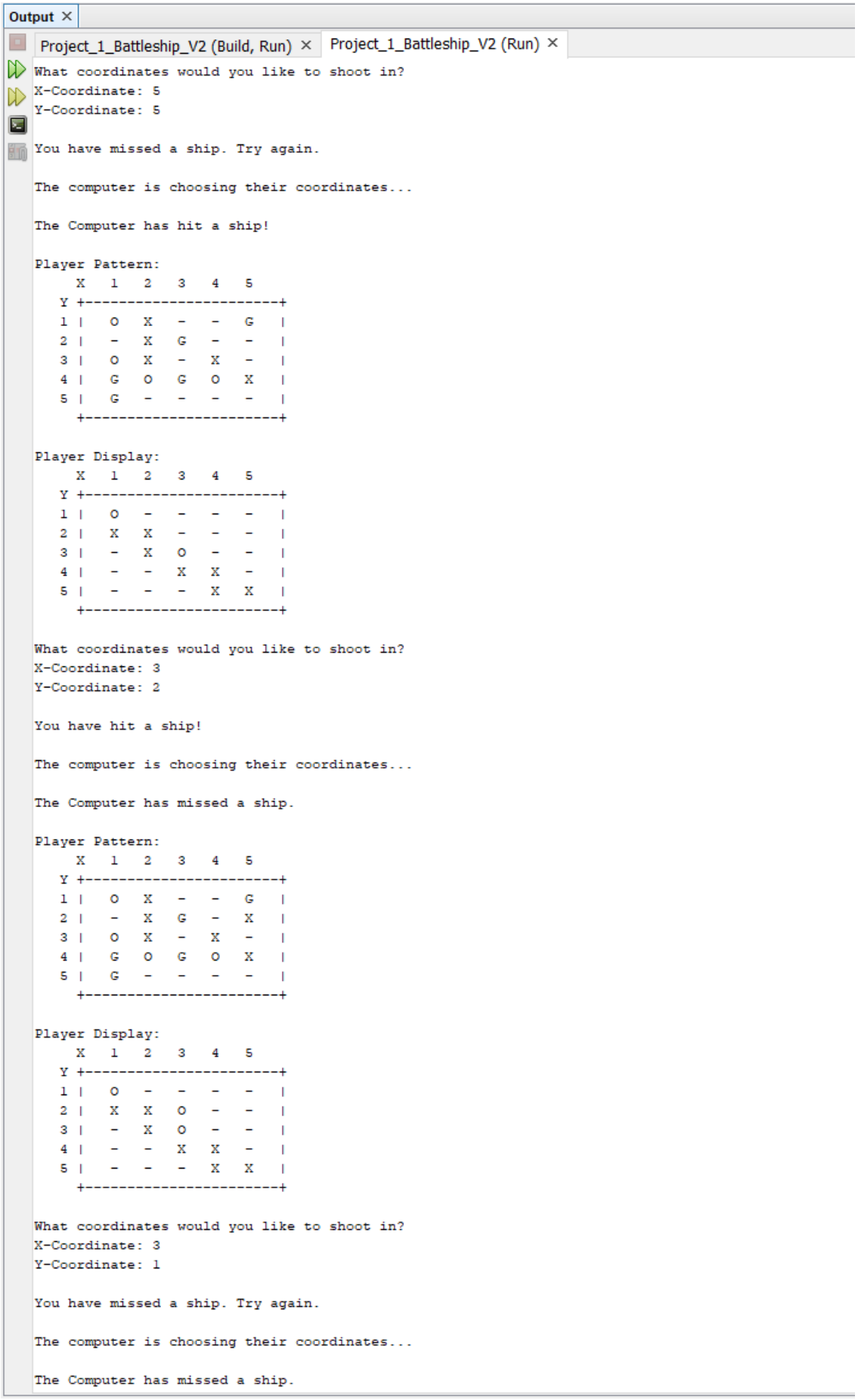


```
Output
Project_1_Battleship_V2 (Build, Run) × Project_1_Battleship_V2 (Run) ×
Hello player, what is your name?
Enter name here: Jyro
Would you like to play a game of Battleship? (y/n): y

Player Display:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   -   -   -   -   -   |
   2 |   -   -   -   -   -   |
   3 |   -   -   -   -   -   |
   4 |   -   -   -   -   -   |
   5 |   -   -   -   -   -   |
     +-----------------------+

Player Pattern:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   G   -   -   -   G   |
   2 |   -   -   G   -   -   |
   3 |   G   -   -   -   -   |
   4 |   G   G   G   G   -   |
   5 |   G   -   -   -   -   |
     +-----------------------+

This is your table, would like to start playing? (y/n): n

What about now? (y/n): y


Player Display:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   -   -   -   -   -   |
   2 |   -   -   -   -   -   |
   3 |   -   -   -   -   -   |
   4 |   -   -   -   -   -   |
   5 |   -   -   -   -   -   |
     +-----------------------+

What coordinates would you like to shoot in?
X-Coordinate: 1
Y-Coordinate: 1

You have hit a ship!

The computer is choosing their coordinates...

The Computer has missed a ship.

Player Pattern:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   G   X   -   -   G   |
   2 |   -   -   G   -   -   |
   3 |   G   -   -   -   -   |
   4 |   G   G   G   G   -   |
   5 |   G   -   -   -   -   |
     +-----------------------+

Player Display:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   -   -   -   -   |
   2 |   -   -   -   -   -   |
   3 |   -   -   -   -   -   |
   4 |   -   -   -   -   -   |
   5 |   -   -   -   -   -   |
     +-----------------------+
```

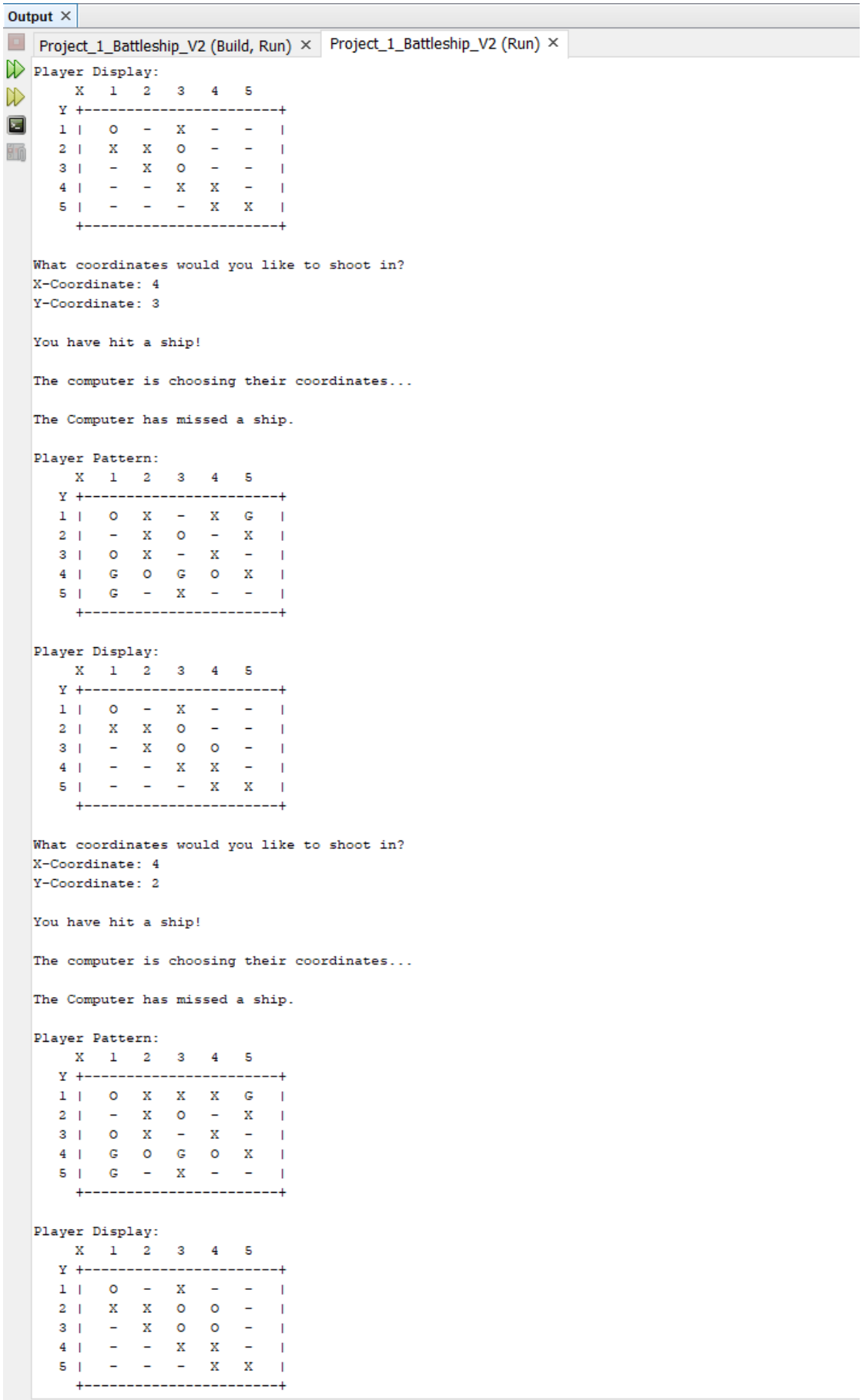

Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
What coordinates would you like to shoot in?
X-Coordinate: 1
Y-Coordinate: 2
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - - G - - |
3 | G X - - - |
4 | G G G G - |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X - - - - |
3 | - - - - - |
4 | - - - - - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 2
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - - G - - |
3 | G X - X - |
4 | G G G G - |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - - - - - |
4 | - - - - - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 3
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.

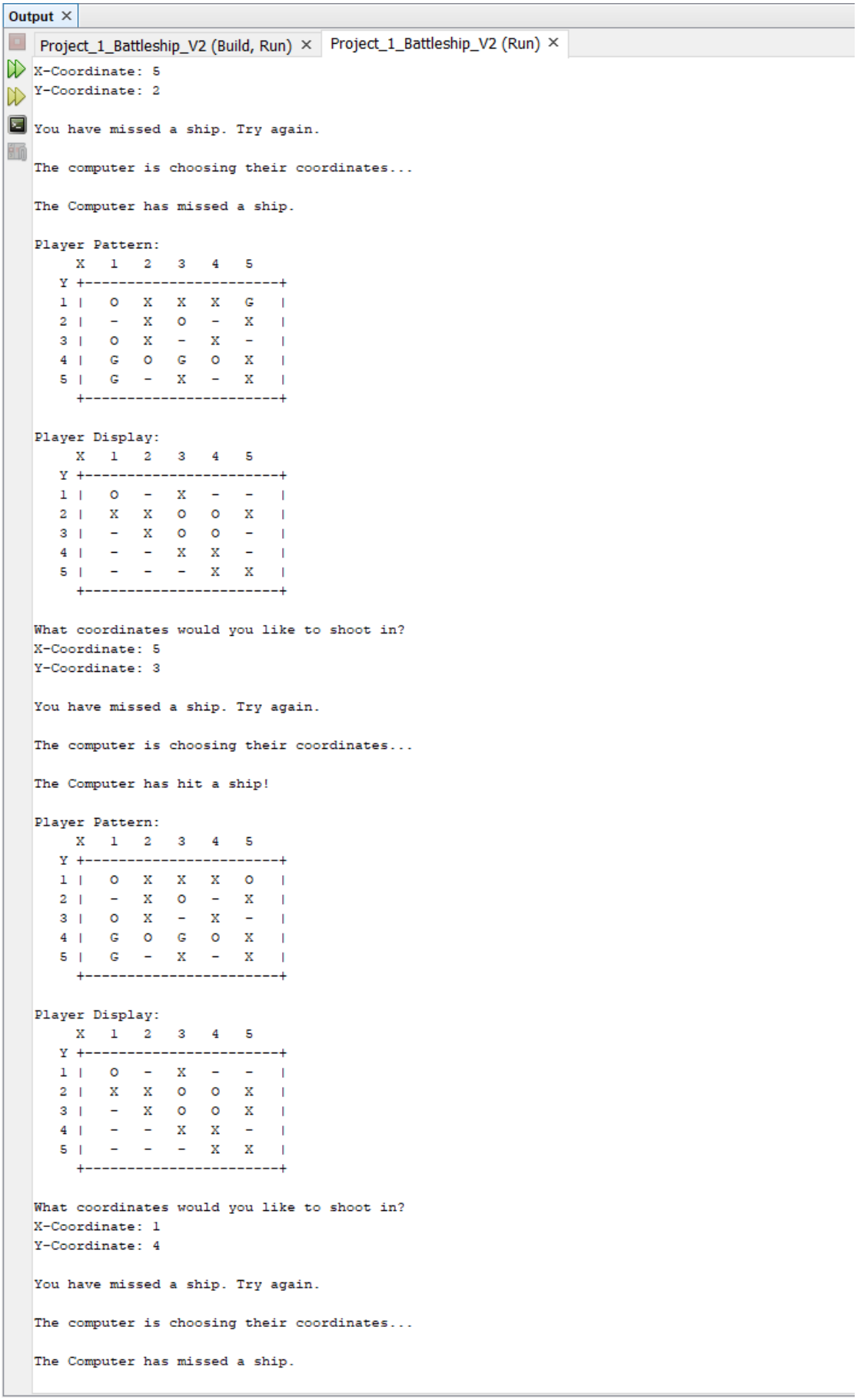

Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - X G - - |
3 | G X - X - |
4 | G G G G - |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X - - - |
4 | - - - - - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 3
You have hit a ship!
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - X G - - |
3 | G X - X - |
4 | G O G G - |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X O - - |
4 | - - - - - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 4
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - X G - - |
3 | G X - X - |
4 | G O G O - |
5 | G - - - - |
+----------+


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X O - - |
4 | - - X - - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 4
Y-Coordinate: 4
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - X G - - |
3 | G X - X - |
4 | G O G O X |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - - - |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 4
Y-Coordinate: 5
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | G X - - G |
2 | - X G - - |
3 | O X - X - |
4 | G O G O X |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X - |
+----------+


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
What coordinates would you like to shoot in?
X-Coordinate: 5
Y-Coordinate: 5
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - - G |
2 | - X G - - |
3 | O X - X - |
4 | G O G O X |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X - - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 2
You have hit a ship!
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - - G |
2 | - X G - X |
3 | O X - X - |
4 | G O G O X |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X O - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 1
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - - G |
2 | - X G - X |
3 | O X - X - |
4 | G O G O X |
5 | G - - - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - - - - |
2 | X X O - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 1
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - - G |
2 | - X G - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 4
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - - G |
2 | - X O - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - - |
+----------+

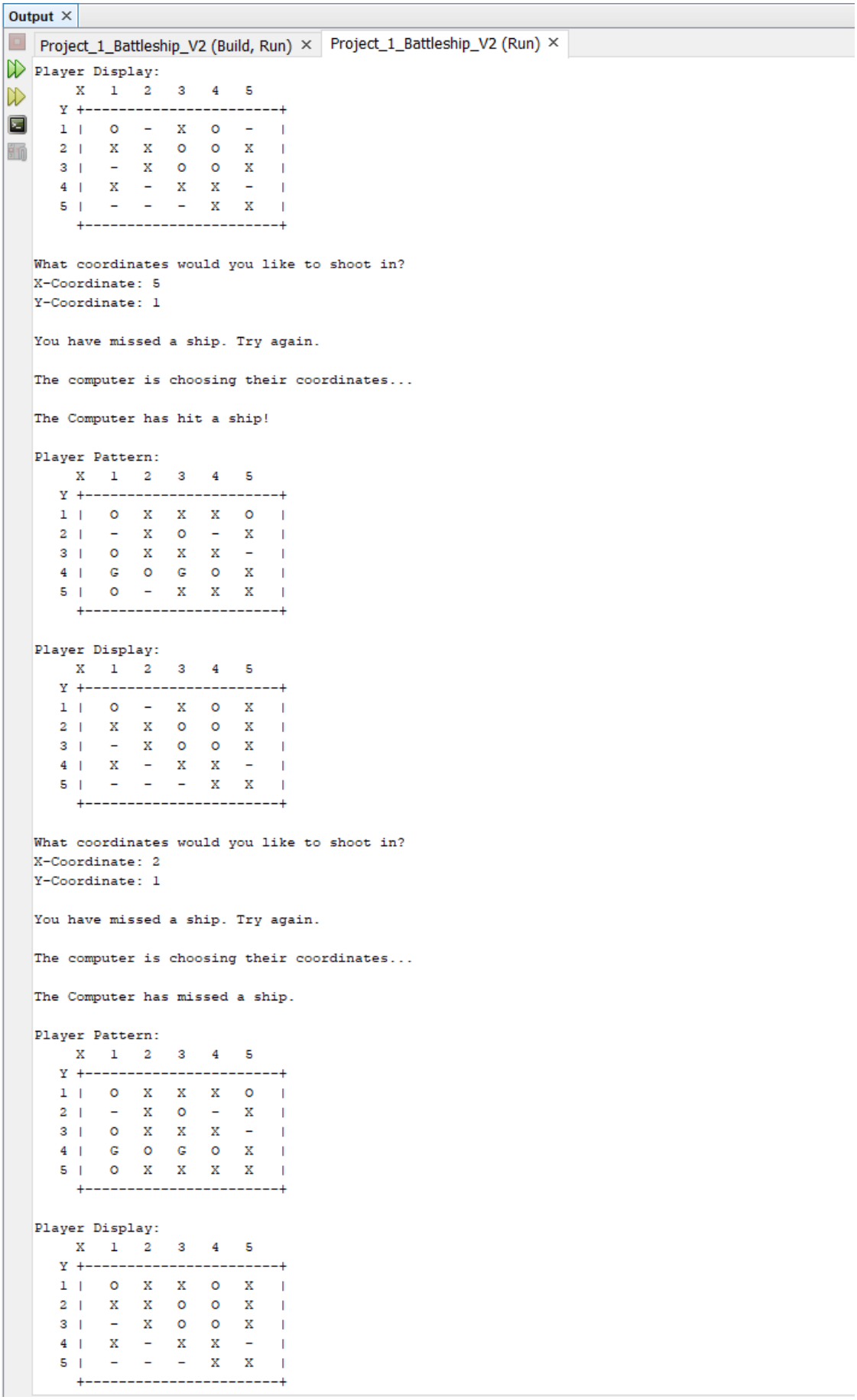

Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O - - |
3 | - X O - - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 4
Y-Coordinate: 3
You have hit a ship!
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X - X G |
2 | - X O - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O - - |
3 | - X O O - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 4
Y-Coordinate: 2
You have hit a ship!
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X G |
2 | - X O - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - - |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O O - |
3 | - X O O - |
4 | - - X X - |
5 | - - - X X |
+----------+

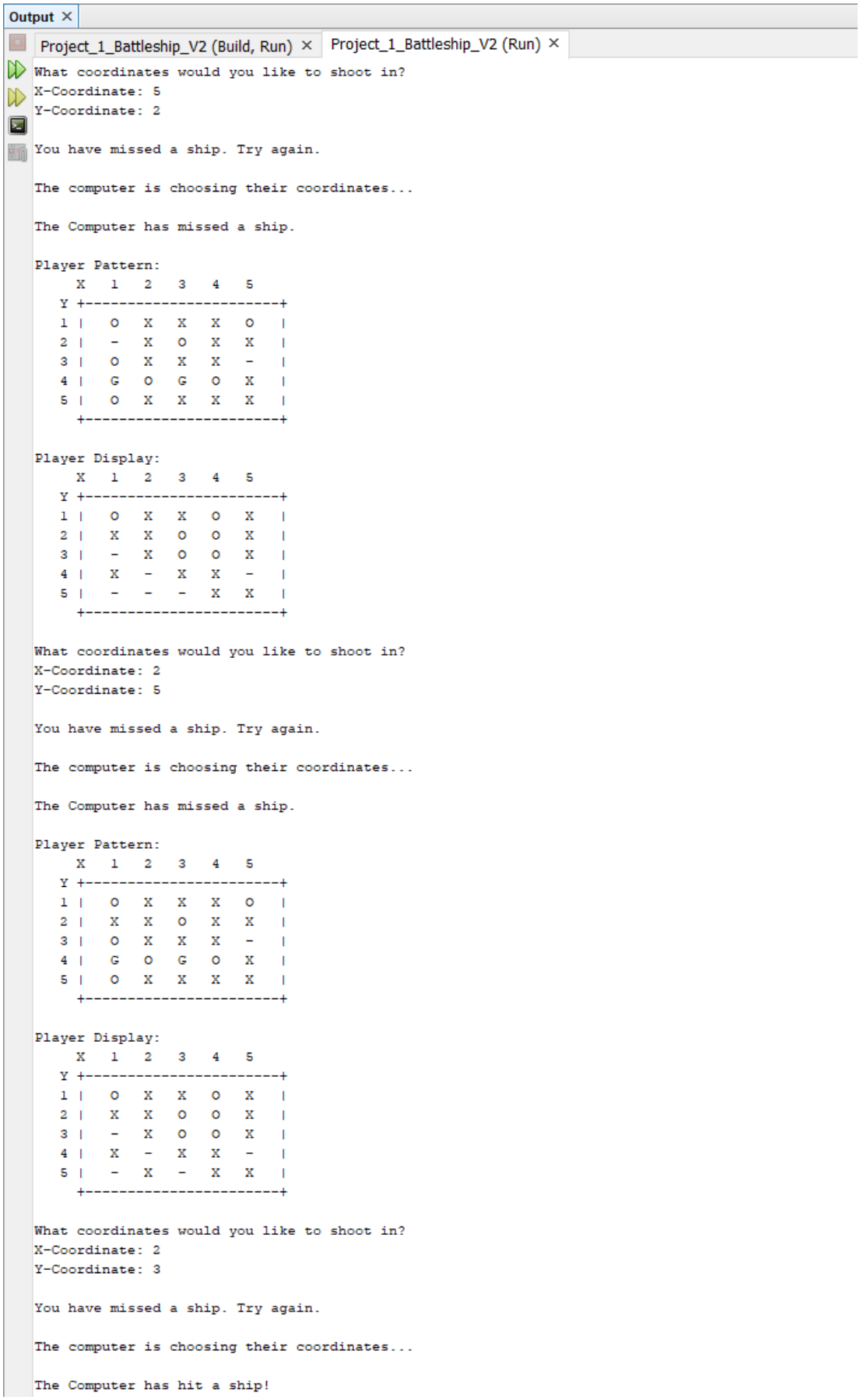

Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
X-Coordinate: 5
Y-Coordinate: 2
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X G |
2 | - X O - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O O X |
3 | - X O O - |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 5
Y-Coordinate: 3
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | - X O - X |
3 | O X - X - |
4 | G O G O X |
5 | G - X - X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X - - |
2 | X X O O X |
3 | - X O O X |
4 | - - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 1
Y-Coordinate: 4
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.

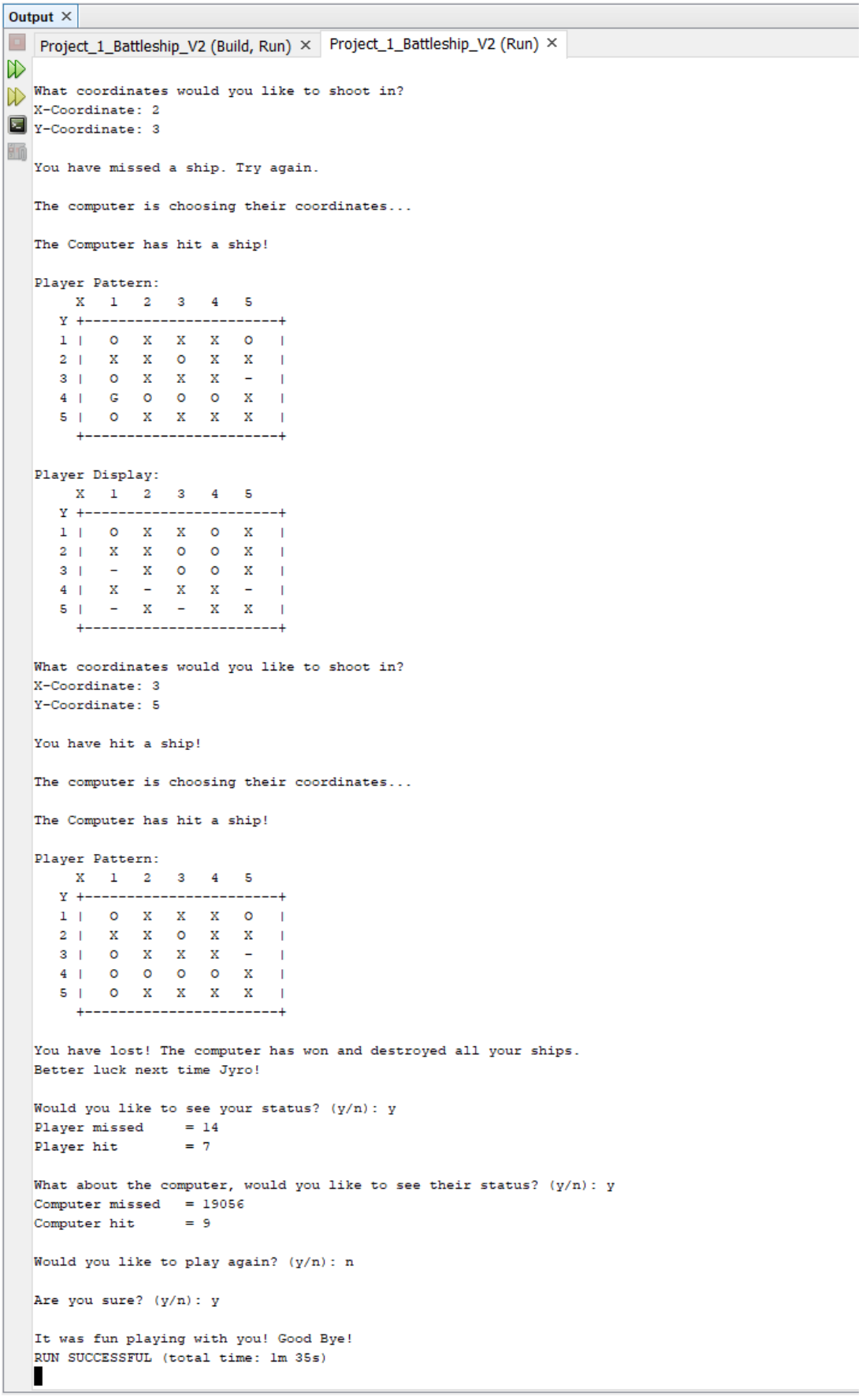
Output

Project_1_Battleship_V2 (Build, Run) | Project_1_Battleship_V2 (Run)

```
Player Pattern:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   X   X   X   O   |
   2 |   -   X   O   -   X   |
   3 |   O   X   -   X   -   |
   4 |   G   O   G   O   X   |
   5 |   G   -   X   -   X   |
     +-----------------------+

Player Display:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   -   X   -   -   |
   2 |   X   X   O   O   X   |
   3 |   -   X   O   O   X   |
   4 |   -   -   X   X   -   |
   5 |   -   -   -   X   X   |
     +-----------------------+

What coordinates would you like to shoot in?
X-Coordinate: 1
Y-Coordinate: 4

You have missed a ship. Try again.

The computer is choosing their coordinates...

The Computer has missed a ship.

Player Pattern:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   X   X   X   O   |
   2 |   -   X   O   -   X   |
   3 |   O   X   -   X   -   |
   4 |   G   O   G   O   X   |
   5 |   G   -   X   X   X   |
     +-----------------------+

Player Display:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   -   X   -   -   |
   2 |   X   X   O   O   X   |
   3 |   -   X   O   O   X   |
   4 |   X   -   X   X   -   |
   5 |   -   -   -   X   X   |
     +-----------------------+

What coordinates would you like to shoot in?
X-Coordinate: 4
Y-Coordinate: 1

You have hit a ship!

The computer is choosing their coordinates...

The Computer has missed a ship.

Player Pattern:
     X   1   2   3   4   5
   Y +-----------------------+
   1 |   O   X   X   X   O   |
   2 |   -   X   O   -   X   |
   3 |   O   X   X   X   -   |
   4 |   G   O   G   O   X   |
   5 |   G   -   X   X   X   |
     +-----------------------+
```


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X O - |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 5
Y-Coordinate: 1
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | - X O - X |
3 | O X X X - |
4 | G O G O X |
5 | O - X X X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O - X O X |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 1
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | - X O - X |
3 | O X X X - |
4 | G O G O X |
5 | O X X X X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O X X O X |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - - - X X |
+----------+


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
What coordinates would you like to shoot in?
X-Coordinate: 5
Y-Coordinate: 2
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | - X O X X |
3 | O X X X - |
4 | G O G O X |
5 | O X X X X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O X X O X |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - - - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 5
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has missed a ship.
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | X X O X X |
3 | O X X X - |
4 | G O G O X |
5 | O X X X X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O X X O X |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - X - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 3
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!


Output
Project_1_Battleship_V2 (Build, Run)
Project_1_Battleship_V2 (Run)
What coordinates would you like to shoot in?
X-Coordinate: 2
Y-Coordinate: 3
You have missed a ship. Try again.
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | X X O X X |
3 | O X X X - |
4 | G O O O X |
5 | O X X X X |
+----------+
Player Display:
X 1 2 3 4 5
Y +----------+
1 | O X X O X |
2 | X X O O X |
3 | - X O O X |
4 | X - X X - |
5 | - X - X X |
+----------+
What coordinates would you like to shoot in?
X-Coordinate: 3
Y-Coordinate: 5
You have hit a ship!
The computer is choosing their coordinates...
The Computer has hit a ship!
Player Pattern:
X 1 2 3 4 5
Y +----------+
1 | O X X X O |
2 | X X O X X |
3 | O X X X - |
4 | O O O O X |
5 | O X X X X |
+----------+
You have lost! The computer has won and destroyed all your ships.
Better luck next time Jyro!
Would you like to see your status? (y/n): y
Player missed = 14
Player hit = 7
What about the computer, would you like to see their status? (y/n): y
Computer missed = 19056
Computer hit = 9
Would you like to play again? (y/n): n
Are you sure? (y/n): y
It was fun playing with you! Good Bye!
RUN SUCCESSFUL (total time: 1m 35s)

Program

```
//System Libraries Here
#include <iostream>
#include <iomanip>
#include <cmath>
#include <cstdlib>
#include <ctime>
#include <fstream>
using namespace std;

//User Libraries Here
#include "data.h"

//Global Constants Only, No Global Variables
// Columns in the 2-D Arrays

//Like PI, e, Gravity, or conversions

//Function Prototypes Here
bool choice(char);
data *reset(int);
void prntDisplay(data *, int);
void plyrPatrn(data *, int);
void askAgain(char &);
void game(data *, int, char, char, string, fstream &, fstream &);
void results(string, data *);
int pMissed(data *, int);
int pAccuracy(data *, int);
int rMissed(data *, int);
int rAccuracy(data *, int);
void wrtTxt(fstream &, data *, int);
void wrtBin(fstream &, data *, int);
void pStatus(char, data *);
void rStatus(char, data *);
void playAgain(char &);
bool verify(char);
```

```
//Program Execution Begins Here
int main(int argc, char** argv) {
   //Set Random Number seed
   srand(static_cast<unsigned int>(time(0)));

   //Declare Variables
   data *var;
   int size = 5;
   char stats, play;
   string name;
   bool again;

   //Declare fstream variables
   fstream txtFile,binFile;
   txtFile.open("data.txt",ios::in|ios::out);
   binFile.open("data.bin",ios::in|ios::out|ios::binary);

   // Ask if player wants to play
   cout << "Hello player, what is your name?" << endl
      << "Enter name here: ";
   cin.clear();
   getline(cin, name);

   //Ask If they want to play Battleship
   cout << "Would you like to play a game of Battleship? (y/n): ";
   cin >> play;

   // initialize bool operator
   again = choice(play);


   // start game in loop with bool operator
   while(again == true) {
      // reset display table and initialize dynamic structure
      var = reset(size);

      //Display player's board
      prntDisplay(var, size);
```

```
    plyrPatrn(var, size);

    // Ask User if they want to start playing
    cout << endl << "This is your table, would like to start playing? (y/n): ";
    cin >> play;

    // Loop just in case they say no
    askAgain(play);

    //Loop Where user wants to hit and output results
    game(var, size, play, stats, name, txtFile, binFile);

    // Give player choice to play again or not
    playAgain(play);

    // Verify player choice
    again = verify(play);

    //Deallocate Memory
    delete [] var->arrays->display;
    delete [] var->arrays->player;
    delete [] var->arrays->comp;
    delete [] var->arrays;
    delete [] var;
  }

  cout << endl << "It was fun playing with you! Good Bye!";

  //Exit
  return 0;
}

bool choice(char play) {
  switch(play) {
    case 'y':
      return true;
      break;
    case 'n':
      return false;
```

```
        break;
      case 'Y':
        return true;
        break;
      case 'N':
        return false;
        break;
      default:
        return true;
    }
}

data *reset(int size) {
  //Allocate Memory
  data *a = new data;
  a->arrays = new board;
  //Initialize
  int x, y;
  x = y = 0;
  a->nBoats = a->pCount = a->rCount = 9;
  a->pMiss = a->pHit = a->rMiss = a->rHit = 0;
  //Allocate memory for 2 dim-arrays
  a->arrays->display = new char*[size];
  a->arrays->player = new char*[size];
  a->arrays->comp = new char*[size];
  //Allocate the memory for each row
  for(int row = 0; row < size; row++) {
    a->arrays->display[row] = new char[size];
  }
  for(int row = 0; row < size; row++) {
    a->arrays->player[row] = new char[size];
  }
  for(int row = 0; row < size; row++) {
    a->arrays->comp[row] = new char[size];
  }
  //Initialize Arrays with "-" to fill with ships
  for(int rows = 0; rows < size; rows++) {
    for(int cols = 0; cols < size; cols++) {
      a->arrays->display[rows][cols] = '-';
```

```
    }
  }
  for(int rows = 0; rows < size; rows++) {
    for(int cols = 0; cols < size; cols++) {
      a->arrays->player[rows][cols] = '-';
    }
  }
  for(int rows = 0; rows < size; rows++) {
    for(int cols = 0; cols < size; cols++) {
      a->arrays->comp[rows][cols] = '-';
    }
  }
  //Initialize the patterns for the player
  for(int boats = 0; boats < a->nBoats; boats++) {
    x = rand() % 5;
    y = rand() % 5;
    while(a->arrays->player[x][y] == 'G') {
      x = rand() % 5;
      y = rand() % 5;
    }
    a->arrays->player[x][y] = 'G';
  }
  //Initialize the patterns for the computer
  for(int boats = 0; boats < a->nBoats; boats++) {
    x = rand() % 5;
    y = rand() % 5;
    while(a->arrays->comp[x][y] == 'G') {
      x = rand() % 5;
      y = rand() % 5;
    }
    a->arrays->comp[x][y] = 'G';
  }
  //Return data
  return a;
}

void prntDisplay(data *a, int size) {
  cout << endl << "Player Display:" << endl;
  cout << "    X  1  2  3  4  5" << endl;
```

```
  cout << "   Y +----------------------+" << endl;
  for(int rows = 0; rows < size; rows++) {
    cout << "   " << rows + 1 << " |   ";
    for(int cols = 0; cols < size; cols++) {
      cout << a->arrays->display[rows][cols] << "   ";
    }
    cout << "|" << endl;
    if(rows == size - 1) {
      cout << "    +----------------------+" << endl;
    }
  }
}

void plyrPatrn(data *a, int size) {
  cout << endl << "Player Pattern:" << endl;
  cout << "    X  1  2  3  4  5" << endl;
  cout << "   Y +----------------------+" << endl;
  for(int rows = 0; rows < size; rows++) {
    cout << "   " << rows + 1 << " |   ";
    for(int cols = 0; cols < size; cols++) {
      cout << a->arrays->player[rows][cols] << "   ";
    }
    cout << "|" << endl;
    if(rows == size - 1) {
      cout << "    +----------------------+" << endl;
    }
  }
}

void askAgain(char &play) {
  while(play == 'n' || play == 'N') {
    cout << endl << "What about now? (y/n): ";
    cin >> play;
    cout << endl;
  }
}

void game(data *a, int size, char play, char stats, string name, fstream &txtFile, fstream
&binFile) {
```

```
  while(play == 'y' || play == 'Y') {
    // Display player's choices
    prntDisplay(a, size);

    // Ask player where they would like to hit
    cout << endl << "What coordinates would you like to shoot in?" << endl;
    cout << "X-Coordinate: ";
    cin >> a->pChoseX;
    cout << "Y-Coordinate: ";
    cin >> a->pChoseY;
    // Validate User input if choice is invalid
    if(a->pChoseX > 5 || a->pChoseY > 5 || a->pChoseX <= 0 || a->pChoseY <= 0 ||
a->arrays->comp[a->pChoseY - 1][a->pChoseX - 1] == 'O' || a->arrays->comp[a->pChoseY
- 1][a->pChoseX - 1] == 'X') {
      do {
      cout << endl << "These are invalid numbers try again..." << endl;
      cout << "X-Coordinate: ";
      cin >> a->pChoseX;
      cout << "Y-Coordinate: ";
      cin >> a->pChoseY;
      } while (a->pChoseX > 5 || a->pChoseY > 5 || a->pChoseX <= 0 || a->pChoseY <= 0 ||
a->arrays->comp[a->pChoseY - 1][a->pChoseX - 1] == 'O' || a->arrays->comp[a->pChoseY
- 1][a->pChoseX - 1] == 'X');
    }

    // See if player hit a ship or not
    if(a->arrays->comp[a->pChoseY - 1][a->pChoseX - 1] != '-') {
      cout << endl << "You have hit a ship!" << endl;
      a->arrays->display[a->pChoseY - 1][a->pChoseX - 1] = 'O';
      a->rCount -= 1;
    }
    else if(a->arrays->comp[a->pChoseY - 1][a->pChoseX - 1] == '-') {
      cout << endl << "You have missed a ship. Try again." << endl;
      a->arrays->display[a->pChoseY - 1][a->pChoseX - 1] = 'X';
    }

    // Output that computer is choosing
    cout << endl << "The computer is choosing their coordinates..." << endl;
```

```
    // use random and linear search to see if computer chose that number already;
    a->rChoseX = rand() % 5;
    a->rChoseY = rand() % 5;
    //Loop so it doesn't choose the same coordinate again
    while(a->arrays->player[a->rChoseY][a->rChoseX] == 'X' ||
a->arrays->player[a->rChoseY][a->rChoseX] == 'O') {
      a->rChoseX = rand() % 5;
      a->rChoseY = rand() % 5;
    }

    // See if Computer hit a ship or not
    if(a->arrays->player[a->rChoseY][a->rChoseX] != '-') {
      cout << endl << "The Computer has hit a ship!" << endl;
      a->arrays->player[a->rChoseY][a->rChoseX] = 'O';
      a->pCount -= 1;
    }
    else if(a->arrays->player[a->rChoseY][a->rChoseX] == '-') {
      cout << endl << "The Computer has missed a ship." << endl;
      a->arrays->player[a->rChoseY][a->rChoseX] = 'X';
    }

    // Output player's table
    plyrPatrn(a, size);

    if(a->rCount == 0 || a->pCount == 0) {
      play = 'n';
    }

  }

  // Output Results
  results(name, a);

  // Use linear search to determine how much you missed and how much you hit.
  // Calculate player and computer's accuracy
  a->pMiss = pMissed(a, size);
  a->pHit = pAccuracy(a, size);
  a->rMiss = rMissed(a, size);
  a->rHit = rAccuracy(a, size);
```

```
  //Write Both results into binary and txt
  wrtBin(binFile, a, size);
  wrtTxt(txtFile, a, size);

  // Ask if they would like to know their accuracy
  cout << "Would you like to see your status? (y/n): ";
  cin >> stats;
  pStatus(stats, a);

  // Ask if they would like to know computer's accuracy
  cout << endl << "What about the computer, would you like to see their status? (y/n): ";
  cin >> stats;
  rStatus(stats, a);
}

void results(string name, data *a) {
  if(a->rCount == 0) {
    cout << endl << "You have won! The computer has lost all their ships."
      << endl << "Congratulations " << name << "!" << endl << endl;
  }
  else if(a->pCount == 0){
    cout << endl << "You have lost! The computer has won and destroyed all your ships."
      << endl << "Better luck next time " << name << "!" << endl << endl;
  }
  else if(a->pCount == a->rCount){
    cout << endl << "You have tied! your ships and the computer's ships are all destroyed."
      << endl << "Better luck next time " << name << "!" << endl << endl;
  }
}

int pMissed(data *a, int size) {
  for(int i = 0; i < size; i++) {
    for(int u = 0; u < size; u++) {
      if(a->arrays->display[i][u] == 'X') {
        a->pMiss += 1;
      }
    }
  }
```

```
  return a->pMiss;
}

int pAccuracy(data *a, int size) {
  for(int i = 0; i < size; i++) {
    for(int u = 0; u < size; u++) {
      if(a->arrays->display[i][u] == 'O') {
        a->pHit += 1;
      }
    }
  }
  return a->pHit;
}

int rMissed(data *a, int size) {
  for(int i = 0; i < size; i++) {
    for(int u = 0; u < size; u++) {
      if(a->arrays->player[i][u] == 'X') {
        a->rMiss += 1;
      }
    }
  }
  return a->rMiss;
}

int rAccuracy(data *a, int size) {
  for(int i = 0; i < size; i++) {
    for(int u = 0; u < size; u++) {
      if(a->arrays->player[i][u] == 'O') {
        a->rHit += 1;
      }
    }
  }
  return a->rHit;
}

void wrtBin(fstream &out, data *a, int size){
  out.write(reinterpret_cast<char *>(a), sizeof(a));
}
```

```
void wrtTxt(fstream &out, data *a, int size){
  //output Diagrams
  out << endl << "Player Display:" << endl;
  out << "    X  1  2  3  4  5" << endl;
  out << "  Y +----------------------+" << endl;
  for(int rows = 0; rows < size; rows++) {
    out << "  " << rows + 1 << " |  ";
    for(int cols = 0; cols < size; cols++) {
      out << a->arrays->display[rows][cols] << "  ";
    }
    out << "|" << endl;
    if(rows == size - 1) {
      out << "    +----------------------+" << endl;
    }
  }

  out << endl << "Player Pattern:" << endl;
  out << "    X  1  2  3  4  5" << endl;
  out << "  Y +----------------------+" << endl;
  for(int rows = 0; rows < size; rows++) {
    out << "  " << rows + 1 << " |  ";
    for(int cols = 0; cols < size; cols++) {
      out << a->arrays->player[rows][cols] << "  ";
    }
    out << "|" << endl;
    if(rows == size - 1) {
      out << "    +----------------------+" << endl;
    }
  }
  //Output Results
  out << "Player missed    = " << a->pMiss << endl;
  out << "Player hit       = " << a->pHit << endl;
  out << "Computer missed  = " << a->rMiss << endl;
  out << "Computer hit     = " << a->rHit << endl;
}

void pStatus(char stats, data *a) {
  if(stats == 'y' || stats == 'Y') {
```

```
    cout << "Player missed    = " << a->pMiss << endl;
    cout << "Player hit       = " << a->pHit << endl;
  }
  else {
    cout << "That's a shame!" << endl;
  }
}

void rStatus(char stats, data *a) {
  if(stats == 'y' || stats == 'Y') {
    cout << "Computer missed   = " << a->rMiss << endl;
    cout << "Computer hit     = " << a->rHit << endl;
  }
  else {
    cout << "That's a shame!" << endl;
  }
}

void playAgain(char &play) {
  cout << endl << "Would you like to play again? (y/n): ";
  cin >> play;
}

bool verify(char play) {
  if(play == 'n' || play == 'N') {
    cout << endl << "Are you sure? (y/n): ";
    cin >> play;

    if(play == 'y' || play == 'Y') {
      return false;
    }
    else {
      return true;
    }
  }
  else if(play == 'y' || play == 'Y') {
    return true;
  }
}
```