```
File:   main.cpp
Author: Daniel Baca
Created on October 27th, 2022,
6:14 PM
Purpose:  Version 5 - Battleship
```

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

```
//Structure for all player information
struct Playerinfo
{
    char player_name[100];
    string ships[5];
    int ship_coors[20][20];
    string * shts_frd = new string[100];
    char player_board[10][10];
    char enemy_board[10][10];
    string ship_name[5] = {"Carrier",
    "Battleship", "Cruiser",
    "Submarine", "Destroyer"};
};
```

```
//Structure holding information needed for the array of structures
struct Players
{
    const int SIZE = 2;
    const int numshps = 5;
    Playerinfo * info;
};
```

```
//Enum format for player boards output loop
enum grid {A,B,C,D,E,F,G,H,I,J,K};
```

```
//Function Prototypes
Players *game_start();
void coorTonum(Players *);
void numToboard(Players *);
void disp_board1(Players *,ofstream &);
void disp_board2(Players *,ofstream &);
bool shooting1(Players *,int &);
bool shooting2(Players *,int &);
void prntTofile(Players*,int,ofstream &);
void prntToBinFile(Players *,fstream &);
void BinFileToprnt(Players *,fstream &);
```

main

```
//Declare all variables
struct Players * plyrs = new Players;
int count1=0;
int count2=0;
bool gme_ovr;
char choice;
```

```
//Declare the two text files
ofstream plyr_1file;
ofstream plyr_2file;
```

```
//Open the binary file
fstream file;
file.open("Players Info", ios:: in | ios::out | ios:: binary);
```

```
//Open the two text files
plyr_1file.open("Player_1");
plyr_2file.open("Player_2");
```

```
//Display and output the beginning of the board game
"----------- Battleship -----------
The classic strategy board game!!!
----- Somethings to know -----
Carrier Size (5 spaces)
Battleship Size (4 spaces)
Cruiser Size (3 spaces)
Submarine Size (3 spaces)
Destroyer Size (2 spaces)
--- Keep this in mind for when you place your ships!!! ---
--- Once the game starts and each player can shoot ---
--- you can input Q to Quit the Game whenever.     ---"
```

```
//Receive both player's starting
information
plyrs = game_start();
```

```
//Convert the user input to
usable coordinates using the
function
coorTonum(plyrs);
```

```
//Go to function to setup
coordinates to board
numToboard(plyrs);
```

a.

```
                    ( a. )
                      │
                      ▼
          ┌───────────────────────┐   False
          ◇    while(!gme_ovr)      ◇────────────────────────────►  ┌──────────────────────────────────────┐
          └───────────────────────┘                                 │  // Call function to write records    │
                      │                                              │  to binary file                       │
                      │ True                                         │  prntToBinFile(plyrs,file);           │
                      ▼                                              └──────────────────────────────────────┘
          ╱───────────────────────╱                                               │
         ╱  Output:                ╱                                               ▼
        ╱   plyrs->info[0].player_name╱                              ╱──────────────────────────────────────────╱
        ╲───────────────────────────╱                              ╱ //Output the prompt tp players to input    ╱
                      │                                            ╱  choice for Players Info                    ╱
                      ▼                                           ╱ " Would you like to view the overall         ╱
          ╱───────────────────────╱                             ╱  Players Recap? (Y/N) "                        ╱
         ╱  Output:                ╱                            ╲──────────────────────────────────────────────╱
        ╱   " its your turn!"       ╱                                              │
        ╲───────────────────────────╱                                             ▼
                      │                                              ╱──────────────────────╱
                      ▼                                             ╱  Input:                ╱
          ┌───────────────────────┐                               ╱   cin >> choice;         ╱
          │  disp_board1(plyrs,plyr_1file);│                      ╲──────────────────────────╱
          └───────────────────────┘                                              │
                      │                                                           ▼
                      ▼                          ┌──────────────────┐   ┌──────────────────────────────────────┐
          ┌───────────────────────┐             ◇ if(choice == 'Y' ||◇─►│ // Call function to print records     │
          │ gme_ovr = shooting1(plyrs, count1);│  │  choice == 'y')    │ │ from binary file                     │
          └───────────────────────┘             └──────────────────┘   │ BinFileToprnt(plyrs, file);          │
                      │                                 │   True        └──────────────────────────────────────┘
                      ▼                                 │                              │
          ┌─────────────┐   True   ┌──────────────┐    │                              ▼
          ◇ if(gme_ovr)  ◇────────►│ //Break the   │   │ False           ╱──────────────────────────╱
          └─────────────┘          │ while loop    │   ▼                ╱  Output:                   ╱
                │                   │ break;        │ ┌────────────────┐╱  "Thank you for playing!!!" ╱
                │                   └──────────────┘ ◇ else if(choice ==◇╲────────────────────────────╱
                │ False                              │  'N' ||          │      True   ╱──────────────────────────╱
                ▼                                    │  choice == 'n')  │────────────►╱  Output:                   ╱
          ╱───────────────────────╱                 └────────────────┘            ╱  "Thank you for playing!!!" ╱
         ╱  Output:                ╱                        │                      ╲────────────────────────────╱
        ╱   plyrs->info[1].player_name╱                     │ False
        ╲───────────────────────────╱                       ▼
                      │                          ┌──────────────────────────────────┐
                      ▼                          │ //Close the two text files        │
          ╱───────────────────────╱             │ plyr_1file.close();               │
         ╱  Output:                ╱             │ plyr_2file.close();               │
        ╱   " its your turn!"       ╱            └──────────────────────────────────┘
        ╲───────────────────────────╱                        │
                      │                                       ▼
                      ▼                          ┌──────────────────────────────────┐
          ┌───────────────────────┐             │ //Close the binary file           │
          │  disp_board2(plyrs,plyr_2file);│     │ file.close();                     │
          └───────────────────────┘             └──────────────────────────────────┘
                      │                                       │
                      ▼                                       ▼
          ┌───────────────────────┐             ┌──────────────────────────────────┐
          │ gme_ovr = shooting2(plyrs, count2);│ │ //Deallocate memory               │
          └───────────────────────┘             │ delete[] plyrs;                   │
                                                 │ delete[] plyrs->info->shts_frd;   │
                                                 └──────────────────────────────────┘
                                                              │
                                                              ▼
                                                     ╭──────────────────╮
                                                     │ //Exit from main  │
                                                     │ return 0;         │
                                                     ╰──────────────────╯
```

```
Players *game_start()
```

//Declare Variables
struct Players * plyrs = new Players;
plyrs->info = new Playerinfo[plyrs->SIZE];
char choice;

Output:
"Would you like to see some example player input? (Y/N) "

Input:
cin >> choice;

if(choice == 'Y' || choice == 'y') — False / True

Output:
"Name for Player: Mary Sue
Ship Coordinates -

   Carrier: A5 A9
   Battleship : F2 I2
   Cruiser: E5 G5
   Submarine: I8 I10
   Destroyer: C2 D2

-------------- Player Board ------------
   1 2 3 4 5 6 7 8 9 10

A |0|0|0|0|X|X|X|X|X|0|
B |0|0|0|0|0|0|0|0|0|0|
C |0|X|0|0|0|0|0|0|0|0|
D |0|X|0|0|0|0|0|0|0|0|
E |0|0|0|0|X|0|0|0|0|0|
F |0|X|0|0|X|0|0|0|0|0|
G |0|X|0|0|X|0|0|0|0|0|
H |0|X|0|0|0|0|0|0|0|0|
I |0|X|0|0|0|0|0|X|X|X|
J |0|0|0|0|0|0|0|0|0|0|"

//Formatting
cin.ignore(1);

Declare and initialize
In for Loop
i = 0

i++

i <plyrs->SIZE — False / True

```
return plyrs;
```

Output:
"Name for Player #"

Output:
(i + 1)

Input:
cin.getline(plyrs->info[i].player_name, 100);

Output:
plyrs->info[i].player_name

Output:
" choose your ship locations
Selecte the coordinates fro each ship (ex. input A1 A5, for the carrier)

Declare and initialize
In for Loop
j = 0

j++

j<plyrs->numshps — True / False

Output:
plyrs->info[i].ship_name[j]

Input:
getline(cin,plyrs->info[i].ships[j]);

while(plyrs->info[i].ships[j].length() > 7 || !isalpha(plyrs->info[i].ships[j][0])) — False / True

Output:
"Please follow the input format for the ship coordinates (ex. B3 B7)"

Output:
plyrs->info[i].ship_name[j]

Input:
getline(cin,plyrs->info[i].ships[j]);

```
void coorTonum()
```

Declare and initialize
In for Loop
x = 0

x++

x <plyrs->SIZE — True → Declare and initialize
In for Loop
i = 0

False

return:

i++

i <plyrs->numshps

False

True

if(plyrs->info[x].ships[i].length() == 5 || plyrs->info[x].ships[i].length() == 6) — True → Declare and initialize
In for Loop
j = 0

j++

False

//Convert string of chars to usable number coordinates
plyrs->info[x].ship_coors[i][0] = plyrs->info[x].ships[i][0] - 64;
plyrs->info[x].ship_coors[i][3] = plyrs->info[x].ships[i][4] - 64;
plyrs->info[x].ship_coors[i][2] = 0;
plyrs->info[x].ship_coors[i][1] = 10;
plyrs->info[x].ship_coors[i][4] = 10;

j <plyrs->numships

False

True

if(isalpha(plyrs->info[x].ships[i][j])) — True → plyrs->info[x].ship_coors[i][j] = plyrs->info[x].ships[i][j] - 64;

False

if(isdigit(plyrs->info[x].ships[i][j]) && plyrs->info[x].ships[i][j+1] == '0') — True → plyrs->info[x].ship_coors[i][j] = 10;
break;

False

else if(isdigit(plyrs->info[x].ships[i][j])) — True → plyrs->info[x].ship_coors[i][j] = plyrs->info[x].ships[i][j] - 48;

False

if(isspace(plyrs->info[x].ships[i][j])) — True → plyrs->info[x].ship_coors[i][j] = 0;

False

```
void numToboard()
```

Declare and initialize
In for Loop
z= 0

z++

Declare and initialize
In for Loop
i = A

i++

z <plyrs->SIZE — True

False

i < K — True

False

Declare and initialize
In for Loop
j = A

j++

j < K

False

True

plyrs->info[z].player_board[i][j] = '0';
plyrs->info[z].enemy_board[i][j] = '0';

Declare and initialize
In for Loop
z = 0

z++

z < plyrs->SIZE — True

False

Declare and initialize
In for Loop
i = 0

i++

```
return;
```

i < plyrs->numshps

False

True

Declare and initialize In for
Loop
k = (plyrs-
>info[z].ship_coors[i][1]-1)

k++

if(plyrs->info[z].ship_coors[i][0] == plyrs-
>info[z].ship_coors[i][3])

True

k<plyrs->info[z].ship_coors[i][4]

True

plyrs-
>info[z].player_board[(plyrs-
>info[z].ship_coors[i][0]-1)][k] =
'X';

False

False

Declare and initialize In for
Loop
a=(plyrs-
>info[z].ship_coors[i][0]-1)

a++

a<plyrs->info[z].ship_coors[i][3]

True

plyrs->info[z].player_board[a]
[(plyrs->info[z].ship_coors[i][1]-1)]
= 'X';

False

```
bool shooting1()
```

//Declare Variables
int num[2];
int hits = 0;
int count = 0;

Output:
plyrs->info[0].player_name

Output:
", where to shoot?"

Input:
getline(cin,plyrs->info[0].shts_frd[count1]);

if(plyrs->info[0].shts_frd[count1][0] == 'Q' || plyrs->info[0].shts_frd[count1][0] == 'q')

True → Output: "Game over!" → return true;

False

while(!isalpha(plyrs->info[0].shts_frd[count1][0])
&& !isupper(plyrs->info[0].shts_frd[count1][0])
|| !isdigit(plyrs->info[0].shts_frd[count1][1]))

False

True

Input:
getline(cin,plyrs->info[0].shts_frd[count1]);

Output:
", where to shoot?"

Output:
plyrs->info[0].player_name

Output:
"Please input a correct coordinate. (ex. A10, B7, I3)"

Declare and initialize
In for Loop
j = 0

j++

j < plyrs->SIZE

True

if(isalpha(plyrs->info[0].shts_frd[count1][j]))

False

True

num[j] = plyrs->info[0].shts_frd[count1][j] - 64;

False

b.

if(isdigit(plyrs->info[0].shts_frd[count1][j]) && plyrs->info[0].shts_frd[count1][j+1] == '0')

True → num[j] = 10;
break;

False

else if(isdigit(plyrs->info[0].shts_frd[count1][j]))

True → num[j] = plyrs->info[0].shts_frd[count1][j] - 48;

False

```
                                              ┌─────────┐
                                              │  m++    │
                                              └─────────┘
      ( b. )

 ┌──────────────────┐          ┌──────────────────┐
 │ Declare and      │          │ Declare and      │
 │ initialize       │          │ initialize       │        ┌──────┐
 │ In for Loop      │          │ In for Loop      │        │ l++  │
 │ m = 0            │          │ l = 0            │        └──────┘
 └──────────────────┘          └──────────────────┘

      ◇ m < 11 ◇   True              ◇ l < 11 ◇  True   ◇ if(l == num[1] && m == num[0]) ◇  True   ◇ if(plyrs->info[1].player_board[m-1][l-1]== 'X') ◇   True

      False                          False                        False                                       False

 plyrs->info[0].enemy_board[m-1][l-1] = 'M';        Output:
 plyrs->info[1].player_board[m-1][l-1] = 'M';       "It was a miss!"

 ( c. )   plyrs->info[0].enemy_board[m-1][l-1] = 'H';   Output:
          plyrs->info[1].player_board[m-1][l-1] = 'H';   "It was a hit!"

 ┌──────────────────┐
 │ Declare and      │
 │ initialize       │
 │ In for Loop      │
 │ z = 0            │
 └──────────────────┘

      ◇ z < 11 ◇  True   ┌──────────────────┐   ┌──────┐
                         │ Declare and      │   │ z++  │
                         │ initialize       │   └──────┘
      False              │ In for Loop      │
                         │ y = 0            │   ┌──────┐
                         └──────────────────┘   │ y++  │
                                                └──────┘

              ◇ y < 11 ◇  True
              False

 ◇ if(plyrs->info[1].player_board[z][y] == 'X') ◇  True   ┌──────┐
                                                          │hits++│
 False                                                    └──────┘

              ◇ if (hits == 0) ◇  False
              True

              Output:
              plyrs->info[0].player_name

              Output:
              " has won the game!"

              ( return true; )

              ┌──────────┐
              │ count1++:│
              └──────────┘

              ( return false; )
```

```
  ( c. ) ──────▶ ┌─────────────────────┐                    ┌─────────┐
                 │ Declare and initialize │◀───────────────  │  x++    │ ◀──────────────┐
                 │   In for Loop        │                    └─────────┘                │
                 │      x = 0           │                                                │
                 └─────────────────────┘                                                │
                          │                                                             │
                          ▼                                                             │
                   ◇ x < plyrs->numshps ◇                                               │
        False ◀───                                                                      │
                          │ True                                                        │
                          ▼                                                             │
```

if(m >= plyrs->info[1].ship_coors[x][0] && m <= plyrs->info[1].ship_coors[x][3] && l == plyrs->info[1].ship_coors[x][1] && l == plyrs->info[1].ship_coors[x][4])

False

else if(l >= plyrs->info[1].ship_coors[x][1] && l <= plyrs->info[1].ship_coors[x][4] && m == plyrs->info[1].ship_coors[x][0] && m == plyrs->info[1].ship_coors[x][3])

False

True

count = (plyrs->info[1].ship_coors[x][3]+1) - plyrs->info[1].ship_coors[x][0];

True

count = (plyrs->info[1].ship_coors[x][4]+1) - plyrs->info[1].ship_coors[x][1];

if(count == 5) — True → Output: "You hit their Carrier!

False

else if(count == 4) — True → Output: "You hit their Battleship!

False

else if(count == 3) — True → Output: "You hit their Submarine or Cruiser!

False

else if(count == 2) — True → Output: "You hit their Destroyer!

False

if(count == 5) — True → Output: "You hit their Carrier!

False

else if(count == 4) — True → Output: "You hit their Battleship!

False

else if(count == 3) — True → Output: "You hit their Submarine or Cruiser!

False

else if(count == 2) — True → Output: "You hit their Destroyer!

False

```
bool shooting2()
```

//Declare Variables
int num[2];
int hits = 0;
int count = 0;

Output:
plyrs->info[1].player_name

Output:
", where to shoot?"

Input:
getline(cin,plyrs->info[1].shts_frd[count2]);

if(plyrs->info[1].shts_frd[count2][0] == 'Q' || plyrs->info[1].shts_frd[count2][0] == 'q')

True → Output: "Game over!" → ```return true;```

False

while(!isalpha(plyrs->info[1].shts_frd[count2][0])
&& !isupper(plyrs->info[1].shts_frd[count2][0])
|| !isdigit(plyrs->info[1].shts_frd[count2][1]))

False

True

Input:
getline(cin,plyrs->info[1].shts_frd[count2]);

Output:
", where to shoot?"

Output:
plyrs->info[1].player_name

Output:
"Please input a correct coordinate. (ex. A10, B7, I3)"

Declare and initialize
In for Loop
j = 0

j++

j < plyrs->SIZE

True

if(isalpha(plyrs->info[1].shts_frd[count2][j]))

False

True

num[j] = plyrs->info[1].shts_frd[count2][j] - 64;

False → d.

if(isdigit(plyrs->info[1].shts_frd[count2][j]) && plyrs->info[1].shts_frd[count2][j+1] == '0')

True →
num[j] = 10;
break;

False

else if(isdigit(plyrs->info[1].shts_frd[count2][j]))

True →
num[j] = plyrs->info[1].shts_frd[count2][j] - 48;

False

```
d.

m++

Declare and initialize
In for Loop
m = 0

Declare and initialize
In for Loop
l = 0

l++

m < 11

l < 11    True    if(l == num[1] && m == num[0])    True    if(plyrs->info[0].player_board[m-1][l-1]== 'X')

False                      False                                    False                    True

plyrs->info[1].enemy_board[m-1][l-1] = 'M';      Output:
plyrs->info[0].player_board[m-1][l-1] = 'M';     "It was a miss!

plyrs->info[1].enemy_board[m-1][l-1] = 'H';      Output:
e.    plyrs->info[0].player_board[m-1][l-1] = 'H';   "It was a hit!

Declare and initialize
In for Loop
z = 0

z++

z < 11    True    Declare and initialize
In for Loop
y = 0

y++

False

y < 11
False

True

if(plyrs->info[0].player_board[z][y] == 'X')    True    hits++
False

if (hits == 0)    False

True

Output:
plyrs->info[1].player_name

Output:
" has won the game!"

return true;

count1++:

return false;
```

```
     ┌──────────────────────┐
(e.)─→│ Declare and initialize│
      │ In for Loop          │         ┌─────┐
      │ x = 0                │         │ x++ │←──────────────────────────────┐
      └──────────┬───────────┘         └──┬──┘                               │
                 │                         ↑                                  │
                 ↓                                                            │
          ◇ x < plyrs->numshps ◇                                             │
```

if(m >= plyrs->info[0].ship_coors[x][0] && m <= plyrs->info[0].ship_coors[x][3] && l == plyrs->info[0].ship_coors[x][1] && l == plyrs->info[0].ship_coors[x][4])

False

else if(l >= plyrs->info[0].ship_coors[x][1] && l <= plyrs->info[0].ship_coors[x][4] && m == plyrs->info[0].ship_coors[x][0] && m == plyrs->info[0].ship_coors[x][3])

False

True

count = (plyrs->info[0].ship_coors[x][3]+1) - plyrs->info[0].ship_coors[x][0];

True

count = (plyrs->info[0].ship_coors[x][4]+1) - plyrs->info[0].ship_coors[x][1];

if(count == 5) — True → Output: "You hit their Carrier!

False

else if(count == 4) — True → Output: "You hit their Battleship!

False

else if(count == 3) — True → Output: "You hit their Submarine or Cruiser!

False

else if(count == 2) — True → Output: "You hit their Destroyer!

False

if(count == 5) — True → Output: "You hit their Carrier!

False

else if(count == 4) — True → Output: "You hit their Battleship!

False

else if(count == 3) — True → Output: "You hit their Submarine or Cruiser!

False

else if(count == 2) — True → Output: "You hit their Destroyer!

False

```
void disp_board1()
```

// The char array for letters in the game board
char letters[10] =
{'A','B','C','D','E','F','G','H','I','J'};

Output:

"X = Your ships
H = The enemy ships has been hit
M = Your shot has not hit any ships

-------------- Enemy Board ------------
  1 2 3 4 5 6 7 8 9 10
  _ _ _ _ _ _ _ _ _ _ "

Declare and initialize
In for Loop
x = A

x < K

True

Output:
cout << "   " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

z < K

False

True

Output:
cout << "| " << plyrs->info[0].enemy_board[x][z]<< " ";

Output:
cout << "|" << endl
<< "   - - - - - - - - - - -"<< endl;

x++

False

prntTofile(plyrs,0,plyr_1file);

Output:

"X = Your ships
H = Your ship has been hit
M = The enemy has missed your ships

-------------- Player Board ------------
  1 2 3 4 5 6 7 8 9 10
  _ _ _ _ _ _ _ _ _ _ "

Declare and initialize
In for Loop
x = A

x < K

True

Output:
cout << "   " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

z < K

False

True

Output:
cout << "| " << plyrs->info[0].player_board[x][z]<< " ";

Output:
cout << "|" << endl
<< "   - - - - - - - - - - -"<< endl;

x++

False

return;

void disp_board2()

```
// The char array for letters in the game board
char letters[10] = {'A','B','C','D','E','F','G','H','I','J'};
```

Output:

"X = Your ships
H = The enemy ship has been hit
M = Your shot has not hit any ships

--------------- Enemy Board ------------
1 2 3 4 5 6 7 8 9 10
_ _ _ _ _ _ _ _ _ _"

Declare and initialize
In for Loop
x = A

x < K

True →

Output:
cout << " " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

z < K

False

True

Output:
cout << "| " << plyrs->info[1].enemy_board[x][z]<< " ";

x++

Output:
cout << "|" << endl
<< " - - - - - - - - - - -"<< endl;

False

prntTofile(plyrs,1,plyr_2file);

Output:

X = Your ships
H = Your ship has been hit
M = The enemy has missed your ships

--------------- Player Board ------------
1 2 3 4 5 6 7 8 9 10
_ _ _ _ _ _ _ _ _ _"

Declare and initialize
In for Loop
x = A

x < K

True

Output:
cout << " " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

z < K

False

True

Output:
cout << "| " << plyrs->info[1].player_board[x][z]<< " ";

x++

Output:
cout << "|" << endl
<< " - - - - - - - - - - -"<< endl;

False

return;

```
void prntTofile()
```

Output
plyr_file << endl
<< "   ----- All Plays Made during the Game by this player -----"
<< endl << endl;

// The char array for letters in the game
board
char letters[10] =
{'A','B','C','D','E','F','G','H','I','J'};

Output:
plyr_file << "   1 2 3 4 5 6 7 8 9 10" << endl
<< "   _ _ _ _ _ _ _ _ _ _ " << endl << endl;

Declare and initialize
In for Loop
x = A

x++

Output:
plyr_file << "|" << endl
<< "   - - - - - - - - - - - -"<< endl;

x < K — True → Output:
plyr_file << "   " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

False

z < K

True

Output:
plyr_file << "| " << plyrs->info[1].enemy_board[x][z]<< " ";

False

Output:
plyr_file << "   1 2 3 4 5 6 7 8 9 10" << endl
<< "   _ _ _ _ _ _ _ _ _ _ " << endl << endl;

Declare and initialize
In for Loop
x = A

x++

Output:
prnt_file << "|" << endl
<< "   - - - - - - - - - - - -"<< endl;

x < K — True → Output:
plyr_file << "   " <<
letters[x] << " ";

Declare and initialize
In for Loop
z= A

z++

False

return;

False

z < K

True

Output:
plyr_file << "| " << plyrs->info[1].player_board[x][z]<< " ";
```

## void prntToBinFile()

if(!file) — **True** → Output: "error"

**False**

Declare and initialize
In for Loop
z = 0

z < plyrs->SIZE — **True** → file.write(reinterpret_cast(&plyrs->info[z]), sizeof(plyrs->info[z]));

z++

**False**

//Close the Binary File
file.close();

return;

Declare and initialize
In for Loop
count = 0

count++

count < plyrs->SIZE — **True** → Output:
"    ---- Player #"

Output:
(count + 1)

Output:
" Information ---"

Output:
"Name: "

Output:
temp[count].player_name

**False**

return;

## void BinFileToprnt()

//Reopen the Binary file
file.open("Players Info", ios:: in | ios:: binary);

//Temporary struct array used for output
struct Playerinfo temp[2];

if(!file) — **True** → Output:
"error"

**False**

//Read the records from the Binary file
file.read(reinterpret_cast(&temp), sizeof(temp));

Declare and initialize
In for Loop
z = 0

z++

z < plyrs->numshps — **True** → Output:
cout << "     " << temp[count].ships[z] << endl;

**False**

Output:
"Shots Fired: "

Declare and initialize
In for Loop
z = 0

z++

z < 100 — **True** → Output:
cout << " " << temp[count].shts_frd[z];

**False**

Output:
"Final Player Board: " → f.

Output:
" Coordinates of ships"

```
           ( f. )
             │
             ▼
┌─────────────────────┐
│ Declare and initialize │
│    In for Loop       │ ◄──────────────┐
│       z = 0          │                │
└─────────────────────┘         ┌──────────┐
             │                  │   z++    │ ◄──────────────────┐
             │                  └──────────┘                    │
             ▼                                                  │
   False    ╱╲          True    ┌─────────────────────┐         │
 ◄─────────╱  ╲────────────────►│ Declare and initialize │       │
          ╲ z < 10 ╱           │    In for Loop       │         │
           ╲╱                  │       f = 0          │ ◄────┐   │
                               └─────────────────────┘  ┌──────┐ │
                                          │             │ f++  │◄┤
                                          ▼             └──────┘ │
                               ╱╲   True  ┌──────────────────────────┐
                              ╱  ╲───────►│        Output:           │
                             ╲ f < 10╱    │     cout << " " <<       │
                              ╲╱          │ temp[count].player_board[z][f]; │
                               │ False    └──────────────────────────┘
                               └──────────────────────────────────────┘
```