

Worksheet 5 - Jake Richards

by Jake Richards

Submission date: 09-Jan-2026 12:46PM (UTC+0000)

Submission ID: 272669360

File name: 64894_Jake_Richards_Worksheet_5_-_Jake_Richards_703116_1680970186.docx (297.38K)

Word count: 781

Character count: 4359

Name: _____

Student Number: _____

WORKSHEET 5 – TESTING (Battleships)

For this worksheet, the program can be implemented in a number of different ways and demonstrate different features.

Create your own test set to demonstrate strengths and/or limitations of your solution. Use as many of the template boxes below as you need.

Test: Enter invalid coordinates

Screenshot(s) of output:

```
Welcome to Battleship!

 0 1 2 3 4 5 6 7 8 9
0 ~ ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)10
Please enter a column (between 0 and 9)10
Please enter a valid coordinate.

 0 1 2 3 4 5 6 7 8 9
0 ~ ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Told the user to re-enter the coordinates.

Test: Enter coordinates already entered

Screenshot(s) of output:

```
Please enter a row (between 0 and 9)0
Please enter a column (between 0 and 9)0
You missed!

  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)0
Please enter a column (between 0 and 9)0
Position already fired upon!

  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Told the user that the position had already been fired upon.

Test: Miss a ship

Screenshot(s) of output:

```
Please enter a row (between 0 and 9)0
Please enter a column (between 0 and 9)0
You missed!

  0 1 2 3 4 5 6 7 8 9
0  0 ~ ~ ~ ~ ~ ~ ~ ~
1  ~ ~ ~ ~ ~ ~ ~ ~
2  ~ ~ ~ ~ ~ ~ ~ ~
3  ~ ~ ~ ~ ~ ~ ~ ~
4  ~ ~ ~ ~ ~ ~ ~ ~
5  ~ ~ ~ ~ ~ ~ ~ ~
6  ~ ~ ~ ~ ~ ~ ~ ~
7  ~ ~ ~ ~ ~ ~ ~ ~
8  ~ ~ ~ ~ ~ ~ ~ ~
9  ~ ~ ~ ~ ~ ~ ~ ~
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Told the user that they missed. Print shows a O to represent a miss.

Test: Strike a ship

Screenshot(s) of output:

```
Please enter a row (between 0 and 9)4
Please enter a column (between 0 and 9)4
You missed!

  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ 0 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ 0 ~ ~ ~ ~
5 ~ ~ ~ ~ ~ ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)5
Please enter a column (between 0 and 9)5
Strike! You have damaged the Battleship.

  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ 0 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ ~ ~ ~ ~
4 ~ ~ ~ ~ 0 ~ ~ ~ ~
5 ~ ~ ~ ~ ~ X ~ ~ ~
6 ~ ~ ~ ~ ~ ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~

Please enter a row (between 0 and 9)
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Told the user that a battleship had been struck. Print X demonstrates a strike.

Test: Sink a ship

Screenshot(s) of output:

```
Please enter a column (between 0 and 9)5
Strike! You have damaged the Battleship.
```

```
  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ 0 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ 0 ~ ~ ~ ~
4 ~ ~ ~ ~ 0 X ~ ~ ~
5 ~ ~ ~ ~ 0 X ~ ~ ~
6 ~ ~ ~ ~ X ~ ~ ~
7 ~ ~ ~ ~ ~ ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~
```

```
Please enter a row (between 0 and 9)7
```

```
Please enter a column (between 0 and 9)5
Strike! You have damaged the Battleship.
Kaboom! You have sunk the Battleship.
```

```
  0 1 2 3 4 5 6 7 8 9
0 0 ~ ~ ~ ~ ~ ~ ~ ~
1 ~ ~ ~ ~ ~ ~ ~ ~
2 ~ ~ ~ 0 ~ ~ ~ ~ ~
3 ~ ~ ~ ~ 0 ~ ~ ~ ~
4 ~ ~ ~ ~ 0 X ~ ~ ~
5 ~ ~ ~ ~ 0 X ~ ~ ~
6 ~ ~ ~ ~ X ~ ~ ~
7 ~ ~ ~ ~ X ~ ~ ~
8 ~ ~ ~ ~ ~ ~ ~ ~
9 ~ ~ ~ ~ ~ ~ ~ ~
```

```
Please enter a row (between 0 and 9)
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

Told the user that a battleship had been sunk. No change to print as expected.

Test: Win the game.

Screenshot(s) of output:

```
Please enter a row (between 0 and 9)7 3
Please enter a column (between 0 and 9)Strike! You have damaged the Patrol Boat.

  0 1 2 3 4 5 6 7 8 9
0 0 0 0 0 0 0 0 0 0
1 0 X 0 0 0 0 0 X X 0
2 ~ X ~ 0 ~ ~ ~ X X ~
3 ~ X ~ ~ 0 ~ ~ X X ~
4 ~ ~ ~ 0 X 0 X ~ ~
5 ~ ~ ~ 0 X ~ X ~ ~
6 ~ ~ ~ ~ X ~ ~ ~ ~
7 0 0 0 X ~ X ~ ~ ~
8 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0

Please enter a row (between 0 and 9)7 4
Please enter a column (between 0 and 9)Strike! You have damaged the Patrol Boat.
Kaboom! You have sunk the Patrol Boat.

  0 1 2 3 4 5 6 7 8 9
0 0 0 0 0 0 0 0 0 0
1 0 X 0 0 0 0 0 X X 0
2 ~ X ~ 0 ~ ~ ~ X X ~
3 ~ X ~ ~ 0 ~ ~ X X ~
4 ~ ~ ~ 0 X 0 X ~ ~
5 ~ ~ ~ 0 X ~ X ~ ~
6 ~ ~ ~ ~ X ~ ~ ~ ~
7 0 0 0 X X ~ ~ ~ ~
8 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0
You won!
C:\Users\Jake\Desktop\University Year 1\Software Development\worksheetFive\64\
```

Result: Did the programme behave as expected ie syntactically and logically correct?

YES

After striking the final boat, the game immediately ended by "You won!"

Final code:

[Copy and paste your code here – note, do not use a screenshot.

Note that there is no “before” and “after” code for this Worksheet because it is “Open” – not done in class. Therefore just post your final code below]

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <Windows.h>
#include <ctype.h>
#include <time.h>

// Functions
void init_battle_grid(void);
void print_battle_grid(void);
void pos_battle_grid(void);
void play_battle_grid(void);
int check_ship_placement(int r, int c, int s, int v);
int check_ship_health();

// Global Params
int battleGridSize = 10;
char battleGrid[10][10] = { 0 };
char shipNames[5][12] = { "Carrier", "Battleship", "Destroyer", "Submarine", "Patrol Boat" };
int shipSizes[5] = { 5,4,3,3,2 };
int shipHealth[5] = { 5,4,3,3,2 };

int main()
{
    system("color 1f");
    srand(time(NULL));
    printf("Welcome to Battleship!");
    printf("\n");
    init_battle_grid();
    pos_battle_grid();
    play_battle_grid();
}

void init_battle_grid()
{
    for (int i = 0; i < battleGridSize; i++)
    {
        for (int j = 0; j < battleGridSize; j++)
        {
            if (battleGrid[i][j] == '\0')
            {
                battleGrid[i][j] = '~';
            }
        }
    }
}
```

```

    }
    }
    //printf("Initialised Battle Grid");
}

void print_battle_grid()
{
    printf("\n\n ");
    //printf("Initialised Battle Grid\n");
    for (int i = 0; i < battleGridSize; i++)
    {
        printf("%d ", i);
    }
    printf("\n");
    for (int r = 0; r < battleGridSize; r++)
    {
        printf("%d ", r);
        for (int c = 0; c < battleGridSize; c++)
        {
            char shipPos = battleGrid[r][c];
            if (shipPos >= '0' && shipPos <= '4')
            {
                shipPos = '~';
            }
            printf("%c ", shipPos);
        }
        printf("\n");
    }
}

int check_ship_placement(int r, int c, int s, int v)
{
    if (v) // Place Vertically
    {
        if (r + s > battleGridSize)
            return 0;

        for (int i = 0; i < s; i++)
            if (battleGrid[r + i][c] != '~')
                return 0;
    }
    else // Place Horizontally
    {
        if (c + s > battleGridSize)
            return 0;

        for (int i = 0; i < s; i++)
            if (battleGrid[r][c + i] != '~')
                return 0;
    }
}

```

```

        return 1;
    }

void pos_battle_grid()
{
    for (int shipNo = 0; shipNo < 5; shipNo++)
    {
        int vertical, rowCoord, colCoord;

        do
        {
            vertical = rand() % 2;
            rowCoord = rand() % battleGridSize;
            colCoord = rand() % battleGridSize;
        } while (!check_ship_placement(rowCoord, colCoord, shipSizes[shipNo], vertical));

        // Place ship AFTER valid position is found
        for (int i = 0; i < shipSizes[shipNo]; i++)
        {
            if (vertical)
                battleGrid[rowCoord + i][colCoord] = 'O' + shipNo;
            else
                battleGrid[rowCoord][colCoord + i] = 'O' + shipNo;
        }
    }
}

int check_ship_health()
{
    for (int i = 0; i < 5; i++)
    {
        if (shipHealth[i] > 0)
        {
            return 1;
        }
    }
    return 0;
}

void play_battle_grid()
{
    int playerRow, playerCol;
    while (check_ship_health())
    {
        print_battle_grid();
        printf("\n\nPlease enter a row (between 0 and 9)");
        scanf("%d", &playerRow);
        printf("\nPlease enter a column (between 0 and 9)");
        scanf("%d", &playerCol);
    }
}

```

```

        if (playerRow < 0 || playerRow >= battleGridSize || playerCol < 0 || playerCol >=
battleGridSize)
        {
            printf("Please enter a valid coordinate.");
            continue;
        }
        char shipPos = battleGrid[playerRow][playerCol];
        if (shipPos == 'X' || shipPos == 'O')
        {
            printf("Position already fired upon!");
            continue;
        }
        if (shipPos >= '0' && shipPos <= '4')
        {
            int shipNo = shipPos - '0';
            shipHealth[shipNo]--;
            battleGrid[playerRow][playerCol] = 'X';
            printf("Strike! You have damaged the %s.", shipNames[shipNo]);
            if (shipHealth[shipNo] == 0)
            {
                printf("\nKaboom! You have sunk the %s.", shipNames[shipNo]);
            }
        }
        else
        {
            battleGrid[playerRow][playerCol] = 'O';
            printf("You missed!");
        }
    }
    print_battle_grid();
    printf("You won!");
}

```

Worksheet 5 - Jake Richards

GRADEMARK REPORT

FINAL GRADE

27/0

GENERAL COMMENTS

WORKSHEET 5

Coding- Quite a complex solution, although be careful with global parameters, they should be avoided (I am OK with the arrays being global at Level 4)

Complex solution, particularly around the randomisation of the ships. Good validation, just a few minor inefficiencies. Well done.

14/15

Testing - Could have been just a little more extensive, eg showing the functioning of the randomisation, but overall most eventualities show. Discussion is a bit on the brief side, but I guess there wasn't much more to say! 13.5/15

Overall mark Worksheet 5: 27.5/30

Organisation of report, writing style, presentation of work

Generally good, although I don't think you've filled in your name and student number at the top for (m)any worksheets. Writing style is good too, make sure you've communicated everything you need to, sometimes you're overly brief. Good improvement in screenshots display from Worksheets 1 and 2 - much better and clearer in Worksheets 3,4 and 5 so good to see you've improved due to the feedback there.

7.5/10

Overall mark for assignment (5 worksheets) -
78.5%

Test mark - 83%

Overall mark for module: 80%

PAGE 1
PAGE 2
PAGE 3
PAGE 4
PAGE 5
PAGE 6
PAGE 7
PAGE 8
PAGE 9
PAGE 10