```
Script started on 2023-11-07 03:16:47+00:00 [TERM="xterm-256color" TTY="/dev/pts/0" COLUM
NS="67" LINES="97"]
\[\033[01;34m\]\w\[\033[00m\]$ pwd
/home/runner/Lab-16-Minesweeper-Refactoring-and-Dynamic-Allocati-kcp3s
\[\033[01;34m\]\w\[\033[00m\]$ ls -la
total 2516
drwxr-xr-x 1 runner runner     268 Nov  7 03:16 .
drwxrwxrwx 1 runner runner     166 Nov  7 02:09 ..
-rwxr-xr-x 1 runner runner   17992 Nov  7 03:15 a.out
-rw-r--r-- 1 runner runner    2353 Nov  7 03:12 Board.cpp
-rw-r--r-- 1 runner runner     617 Nov  7 01:43 Board.h
-rw-r--r-- 1 runner runner      17 Oct 27 20:51 .breakpoints
drwxr-xr-x 1 runner runner      12 Jan 24  2022 .cache
drwxr-x--- 1 runner runner     638 Nov  4 15:59 .ccls-cache
drwxr-xr-x 1 runner runner      68 Nov  6 21:00 .lesson
-rwxr-xr-x 1 runner runner 1254392 Oct 27 20:53 main
-rw-r--r-- 1 runner runner    1013 Nov  6 02:07 main.cpp
-rwxr-xr-x 1 runner runner 1255712 Oct 27 20:53 main-debug
-rw-r--r-- 1 runner runner     449 Oct 27 20:53 Makefile
-rw-r--r-- 1 runner runner       0 Nov  7 03:16 Patel_Lab_16.log
-rw-r--r-- 1 runner runner    1426 Dec 21  2022 .replit
-rw-r--r-- 1 runner runner     121 Oct 31 17:54 replit.nix
-rw-r--r-- 1 runner runner     475 Nov  7 02:58 Tile.cpp
-rw-r--r-- 1 runner runner     279 Nov  7 01:43 Tile.h
\[\033[01;34m\]\w\[\033[00m\]$ cat -n main.cpp
     1  #include "Board.h"
     2  #include "Tile.h"
     3  #include <iostream>
     4
     5  int main() {
     6    Tile test_tile;                // should default to true
     7    test_tile.display();           // print a numeric value i.e. 0
     8    test_tile.set_revealed(false); // sets it to false
     9    test_tile.display();            // should see a # instead
    10
    11    std::cout << "\n---First Board Test---\n";
    12    Board play_area;   // default constructor creates an 8x8 board with 10 mines
    13    play_area.print(); // displays an 8x8 board with mines and counts
    14
    15    std::cout << "\n---Second Board Test---\n";
    16    int rows;
    17    int columns;
    18    int mine_count;
    19    std::cout << "Enter a board size rows followed by columns: ";
    20    std::cin >> rows >> columns;
    21    std::cout << "Enter number of mines for custom difficulty: ";
    22    std::cin >> mine_count;
    23    Board play_area2(rows, columns,
    24                     mine_count); // uses the overloaded constructor
    25    play_area2.print(); //  should display a board of rows x columns with
    26                        //  mine_count mines and the counts
    27  }\[\033[01;34m\]\w\[\033[00m\]$ cat -n Board.h
     1  // header file for the Board class
     2  #ifndef BOARD_H
     3  #define BOARD_H
     4  #include "Tile.h"
     5
     6  // int const ROW{8};
     7  // int const COLUMN{8};
     8
     9  class Board {
    10  private:
    11    // std::array<std::array<int, COLUMN>, ROW> board;
    12    // int mine_count = 10;
    13
    14    int m_board_width;
    15    int m_board_height;
    16    int m_size;
    17    Tile *tiles {};
    18
    19    void place_mines(int mine_count);
    20    void update_counts();
    21
    22  public:
    23    Board(); // Default to 8x8 and 10 mines and run from second constructor
    24    Board(int rows, int columns, int mine_count); // Custom usernum board
    25    ˜Board(); // When destroying delete the dynamic location
    26    void print() const;
    27  };
    28
    29  #endif\[\033[01;34m\]\w\[\033[00m\]$ cat -n Board.cpp
     1  // implementation file for the Board class
     2  #include "Board.h"
     3  #include <cstdlib>
     4  #include <ctime>
     5  #include <iostream>
     6
     7  // Default constructor
     8  Board::Board() {
     9    m_board_width = 8;
    10    m_board_height = 8;
    11    m_size = (m_board_width * m_board_height);
    12    tiles = new Tile[m_size];
    13    place_mines(10);
    14    update_counts();
    15  }
    16
    17  // Destructor to destroy/delete the location at the end
    18  Board::˜Board() { delete[] tiles; }
    19
    20  // User given dimentions and mine construction
    21  Board::Board(int rows, int columns, int mine_count) {
    22    m_board_width = rows;
    23    m_board_height = columns;
    24    m_size = (m_board_width * m_board_height);
    25    tiles = new Tile[m_size];
    26    place_mines(mine_count);
    27    update_counts();
    28  }
    29
    30  // Placing the mines both default and user defined
    31  void Board::place_mines(int mine_count) {
    32    srand(time(0));
    33    int random;
    34    int initial_mines = 0;
    35
    36    while (initial_mines < mine_count) {
    37      random = rand() % m_size;
    38      if (tiles[random].get_value() != 9) {
    39        tiles[random].set_value(9);
    40        initial_mines++;
    41      }
    42    }
    43  }
    44
    45  // Updating the counter
    46  void Board::update_counts() {
    47    for (int i = 0; i < m_size; i++) {
    48      if (tiles[i].get_value() != 9) {
    49        int counter = 0;
    50        int i_row = i / m_board_width;
    51        int i_col = i % m_board_width;
    52        for (int r = -1; r <= 1; r++) {
    53          for (int c = -1; c <= 1; c++) {
    54            int rows = i_row + r;
    55            int cols = i_col + c;
    56
    57            if (rows >= 0 && rows < m_board_height && cols >= 0 &&
    58                cols < m_board_width) {
    59              int index = m_board_width * rows + cols;
    60              if (tiles[index].get_value() == 9) {
    61                counter++;
    62              }
    63            }
    64          }
    65        }
    66        tiles[i].set_value(counter);
    67      }
    68    }
    69  }
    70  // Printing the board
    71  void Board::print() const {
    72    std::cout << "|---";
    73    for (int i = 1; i < m_board_width; i++) {
    74      std::cout << "|---";
    75    }
    76    std::cout << "|" << std::endl;
    77    for (int i = 0; i < m_board_height; i++) {
    78      std::cout << "| ";
    79      for (int k = 0; k < m_board_width; k++) {
    80        if (tiles[i * m_board_width + k].get_value() == 9) {
    81          std::cout << "M | ";
    82        } else {
    83          std::cout << tiles[i * m_board_width + k].get_value() << " | ";
    84        }
    85      }
    86      std::cout << "\n"
    87                << "|---";
    88      for (int i = 1; i < m_board_width; i++) {
    89        std::cout << "|---";
    90      }
    91      std::cout << "|" << std::endl;
    92    }
    93  }\[\033[01;34m\]\w\[\033[00m\]$ cat -n Tile.hpp
     1  // header file for the Tile class
     2  #ifndef TILE_H
     3  #define TILE_H
     4  #include <iostream>
     5
     6  class Tile {
     7  private:
     8    int m_value;
     9    bool m_revealed = true;
    10
    11  public:
    12  Tile();
    13  void display() const;
    14  void set_revealed(bool reveal);
    15  int get_value() const;
    16  void set_value(int value);
    17  };
    18
    19  #endif\[\033[01;34m\]\w\[\033[00m\]$ cat -n Tile.cpp
     1  // Implementation file for the set_revealed and display methods of Tile
     2  #include "Tile.h"
     3  #include <iostream>
     4  Tile::Tile() {}
     5
     6  void Tile::set_value(int value) { m_value = value; }
     7
     8  int Tile::get_value() const { return m_value; }
     9
    10  void Tile::set_revealed(bool reveal) { m_revealed = reveal; }
    11
    12  void Tile::display() const {
    13    if (m_revealed) {
    14      if (m_value == 9) {
    15        std::cout << "M";
    16      } else {
    17        std::cout << m_value;
```

```
      }
    } else {
      std::cout << "#";
    }
  }\[\033[01;34m\]\w\[\033[00m\]$ g++ main.cpp Board.cpp Tile.cpp -o minesweeper
\[\033[01;34m\]\w\[\033[00m\]$ ./minesweeper
997967056#
---First Board Test---
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | M | 4 | M | 3 | M |
|---|---|---|---|---|---|---|---|
| M | 1 | 3 | M | 6 | M | 3 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | M | M | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 2 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | M | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | M | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

---Second Board Test---
Enter a board size rows followed by columns: 4 4
Enter number of mines for custom difficulty: 2
|---|---|---|---|
| 1 | 1 | 0 | 0 |
|---|---|---|---|
| M | 2 | 0 | 0 |
|---|---|---|---|
| M | 2 | 0 | 0 |
|---|---|---|---|
| 1 | 1 | 0 | 0 |
|---|---|---|---|
\[\033[01;34m\]\w\[\033[00m\]$ ./minesweeper
808184576#
---First Board Test---
|---|---|---|---|---|---|---|---|
| 1 | M | 1 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 2 | 1 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 2 | M | 4 | 2 | 1 | 1 | M |
|---|---|---|---|---|---|---|---|
| 0 | 2 | M | M | M | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | M | 3 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| M | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 2 | M | 2 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | M | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

---Second Board Test---
Enter a board size rows followed by columns: 5 5
Enter number of mines for custom difficulty: 3
|---|---|---|---|---|
| 0 | 1 | M | M | 1 |
|---|---|---|---|---|
| 0 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 1 |
|---|---|---|---|---|
| 0 | 0 | 1 | M | 1 |
|---|---|---|---|---|
\[\033[01;34m\]\w\[\033[00m\]$ ./minesweeper
-1801107776#
---First Board Test---
|---|---|---|---|---|---|---|---|
| 3 | M | 2 | 0 | 1 | 2 | M | 1 |
|---|---|---|---|---|---|---|---|
| M | M | 2 | 0 | 1 | M | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 2 | 2 | 1 | 0 | 1 | 1 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | M | 1 | 1 | M | 3 | M |
|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 2 | M | 2 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 2 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | M | 1 | 0 |
|---|---|---|---|---|---|---|---|

---Second Board Test---
Enter a board size rows followed by columns: 6 6
Enter number of mines for custom difficulty: 10
|---|---|---|---|---|---|
| 3 | M | M | M | 1 | 0 |
|---|---|---|---|---|---|
| M | M | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|
| 2 | 2 | 1 | 1 | M | 1 |
|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 2 | 1 |
|---|---|---|---|---|---|
| 0 | 1 | M | M | 3 | 1 |
|---|---|---|---|---|---|
| 0 | 1 | 3 | M | 3 | M |
|---|---|---|---|---|---|
\[\033[01;34m\]\w\[\033[00m\]$ ./minesweeper
1625932832#
---First Board Test---
|---|---|---|---|---|---|---|---|
| 1 | 2 | 1 | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| M | 3 | M | 1 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 2 | M | 3 | 2 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | M | 2 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 0 | 1 | M | 2 | 2 | M | 2 | M |
|---|---|---|---|---|---|---|---|
| 1 | 3 | 3 | 2 | 1 | 1 | 2 | 1 |
|---|---|---|---|---|---|---|---|
| 1 | M | M | 2 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | M | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

---Second Board Test---
Enter a board size rows followed by columns: 8 8
Enter number of mines for custom difficulty: 10
|---|---|---|---|---|---|---|---|
| 2 | M | 1 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| M | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | M | 2 | 1 | 2 | 2 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | M | 3 | 1 | M | M |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 3 | M | 4 | 2 | 2 | 2 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | M | M | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 3 | 3 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | M | 1 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|
\[\033[01;34m\]\w\[\033[00m\]$ exit

Script done on 2023-11-07 03:19:49+00:00 [COMMAND_EXIT_CODE="0"]
```