

Forum:[Hack Forums](#)

Thread: [Coding a game-board to play in C++ \[Requesting Assistance\]](#)

Zink - April 2, 2018, 1:19 pm

I am coding a game-board challenge, but I am getting an error when I set the required number for the code to initialize the game-board.

[img]http://#/laMtBd.png[/img]

Code Here is the current code I have.

Annabelle Lee - April 2, 2018, 2:23 pm

The obvious error I see is: `var+1`; This is not going to do anything, you need to either increment with `var++`; or do `var = var + 1`; Both should work. Apart from this you don't seem to have correct formatting when printing out the board. I've taken the liberty to fix up your print function to display your intended board a bit better. You can go from here to add the column headers yourself. Overall your code has some other issues, which I would be happy to go over if you'd like me to.

Here is an updated print function:

```
Code void prntBoard(Pieces board[][COLUMNS],int& num_rows)
{
    char var = 'A';
    for(int i=0;i<num_rows;i++)
    {
        cout << var << " ";
        for(int j=1;j<=COLUMNS;j++)
        {
            cout << "-";
            // print out the var command.
            cout << " ";
        }
        var = var + 1;
        cout << endl;
    }
}
```

Zink - April 2, 2018, 4:42 pm

(Apr 2, 2018 02:23 PM)Annabelle Lee Wrote: [\[CITE\]](#) The obvious error I see is: `var+1`; This is not going to do anything, you need to either increment with `var++`; or do `var = var + 1`; Both should work. Apart from this you don't seem to have correct formatting when printing out the board. I've taken the liberty to fix up your print function to display your intended board a bit better. You can go from here to add the column headers yourself. Overall your code has some other issues, which I would be happy to go over if you'd like me to.

Here is an updated print function:

```

Code void prntBoard(Pieces board[][COLUMNS],int& num_rows)
{
char var = 'A';
for(int i=0;i<num_rows;i++)
{
    cout << var << " ";
    for(int j=1;j<=COLUMNS;j++)
    {
        cout << "-";
        // print out the var command.
        cout << " ";
    }
    var = var + 1;
    cout << endl;
}
}

```

Ohhh okay I see what I did wrong.

Yeah If you can that would be great, I still have a bit to do.

Annabelle Lee - April 2, 2018, 5:23 pm

Okay. Going to step through this code line by line and go over some of the issues I feel like are glaring. Please keep in mind that some of it might be nitpicking, so please bear with me on that.

First issue overall I see is just the way the formatting, spacing, and consistency of the usage of caps and variable names should be a bit better. Now for the rest of it:

```

Code #include <iostream>
#include <cmath>    // Libraries used for the project files.

```

```

using namespace std;
This is fine, nothing to see here.

```

```

Code enum Pieces{SPACE = ' ', RED = 2, BLUE = 3}; // Enumeration for board pieces.
const int COLUMNS = 6;
void header();
void instructions();
void prntBoard(Pieces board[][COLUMNS],int& num_rows);

```

Few issues here is the lack of consistency for upper/lowercase. (Nitpicking.) As well as the way you named your variables. You don't need a const int for COLUMNS, you can simply use a macro (#define) to make it a bit neater. Likewise if you have a COLUMNS, you should name your rows: ROWS, not num_rows. (It reads better. Obviously it is the number of rows.) So a slightly neater version is thus. Also, don't make spelling mistakes in function names for the sake of brevity. (prnt instead of print.) Lastly, you don't need an enum here, and in my opinion it simply makes the program more confusing. Instead simply use a 2D array of ints for the board pieces, and when iterating over board, replace with characters in the print function. For example a 0 on board is empty. A 1 is Red and a 2 is Blue.

```

Code #define COLUMNS 6
void header();
void instructions();
void printBoard(int board[][COLUMNS],int& num_rows);
Code int main()
{

```

```

header();
instructions();
int ROWS;
do
{
    cout << " Enter the number of rows on the board between 4 & 6, inclusively : " << endl;
    cin >> ROWS;
    if(ROWS >= 4 && ROWS <= 6)
    {
        cout <<" Initializing game... " << endl;
    }
    else
    {
        cout << "Invalid value outside range, please try again. : " << endl;
    }
} while(ROWS < 4 || ROWS > 6);

int board[ROWS][COLUMNS];
// Set all the board spots to 0; Else we will have undefined behaviour.
for(int i = 0; i < ROWS; i++)
{
    for(int j = 0; j < COLUMNS; j++)
    {
        board[i][j] = 0;
    }
}
printBoard(board, ROWS);
return 0;
}

```

Fixed up main a bit to look nicer and follow guidelines. Also note how the board is now a 2D array of integers, and is using 2 for loops to make sure all elements are zeroed.

```

Code void header()
{
    cout << "#++++++#\n";
    cout << "!   Computer Science and Engineering           !\n";
    cout << "!   CSCE 1030 Sec 002 - Computer Science I       !\n";
    cout << "#++++++#\n\n";
}

```

```

void instructions()
{
    cout << " W e l c o m e   t o   1 0 3 0   C o n n e c t ! " << endl; // Project instructions that is
created.
    cout << " -----" << endl;
}

```

Header and instruction functions are fine, just fix up a bit of the spacing/tabbing.

```

Code void printBoard(int board[][COLUMNS], int ROWS)
{
    char var = 'A';
    cout << " 1 2 3 4 5 6" << endl;
    for(int i = 0; i < ROWS; i++)

```

```

{
    cout << var << " ";
    for(int j = 0; j < COLUMNS; j++)
    {
        if(board[i][j] == 0)
        {
            cout << "-";
        } else if(board[i][j] == 1)
        {
            cout << "R";
        } else if(board[i][j] == 2)
        {
            cout << "B";
        }
        // print out the var command.
        cout << " ";
    }
    var = var + 1;
    cout << endl;
}

```

}Even more updated print function. Not sure why you follow a zero index for ROWS but not for COLUMNS, but that has been fixed.

So the finished, neater code is like so: Code #include <iostream>
#include <cmath> // Libraries used for the project files.

using namespace std;

#define COLUMNS 6

```

/*
    0 = Empty
    1 = Red
    2 = Blue
*/

```

```

void header();
void instructions();
void printBoard(int board[][COLUMNS], int ROWS);

```

```

int main()
{
    header();
    instructions();
    int ROWS;
    do
    {
        cout << " Enter the number of rows on the board between 4 & 6, inclusively : " << endl;
        cin >> ROWS;
        if(ROWS >= 4 && ROWS <= 6)
        {
            cout << " Initializing game... " << endl;
        }
    }
}

```

```

    else
    {
        cout << "Invalid value outside range, please try again. : " << endl;
    }
} while(ROWS < 4 || ROWS > 6);

int board[ROWS][COLUMNS];
// Set all the board spots to 0; Else we will have undefined behaviour.
for(int i = 0; i < ROWS; i++)
{
    for(int j = 0; j < COLUMNS; j++)
    {
        board[i][j] = 0;
    }
}

printBoard(board, ROWS);
return 0;
}

void header()
{
    cout << "#++++++#\n";
    cout << "!    Computer Science and Engineering    !\n";
    cout << "!    CSCE 1030 Sec 002 - Computer Science I    !\n";
    cout << "#++++++#\n\n";
}

void instructions()
{
    cout << " W e l c o m e   t o   1 0 3 0   C o n n e c t ! " << endl; // Project instructions that is
    cout << " -----" << endl;
}

void printBoard(int board[][COLUMNS], int ROWS)
{
    char var = 'A';
    cout << " 1 2 3 4 5 6" << endl;
    for(int i = 0; i < ROWS; i++)
    {
        cout << var << " ";
        for(int j = 0; j < COLUMNS; j++)
        {
            if(board[i][j] == 0)
            {
                cout << "-";
            } else if(board[i][j] == 1)
            {
                cout << "R";
            } else if(board[i][j] == 2)
            {
                cout << "B";
            }
        }
    }
}

```

```
    }  
    // print out the var command.  
    cout << " ";  
}  
var = var + 1;  
cout << endl;  
}
```

Hopefully you can take it from there to implement some other functions. Remember to be careful about arrays starting at zero when grabbing user input. You'll need to subtract one from the column number the user gives and then access that array column.

I think that covered everything I saw wrong with it. Good luck.