Chess Program 6 - Move draw functionality to `Piece`.

1. Now we want to separate the concerns of drawing the chessboard, moving the parts concerning drawing the piece in `drawBoard()` into the `Piece` class.
2. To do this lets make a new method in `Piece` called `getSybolAt()` with the signature, `char getSymbolAt(int row, int column);`
3. `getSybolAt()` lives in `Piece` and returns the `char` symbol that should be drawn for that piece at a particular `row` and `column` on the chessboard. If no symbol should be drawn have `getSybolAt()` return a space (`' '`).
4. As `getSybolAt()` needs the piece constants you will now need to `#include "chess.hpp"` in `Piece`.
5. The `drawBoard()` method in `Chess` now only needs to concern itself with actually displaying the board, for specifics of what to draw for the piece it calls `getSybolAt()`.
6. The loops that iterate through row and column in `drawBoard()` should be simplified to look like this,

```
int indexRow, indexColumn;
for (indexRow = 1; indexRow <= BOARD_SIZE; ++indexRow)
{
    // print two spaces and then the row number
    cout << setw(2) << indexRow;

    for (indexColumn = 1; indexColumn <= BOARD_SIZE;
         indexColumn++)
    {
        char ch = mPtrPiece->getSymbolAt(indexRow, indexColumn);
        if (ch == ' ')
        {
            if ((indexColumn + indexRow)%2 == 0)
            {
                // black square
                cout << " -";
            }
            else
            {
                // white square
                cout << " .";
            }
        }
        else
        {
            cout << " " << ch;
        }
    }
    cout << endl;
}
cout << endl;
```

7. Notice that the `Piece` accessors `getRow()`, `getColumn()`, and `getType()` are no longer needed in `Piece` as the `Chess` instance no longer needs access to them. Typically such accessors might be needed, but for our example we can remove them.
8. Notice how there is very little exposed as `public` on `Piece` now.