

Dr. Conlon's Guidelines for Program Listings

1. Program listings should start with a heading—just as you learned in elementary school! A listing should include your name, the instructor's name, the course name, the dates of project initiation and of last modification, and the purpose of the assignment. The heading should normally consist of one or more comments, rather than of program output.
2. Identifiers must have meaningful names that explain the purpose of the variable, constant, class, object, function, or method, whenever possible.
3. If your program contains functions, methods, procedures, classes, or other types of subprograms, each of these should be labeled with its purpose.
4. Any place that the logic of your program may be hard to follow, include comments explaining what the code is doing.
5. Print and submit listings as text files. Do not process with a word processor. Use a mono-spaced font such as Courier or DejaVu Sans Mono.
6. Print directly if possible.
 - a) Unix users: use `lpr [-P<printername>] <filename>`
 - b) Windows users: print from your editor. Be sure to minimize margins first. Sometimes, `copy <filename> lpr:` may work, too.
7. Limit lines to 80 characters. For statements longer than 80 characters, continue the statement, with indentation, on the next line. If lines wrap upon printing, shorten your lines to prevent wrapping.
8. Indent to show structure. Indent in fixed increments: 3 or 4 characters is about the best. In general, if one statement is enclosed within the statement on a previous line, it should be indented relative to that previous line.
 - a) The body of an *if*, *else*, *loop*, *switch* or other such statement should be indented from the *if*, *else*, *loop*, *switch*, or other such statement.
 - b) Exception: for an *else* followed by an *if*, put *else if* both on the same line. Indent *else* or *else if* at the same indentation as the previous matching *if*.
 - c) A continued statement should be indented relative to the first line of that statement.
 - . For assignment statements, indent additional lines past the “=” of the first line.
 - . For function, procedure, or method calls, indent additional lines of parameters past the “(“ of the first line.
 - d) If you are too low on horizontal space in some parts of your listing to follow these indentation rules, indent so as to give the neatest and clearest listing you can under the circumstances. Follow the rules wherever possible in your listing.
 - e) Indent comments as much as the line above or below that they explain.
9. Remove all debugging statements once the code is working.
10. English rules count: grammar, spelling, punctuation, spacing, capitalization, and semantics. They count in comments, in strings, and in output.

```

<html>
  <!--
    Author: Michael P. Conlon
    Course: CpSc 217-01
    Instructor: Dr. Conlon
    Initiation date: August 12, 2011

    Purpose of project: Use JavaScript to generate a Web page
                        to display an ASCII chess board.

    Date of last modification: January 17, 2012
  -->
<head>
  <title>Chess Board</title>
</head>
<body id="body">
  <h1>Chess Board</h1>
  <script language="JavaScript">
    function buildLine() {
      //Create a line of tops/bottoms of the squares on the chess board.
      var text = "+";
      for (var i=0; 8>i; i++)
        text += '-----+';
      return text + "<br/>";
    }

    function buildEmptyLine() {
      //Create a line of empty chessboard squares
      // and their left and right edges.
      var text = "";
      for (var i=0; 9>i; i++) text += "|      ";
      return text + "<br/>";
    }

    //Two arrays of strings representing the chess pieces.
    var whitePieces = [ ["WR", "WN", "WB", "WQ", "WK", "WB", "WN", "WR"],
                        ["WP", "WP", "WP", "WP", "WP", "WP", "WP", "WP"] ];
    var blackPieces = [ ["BR", "BN", "BB", "WQ", "BK", "BB", "BN", "BR"],
                        ["BP", "BP", "BP", "BP", "BP", "BP", "BP", "BP"] ];

    //Start the program.
    var text = "<pre>"; //Don't let the browser reformat the text.
    for (var row=0; 8>row; row++) {
      text += buildLine();
      text += buildEmptyLine();
      if (2>row) {
        //Place the white pieces in the top two rows.
        for (var col=0; 8>col; col++)
          text += "| " + whitePieces[row][col] + " ";
        text += "|<br/>";
      } else if (row>5) {
        //Place the black pieces in the last two rows.
        for (col=0; 8>col; col++)
          text += "| " + blackPieces[7-row][col] + " ";
        text += "|<br/>";
      } else
        //It's a middle row, and should be empty.
        text += buildEmptyLine();

      text += buildEmptyLine();
    }
    text += buildLine() + "</pre>";

    document.getElementById("body").innerHTML += text;
  </script>
</body>
</html>

```