ECE 417/617 Assignment #11

Read Chapter 31 of Steve McConnell, *Code Complete*, 2nd edition, Microsoft Press, 2004.

Then answer the following questions.

<u>Chapter 31: Layout and Style</u>

130. What is the Fundamental Theorem of Formatting?

131. From Listing 31-4, what is the danger of multi-line *if* statements without curly braces?

132. What is the significance of the experiments by Chase-Simon and Schneiderman?

133. How should the start of a new "paragraph" of code be communicated?

134. Which is better: 4-space indentation or 6-space indentation?

135. C/C++ does not support pure blocks. What two alternatives remain?

136. Which of these two alternatives do you prefer, and why?

137. Why should you avoid "endline layout"?

138. Do you agree that unindented begin-end pairs violate the Fundamental Theorem of Formatting? Why or why not?

139. What is wrong with double indentation?

140. How is Listing 31-30 better than Listing 31-29?

141. What are the two ways of formatting an *if* statement without using curly braces? What is the main drawback of each?

142. How could Listing 31-33 be improved even further?

143. Why is Listing 31-36 better than Listing 31-35?

144. Study the examples in the section, "Using Spaces for Clarity." Do you agree that the whitespace increases readability in all cases? Why or why not?

145. When formatting a continuation line, do you prefer to place the operator at the end of the line or at the beginning? Why?

146. Consider the two indentation approaches shown in Listing 31-42 and Listing 31-43. What are the advantages/disadvantages of each?

147. When does it make sense to put each argument of a function on its own line?

148. McConnell advocates indenting continuation lines with the standard amount, as seen in Listing 31-45. Provide an argument that this approach could be considered a violation of the Fundamental Theorem of Formatting.

149. What recommendation did McConnell change from the first edition of the book, and why?

150. What surprising result was discovered when comparing the computational efficiency of Listings 31-50 and 31-51?

151. Do you prefer placing one variable declaration per line, or allowing multiple declarations per line? Justify your answer. See Listings 31-53 and 31-54.

152. Why is it important to keep variables live for as short a time as possible?

153. What is an additional benefit of placing each declaration on its own line, when the variables are pointers?

154. What is meant by “overformatting”?