Readings for this quiz: Lakos, Ch.0; Brooks, Ch.1-4.

NOTE: These are sample questions. This is not a sample quiz. Thus, the length of your quizzes will not be as long as this. These are just the questions I threw out so I could fit the quiz inside five minutes.

GRADING: Quizzes are graded on a pass/fail basis.

1. According to Brooks, what is “conceptual integrity?”

   Answer:
   If it sounds like they’ve read ch. 4 of Brooks, then it receives full credit.

2. What is an issue that Brooks brings up when trying to achieve conceptual integrity?

   Answer:
   Either one of these would work.
   - “Aristocracy and Democracy” (with some description). (pg. 44)
   - What do the implementers do while waiting for an architecture team to finish its specs? (pg. 47)

3. According to Brooks, what is a man-month? Name one thing wrong with the man-month.

   Answer:
   A man-month is simply the results that a single person can produce in a single month. (Have you noticed that Brooks assumed that all engineers and programmers were men?)

   Some possible answers for the second part include:
   - Interchangeability of men and months: “Men and months are interchangeable commodities only when a task can be partitioned among many workers with no communication among them.” (pg 16)
   - Communication/coordination overhead. (pg. 17)
   - Initial training time is not figured in when adding extra manpower to an ongoing project. (pg. 18)
   - Anything else that makes sense is valid. However, they need to mention at least on issue brought up by Brooks. After all, the question asks, “According to Brooks . . . ”

4. True or False: Contrary to popular belief, objected-oriented programs in their most general form are fundamentally more difficult to test and verify than their procedural counterparts.

   Answer:
   TRUE. This is a direct quote from Lakos, section 0.2, page 3.

   Note: This might be a little too obscure for a quiz question.
5. Choose the best answer. An “excessive link-time dependency” is:

a) when you have a static archive, the constant factor for the link time increases disproportionately to the size of the archive.

b) linking components that give a small amount of functionality, but bloating the executable size disproportionately to the gain in functionality.

c) including lots of useless .h files in your header and .c files.

d) none of the above

**Answer:**

→ “linking components that give a small amount of functionality, but bloating the executable size disproportionately to the gain in functionality.”

“If you have attempted to link to a small amount of functionality in a library and found that your time to link has increased disproportionately to the benefit you are deriving, then you may have been trying to reuse heavy-weight rather than light-weight components.”

Basically, your code gets bloated because the String class you included has over a hundred functions, even though you just wanted to some a couple basic functions (Lakos pg. 5-6). Note that there’s a difference between a compile-time dependency (#includes, etc.) and link-time dependency.