John Hellrung

CSC 266

1. In a paragraph or more, describe which of the many contributing factors within the bug's origins strikes you as the "most preventable" and why? Of course, this will be in hindsight. Explain.

In the article by James Gleick, I believe the most preventable bug is memory conservation. You, see if only the programmers would have try to tested and looked at the values inputted in there code like the watches in PyScriper. The issue with the programmer’s code was there inputted 64-bit data into a 16-bit space, would have been solved. I can’t see why the programmer’s didn’t find the bug, when debugging there code.

Now, I believe that second bug, “failsafe bug” and the first bug, “memory conservation” are in a tie for the preventable bug. The failsafe bug was that programmer’s beginning was off because setting up the program as a copy program is a terrible way to have a failsafe, when the first part of the program fails. I believe they did not even try to test to see if there failsafe would work all different situation especially when we talk about million dollars projects.

1. The investigative report that uncovered the bug stated that software "does not fail in the same sense as a mechanical system." Do you agree or disagree with this statement? Explain in a paragraph

Yes, in the text, I believe that they mechanical at the logic level which is very similar situation to saying they fail like a mechanical system. You, see when you first start with a logic level, you are designing a mechanical system in a language that is easy to understand. Now, I believe that I lot these bugs would have been solve be working hard on step one fleshing out all the numbers of way to go about make the code until they have a code with a reliable failsafe and a better method rather than conserving 64-bit data to 16-bit data spaces.