Michael Esparza
CS 285 – HW5

Question 2:
What are the exploited data structures?
Freetype Type 1 font parser, interpreter for Type 1 font programs.
Operand/ result stack
Two variables x and y
Decoder->buildchar array

Comex used these data structures to manipulate the pointer and execute his payload by overwriting various data structures and triggering a call to the function pointer that he caused to point out of its intended bounds.

Give a high level description how the attack works:
The general idea of the code is to use an exploit, in this case a missing check on the argument count parameter for the callothersubr operation. This allowed comex to create a malicious font program that moved the stack pointer of the interpreter outside of its intended bounds; this provided read/write access to various fields within the T1_DecoderRec structure. This means he could execute his own code by tricking the stack pointer into pointing to his own code. The entry point to his code is through a font program that renders a single @ within a PDF. The font program is used to render this character and is the entry point.

How could the code be fixed?
We must examine the source of the problem in this case. Adobe small interpreter programs are used to render character outlines for PDFS and comex was able to take advantage of this by manipulating the stack pointer. To prevent this Adobe must rethink their security, especially for the prices that they request of their customers. Maybe before running any code within their programs they should check the hash value or the signed value of the code to ensure that it has not been tampered with, as many vendors and distributors do now. These can be seen as MD5 checksums whenever you download a service or program.