Defensive Programming

A. Defensive programming:
   a. Definition: Designing and implementing software so that it continues to function even when under attack.
   b. Why is it difficult?
      i. Programmers often make assumptions about the type of inputs to a program
      ii. Conflict with business pressures
   c. Programmers now need mindset that involves all input being dangerous

B. There are multiple types of tests that a programmer can make
   a. Unit testing, black box testing.
   b. Security testing is very important but usually overlooked
   c. Russ’ rule: 10% of the program is functionality, 90% should be testing.

C. Safety is important and still relatively new for software engineers
   a. If an interface is defined to be only used by you, why would you bother implementing safety. But it is necessary.
      i. Very tough to think of these cases when something only meant for you is called by an unfriendly program

D. Incorrect Handling is something we have to look out for (Input) - This is common
   a. end user
      i. String (username)
      ii. Files
   b. Interpretation of program input (bad data types)
   c. Characters (ASCII, hex, or numerical? All different)

E. Other
   a. SQL Injection
   b. Code injection (Shell code)
   c. Cross site scripting
   d. XSS reflection
      i. Unsanitized user input can trigger embedded code
F. Defense
   a. Input fuzzing testing
      i. Feeding randomized input to see flaws in programs.
   b. Helper programs to look for flaws