➡ Familiarity with relevant vulnerability classes

➡ Modularity - separate modules for separate functionalities

- Sanitize, validate, restrict input data even between modules or components (mutual suspicion)

- Be "fault tolerant" by having a consistent policy to handle failure

- Use reputable, security conscious and well maintained libraries

- Adopt good programming practices, be security aware

Secure Programming

Secure Programming

- Familiarity with relevant vulnerability classes
- Modularity - separate modules for separate functionalities
- Sanitize, validate, restrict input data even between modules or components (mutual suspicion)
- Be "fault tolerant" by having a consistent policy to handle failure
- Use reputable, security conscious and well maintained libraries
- Adopt good programming practices, be security aware

Software Security Assessment