

Defensive programming (2)

Being security aware programmer

- ✓ Check the inputs, even between components that belongs to the same application (mutual suspicion)
- ✓ Be “fault tolerant” by having a consistent policy to handle failure (managing exceptions)
- ✓ Reuse known and widely used code by using design patterns and existing libraries

Penetration Testing

Testing the functionalities

- ✓ Unit test, Integration test, Performance test and so on ...

Testing the security

- ✓ Penetration tests

- ➔ Try to make the software fail by pushing the limits of a “normal” usage i.e test what the program is not supposed to do