- Unit testing- testing small pieces of code
- system testing- testing more code like a complete api
- Integration testing- testing how code works together
- Acceptance testing- sees if end users can use the code
- Performance testing- see if the application is fast
- Regression testing- basically all of above
- security testing- test if application is secure
- Load testing- test if application is able to handle the traffic
- end-to-end testing- testing the application as a whole
- TDD- test driven development
  - Write test first
  - Should fail to think before implementing
  - Tests are not skipped by time pressure
  - Spots bugs
- Write failing code > implement code to make test pass > refactor (make sure the code is written well) and repeat
- Unit testing Advantages:
  - Validate small units of software
  - Finds bugs early and easy
  - Saves time and money
  - Force developers to write better code
- JUnit- testing framework for java (part of xUnit series for all coding languages)
- JUnit is must have for TDD
- Assertions.assertEquals(expected,value) @test before class
- JUnit Jupiter
- Annotations- @__ before code
- @DisplayName() to change name of test
- Assumptions- settings condition for executing test, when met test is executed
- Dependency injections (@Inject)
- Surefire plugin
- EnableOn___/DisableOn___
- Disable test by adding @Disabled