**Java Type System**

* Everything is an Object.
  + Except int, float, double etc. and null.
* Assignment b = a is valid if actual type of a is a subtype of declared type of b

You can only extend one class.

Exception is an object.

Serializable = you can write something out of the file.

What is an invariable?

**Defensive Programming**

* What error checking would you have in BankAccount in first year?

**Programming by Contract**

* Pre and post-conditions. Examples:
  + Deposit amount has to be +ve
  + Minimum amount to deposit $800
* Class invariant
  + Condition on object state
  + Holds after constructor and before and after each method call- but not during method execution
  + Reasoning: If precondition and invariant hold before method call, post-condition and invariant hold afterwards.

**Pre and Postconditions**

* Conditional contract
* **If** the preconditions holds before am ethod call, **then** the postcondition holds after the method terminates

**Test Driven Development**

* **First** define a set of tests for all methods, **then** develop code to pass each test
* Is this a good idea?
* **TDD does not just mean “I tested my code”**
* Code has to be consistent with pre- and post-conditions
  + If you say something is a PRE-CONDITION, you don’t need to test it
  + If you say something is a POST-CONDITION, you MAY need to check it
* Your code must be consistent with you specification
* Test driven development in C:
  + First thing is to write tests
  + Second thing is to write code to pass all the tests
  + PROS: Gets you to think about tests
  + CONS: More about passing tests, less about logic behind it
* Good test-driven development requires you to write tests

**Assertions**

* Java assertions are different to C assertions
* You should not be using assertions to do anything serious
* In our exams/tests, markers will ignore assertions.
* Assertions are not a substitute for reasoning.
* J-unit assertions are different to Java assertions

**Processes to design a big system:** Important for our first assignment

* How do we design a system that has multiple classes in it.

**Assignment 1 will be due in week 6**