Assertions, pre/post-conditions and invariants

Section 4.2 in Savitch (assertions)

Programming as a contract
- Specifying what each method does
- Precondition
  - What is assumed to be true before the method is executed
  - Caller obligation
- Postcondition
  - Specifies what will happen if the preconditions are met
  - Method obligation

Class Invariants
- A **class invariant** is a condition that all objects of that class must satisfy while it can be observed by clients
- What is an invariant of the Account class?

What is an assertion?
- An **assertion** is a statement that says something about the state of your program
- Should be true if there are no mistakes in the program

```java
// n == 1
while (n < limit) {
    n = 2 * n;
}
// n >= limit
// n is the smallest power of 2 >= limit
```

```
assert
Using assert:
assert n == 1;
while (n < limit) {
    n = 2 * n;
} assert n >= limit;
// n is the smallest power of 2 >= limit.
```
When to use Assertions

Another example

```java
if (i % 3 == 0) { ... } else if (i % 3 == 1) { ... } else { // We know (i % 3 == 2) ... }
```

We can use assertions to guarantee the behavior.

```java
if (i % 3 == 0) { ... } else if (i % 3 == 1) { ... } else { assert i % 3 == 2 : i; ... }
```

Control Flow

If a program should never reach a point, then a constant false assertion may be used

```java
void foo() { for (...) { if (...) return; } assert false; //should never get here }
```

When to use assertions?

- Programming by contract
  - Preconditions in public methods (values of parameters) should be enforced rather than asserted
- Postconditions
  - Assert post-condition

Performance Problems

- Assertions may slow down execution
  - For example, if an assertion checks to see if the element to be returned is the smallest element in the list, then the assertion would have to do the same amount of work that the method would have to do
- Therefore assertions may be enabled and disabled
- Assertions are, by default, disabled at run-time
  - In this case, the assertion has the same semantics as an empty statement
Don't use assertions to flag user errors

Assertions in Eclipse

- Go to Preferences -> Java -> Compiler and set the Compiler Compliance Level to 1.5 or 1.6. Also check Use Default compliance settings. This tells the compiler to recognize and allow assert statements, but does not enable them.
- To enable assert statements, you must set a compiler flag. Go to Run -> Run Configurations -> Arguments, and in the box labeled VM arguments, enter either -enableassertions or just -ea

More Information

- For more information:
  
  http://java.sun.com/j2se/1.4.2/docs/guide/lang/assert.html