“And what is good, Phaedrus,
And what is not good ~
Need we ask anyone to tell us these things?”

(Plato, Phaedrus but also the epigraph to Robert Pirsig’s “Zen and the Art of Motorcycle Maintenance”)

Let’s steal co-opt this phrase and apply it to running a java program…
“And what is correct, BlueJ, and what is not correct ~ need we ask anyone to tell us these things?”

Answer: No. We don’t have to ask anyone. We can find out for ourselves. How? By testing... or by using a tester class to tell us what is correct.
Verification of code requires a

- Verification of code requires a Test-First design
  - ideally we should write tests for the methods we write *before* we write the actual methods
  - if we can’t write the tests for how a method should behave, how can we possibly begin to write the method?
  - this is the only way to test what we *expect* the method to do, rather than what we expect from our (possibly incorrect) implementation!

- Importance of test suites
  - BlueJ is great for providing us a way to interact with the classes and objects we create
  - we can interactively inspect our objects and test individual methods, but
  - we don’t want to have to manually rerun our tests every time we change our code!
  - the Tester library permits us to develop test suites that we can automatically re-run

- We’ll use a tester.java library to test our code
  - Similar to junit (for java) and cppunit (for c++)

- Test is quality assurance and a discipline on its own. Consider the following:
  - Unit test
  - Function test
  - System test
  - Regression test

On to the lab!