Benjamin Tate
Michael Namoff
Jason Zmuda
CS 362 -- Section 400
Final Project Part A

Explain testIsValid function of UrlValidator test code:

- testIsValid() works by using Java's UrlValidator class
- An array of ints, testPartsIndex, is looped through, and the corresponding index of the testObjects array passed to the function is tested for validity
- The URLs are built by appending the item members of each object in the testObjects array passed to the function
- If the URL is valid, the url is printed
- The printed int variable is then incremented to keep track of how many URLs have been printed
- Once printed reaches statusPerLine (60), it is reset to 0

Give how many total number of the URLs it is testing:

- The number of URLs being tested is the product of the number of URL parts in each of the members of the testObjects array (testUrlScheme, testUrlAuthority, testUrlPort, testPath, and testUrlQuery).
- Number of URLs:
  - testUrlScheme = 9
  - testUrlAuthority = 19
  - testUrlPort = 7
  - testPath = 10
  - testUrlQuery = 3
  - **Total = 9 * 19 * 7 * 10 * 3 = 35,910 URLs**

Explain how it is building all the URLs:

- The URLs are built by appending the item members of each object in the testObjects array passed to the function.

Give an example of valid URL being tested and an invalid URL being tested by testIsValid() method:

- Valid:
  - http://www.google.com:80/test1?action=view
- Invalid:
  - 3ht://www.google.com:65a/..?action=view

Do you think that a real world test (URL Validator's testIsValid() test in this case) is very different than the unit tests that we wrote (in terms of concepts & complexity)? Explain in few lines:

- The unit tests within the URL Tester are not very unlike the unit tests we wrote for dominion. The idea is to pick specific instances of input and/or output and assert that the intended behavior holds throughout the execution of the program or function. The code within the URL Tester, however, definitely achieved a higher level of complexity. By implementing a vast array of permutations of URLs, a large number of test cases can be tested in very short order.