Final Project Part A CS362 F17
Logan Wingard - wingarlo
Lilliana Watts - wattsli
Davian Lukman - lukmand

1. Explain testIsValid Function of UrlValidator test code.
   testIsValid function calls two other functions, which are
   testIsValid(testUrlParts, UrlValidator.ALLOW_ALL_SCHEMES);
   - Create a new UrlValidator object
     //UrlValidator urlVal = new UrlValidator(null, null, options);
   - Asserts http://www.google.com and http://www.google.com/ is valid
     //assertTrue(urlVal.isValid("http://www.google.com"));
     //assertTrue(urlVal.isValid("http://www.google.com/"));
   - Do while the url is still valid loop
     //while(incrementTestPartsIndex(testPartsIndex, testObjects));
   - Create a new StringBuffer object
     //StringBuffer testBuffer = new StringBuffer()
   - Combine separate url into a single url
     //ResultPair[] part = (ResultPair[]) testObjects[testPartsIndexIndex];
   - Call isValid function and store the value in boolean result
     //boolean result = urlVal.isValid(url);
   - Compare the result with expected result
     //assertEquals(url, expected, result);
   - If the result is true, print the url

   And Setup();
   - Makes the index of every url to 0

   2. Give how many total number of the URLs it is testing
   - testUrlScheme = 9
   - testUrlAuthority = 19
   - testUrlPort = 7
   - testPath = 10
   - testUrlQuery = 3
     Possibilities = 9*19*7*10*3 = 35910 urls

   3. explain how it is building all the URLs.
   A complete URL is composed of a scheme, authority, port, path, and query. Each part is stored in an array called ResultPair. Each of the array element contains a string of the parts and a boolean value (true or false). The function will loop for every combination of the resultPair arrays. All of the URLs must be individually valid for the entire URL to be considered valid. In order to determine true or false, the function use AND operation to match the boolean value.

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

Valid: http:// go.com:80/test1?action=view

Invalid: 3ht://www.google.com/test1?action=view

5. . 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)

If we compare it with the dominion code, it feels pretty similar. First,we create a state or an object. Write various code to test various variables. The code will create an output and what to be expected.