ChainedHashTable Test Plan - initial

**Constructors –**

**public** ChainedHashTable(**int** capacity, HashProvider<K> hashProvider)

success path:

* positive capacity results in a correctly-sized array within the ChainedHashTable

negative path:

* zero capacity results in an empty array created within the ChainedHashTable
* negative capacity results in an empty array created within the ChainedHashTable

getters/setters w/business logic – none

**public methods:**

**public void** put(K key , V item)

**public** V get(K key)

**public void** remove(K key)

success path(s)

* put new key/item to empty ChainedHashTable, must be able to retrieve item via key
* put new key/item to non-empty ChainedHashTable, must be able to retrieve item via key
* get existing item via key from non-empty ChainedHashTable, validate item is not removed
* remove item via key from non-empty ChainedHashTable, validate item is no longer accessible via key
* attempt to add an item to a non-empty ChainedHashTable using existing key should overwrite existing object (value), confirmed by calling get( ) using key

failure path(s)

* attempt to retrieve item via get( ) from empty ChainedHashTable should return null
* attempt to retrieve item via get( ) from non-empty ChainedHashTable using invalid key should return null
* Question: should a ChainedHashTable allow a put( ) with a null value object, and then actually have a Pair with a valid key and a null object?