All of these methods were utilized with a HashSet which has best case time complexity of:

HashSet	**Best**
O(1)	add
O(1)	getMatch
O(1)	remove

HashSet	**Worst**
O(n)	add
O(n)	getMatch
O(n)	remove

These time complexities, worst case related to a situation in which the whole array needs to be indexed over in order to find/remove/add/getMatch the requisite element (e.g. initially hashes to index 3 in a last of 5, has to go through indices 3,4,5,0,1,2 before it sees 2 has element/is null for adding element).

The following methods use the “add” method and have the above time complexity:
setAbilityScore, addSkill

The following methods use the “getMatch” method and have the above time complexity:
getAbilityScore, getAbilityModifier(uses getAbilityScore), checkAbility (uses getAbilityModifier), getSkillRanks, getRelatedAbility

Use both “add” and “getMatch”:
setSkillRanks(just additive n, not multiplicative)
getSkillModifier(additive n from calls to getSkillRanks and getAbilityModifier)
checkSkill(uses getSkillModifier)