

Character	
-id	int
-baseAttack	int
-baseDefense	int
-headGear	headgear
-handGear	handgear[]
-footwear	footwear[]
+Character(id, baseAttack, baseDefense)	
+pickItem()	
+getBaseAttack()	
+getBaseDefense()	
+getTotalAttack()	
+getTotalDefense()	
+getHeadGear()	
+getHandGear()	
+getFootwear()	

Gear <<Interface>>	
+combine(gear)	
+getGearAttackAmount()	
+getGearDefenseAmount()	
+getName()	
+getAdj()	
+getNoun()	

AbstractGear <<Abstract>>	
-adj	string
-noun	string
-attackAmount	int
-defenseAmount	int
+AbstractGear(adj, noun, att, def)	
+toString()	
-newGear(adj, noun, att, def)	
-checkType(gear)	

HandGear	
+HandGear(id, name, type, capability)	

FootWear	
+Footwear(id, name, type, capability)	

HeadGear	
+HeadGear(id, name, type, capability)	

90-8	Description	Method name	Test case	Expected result	Actual result
Character	Testing initialization normal case	Character()	new Character(1, 20, 20)	normal initialization	normal initialization
	Testing get base defense amount	getBaseDefense()	assertEquals(20, obj.getBaseDefense())	TRUE	TRUE
	Testing get base attack amount	getBaseAttack()	assertEquals(20, obj.getBaseAttack())	TRUE	TRUE
	Testing the output for priority of picking items based on: - Available slots - Attack strength - Defense strength	pickItem()	assertEquals("Strong Helmet -- defense strength: 5", Character.pickItem())	TRUE	TRUE
	Testing combining items	pickItem()	assertEquals("Huge Hat-- defense strength: 3", Character.pickItem()) assertEquals("Huge, Strong Helmet-- defense strength: 8", Character.getHeadGear())	TRUE	TRUE
	Testing get gears	getHeadGear()	assertEquals("Huge, Strong Helmet-- defense strength: 8", Character.getHeadGear())	TRUE	TRUE
Name	Testing get total defense amount	getTotalDefense()	assertEquals(28, obj.getTotalDefense())	TRUE	TRUE
	Testing initialization normal case	Name()	new Name("Scurrying", "Happy HoverBoard")	normal initialization	normal initialization
	Testing initialization abnormal case	Name()	new Name("Scurrying", null)	throw IllegalArgumentException	throw IllegalArgumentException
	Testing the adjective output of a gear	getAdjective()	assertEquals("Scurrying", obj.getAdjective())	TRUE	TRUE
	Testing the noun output of a gear	getNoun()	assertEquals("HoverBoard", obj.getNoun())	TRUE	TRUE
	Testing for full name output a gear	getName()	assertEquals("Scurrying, Happy HoverBoard", obj.getName())	TRUE	TRUE
HeadGear	Testing initialization normal case	HeadGear()	new HeadGear(1, "Strong Helmet", HELMET, {DEFENSE: 5})	normal initialization	normal initialization
	Testing initialization abnormal case with illegal head gear type	HeadGear()	new HeadGear(1, "Strong Helmet", BOOT, {DEFENSE: 5})	throw IllegalArgumentException	throw IllegalArgumentException
	Testing initialization abnormal case with illegal capability	HeadGear()	new HeadGear(1, "Strong Helmet", HELMET, {ATTACK: 5})	throw IllegalArgumentException	throw IllegalArgumentException
	Testing get gear defense amount	getGearDefenseAmt()	assertEquals(5, obj.getGearDefenseAmt())	TRUE	TRUE
HandGear	Testing initialization normal case	HandGear()	new HandGear(1, "Nice Sword", SWORD, {ATTACK: 5})	normal initialization	normal initialization
	Testing initialization abnormal case with illegal head gear type	HandGear()	new HandGear(1, "Nice Sword", BOOT, {ATTACK: 5})	throw IllegalArgumentException	throw IllegalArgumentException
	Testing initialization abnormal case with illegal capability	HandGear()	new HandGear(1, "Nice Sword", SWORD, {DEFENSE: 5})	throw IllegalArgumentException	throw IllegalArgumentException
	Testing get gear attack amount	getGearAttackAmt()	assertEquals(5, obj.getGearAttackAmt())	TRUE	TRUE
Footware	Testing initialization normal case	Footware()	new Footware(1, "Scurrying, Happy Hoverboard", HOVERBOARD, {ATTACK: 5, DEFENSE: 5})	normal initialization	normal initialization
	Testing initialization abnormal case with illegal head gear type	Footware()	new Footware(1, "Scurrying, Happy Hoverboard", SWORD, {ATTACK: 5, DEFENSE: 5})	throw IllegalArgumentException	throw IllegalArgumentException
	Testing get gear defense amount	getGearDefenseAmt()	assertEquals(5, obj.getGearDefenseAmt())	TRUE	TRUE
	Testing get gear attack amount	getGearAttackAmt()	assertEquals(5, obj.getGearAttackAmt())	TRUE	TRUE
HandGear/Footwear/HeadGear	Testing the output for combining same type of gear	combineGear()	assertEquals("Scurrying, Happy HoverBoard -- defense strength: 5, attack strength 5", scurryingSandles.combineGear(happyHoverboard))	TRUE	TRUE
	Testing combining with abnormal input	combineGear()	scurryingSandles.combineGear(happyHelmat)	throw IllegalRunTimeException	throw IllegalRunTimeException