



TEST PLAN

Testing calculateBasicAbilities()	Input Value	Expected Value
Check if the abilities are properly calculated	calculateBasicAbilities()	pass
Check if the abilities are within 6 to 18 after rolling dices	calculateBasicAbilities()	IllegalArgumentException
Check if dices are rolled again if any 1's are rolled	calculateBasicAbilities()	illegalStateException

Testing calculateHealth()	Input Value	Expected Value
Check if the value of health is returned correctly	calculateHealth()	pass
Check if the value returned is actually the sum of the 4 basic abilities	calculateHealth()	IllegalArgumentException
check if the health is not negative	calculateHealth()	IllegalArgumentException

Testing gearDetails()	Input Value	Expected Value
Check if the gear is assigned properly	gearDetails()	pass
Check if the gear is not over the limits	gearDetails()	IllegalArgumentException
Check if gear and the number of gear are returned properly	gearDetails()	illegalArgumentException

Testing StrikingPower()	Input Value	Output Value
Check if the striking power is returned properly	StrikingPower()	pass
Check if it isn't negative	StrikingPower()	illegalArgumentException

Testing avoidaneAbility()	Input Value	Output Value
Checking if the value is returned properly	avoidaneAbility()	pass
Value shouldn't be negative	avoidaneAbility()	illegalArgumentException

Testing ifStrike()	Input Value	Expected Value
Checking if the desired output is displayed	ifStrike()	true

Testing potentialStrikingDamage()	Input Value	Expected Value
Check if the value is returned properly	potentialStrikingDamage()	pass
Check if the value isn't negative	potentialStrikingDamage()	illegalArgumentException

Testing actualDamage()	Input Value	Expected Value
Value returned properly	actualDamage()	pass
Not negative	actualDamage()	illegalArgumentException

Testing firstStrike()	Input Value	Expected Value
Correct player returned	firstStrike()	pass

Testing getWinner()	Input Value	Expected Value
Check if the value returned is correct	getWinner()	pass
Shouldn't return a player's name if the game is not over	getWinner()	IllegalArgumentException

Testing printGear()	Input Value	Expected Value
Check if the gear are printed from top to bottom	printGear()	pass
Check if the gear are printed in alphabetical order	printGear()	pass

Testing deductDamage()	Input Value	Expected Value
Check if the damage is deducted from the players health	deductDamage()	pass
Once the player's health is below or equal to 0, game should be over	deductDamage()	illegalStateException