Test ID	Method Tested	Parameters (Test Data)	Test Rationale	Expected Change	Actual Change
1	setWackyString	String string = “”	Here I will be testing the Exception Handling by passing in a null string.	IllegalArgumentException The string will be set to nothing	The string is set to the empty string It Works
2	setWackyString	String string = “Hello”	Here I will be testing if the string “Hello” is correctly assigned	this.string = Hello	this.string = Hello It Works
3	getWackyString()	String string = “”	Here I will test if the string is empty	The method should return an empty string	This method returns the empty string It Works
4	getWackyString()	String string = “Hello"	Here I will test with the strings “Hello”	The method should return Hello	The method returns Hello It Works
5	getFirstMiddleLast	String string = “Love"	Here I will test this method with an even length string “Love” to make ensure that it can handle with an even string.	The method will return Loe	The method returns Loe It Works

6	getFirstMiddleLast	String string = “Yo"	Here I will test this method with a string that has a length of 2 (“Yo”) to ensure that there is no middle char returned assigned.	The method should return Yo	The method returns Yo It Works
7	getFirstMiddleLast	String string = “Hello"	Here I will be testing this method with an odd length string “Hello” to endure that the method can handle a string of an odd length.	The method should return Hlo	The method returns Hlo It Works
8	getFirstMiddleLast	String string = “”	Here I will be testing this method with a empty string to ensure that the method can handle an empty string	The method should return the empty string “”	The method returns the empty string ”” It Works

9	getEveryThird Character	String string = “Welcome"	Here I will be testing an odd length string to ensure that this method can find every third character of an odd length string.	This method should return lm	The method returns lm It Works
10	getEveryThird Character	String string = “everyThirdChar”	Here I will be testing a longer odd length string to ensure that this method can find every third character of an odd length string.	This method should return eTrh	The method returns eTrh It Works
11	getEveryThird Character	String string = “Love”	Here I will be testing an even length string to ensure that this method can find every third character of an even length string.	The method should return v	The method returns v It Works

12	getEveryThirdCharacter	String string = “getEveryThirdCharacter"	Here I will be testing a longer even length string to ensure that this method can find every third character of an even length string.	The method should return teTrhae	The method returns teTrhae It Works
13	getEveryThirdCharacter	String string = “”	Here I will be testing an empty string with this method to make sure it can handle it.	The method should return the empty string	The method returns the empty string It Works
14	countEvenDigits	String string = “123hello456"	Here I will test a string that has a mixture of letters and even and odd numbers to see if the for loop works to find the numbers and then which ones are even	The method should return 3 (2,4,6)	The method returns 3 It Works
15	countEvenDigits	String string = “13579"	Here I will be testing the method with a string of only odd numbers to check and make sure that the % 2 works.	This method should return 0	The method returns 0 It Works

16	countEvenDigits	String string = “Hello”	Here I will be testing the method with a string that only has letters to make sure that the isDigit works.	This method should return 0	The method returns 0 It Works
17	countEvenDigits	String string = “2468"	Here I will be testing the method with a string that is composed of only even numbers to make sure that the % 2 is working.	This method should return 4	The method returns 4 It Works
18	isValidEID	String string = “V00123456"	Here I will be testing this method with what is a valid ID as it starts with V and is followed by 2 zeros and has a length of 9	This method should return true	The method returns true It Works
19	isValidEID	String string = “U00123456"	Here i will be testing this method with a non valid ID as it doesn’t start with a V	This method should return false	The method returns false It Works

20	isValidEID	String string = “V10123456"	Here i will be testing this method with a non valid ID as it’s second element isn’t a zero to check if it returns false	This method should return false	The method returns false It Works
21	isValidEID	String string = “V01123456"	Here i will be testing this method with a non valid ID as it’s third element isn’t a zero to check if it returns false	This method should return false	The method returns false It Works
22	isValidEID	String string = “V0012345"	Here i will be testing this method with a non valid ID as it’s length is not 9 to check is it will return false	This method should return false	The method returns false It Works
23	isValidEID	String string = “V00e23456"	Here I will be testing this method with a string that has a letter in it after V00. This test will check to see if this method can detect a letter.	This method should return false	The method returns false It Works

24	isValidEID	String string = “V00123e56"	Here I will be testing the same thing as above but with the letter in a different place.	This method should return false	The method returns false It Works
25	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = 2 endPosition = 5	Here I will be testing this method with a string that contains a zero and a letter in the indexes between the starting position and the ending position to ensure that the method won’t spell/leave either of them out	This method should assign the instance variable too 0ONE0hONE2	This method assigns the instance variable to 0ONE0hONE2 It Works
26	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = -1 endPosition = 4	Here I will be testing if the Exception Handling regarding if the starting position is less than zero works.	This method should throw MyIndexOutOfBoundsException	This method throws MyIndexOutOfBoundsException It Works

27	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = 2 endPosition = -1	Here I will be testing if the Exception Handling regarding if the ending position is less than zero works.	This method should throw MyIndexOutOfBoundsException	This method throws MyIndexOutOfBoundsException It Works
28	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = 7 endPosition = 5	Here I will be testing if the Exception Handling regarding if the starting position is greater than the length of the string works.	This method should throw MyIndexOutOfBoundsException	This method throws MyIndexOutOfBoundsException It Works
29	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = 2 endPosition = 8	Here I will be testing if the Exception Handling regarding if the ending position is greater than the length of the string works.	This method should throw MyIndexOutOfBoundsException	This method throws MyIndexOutOfBoundsException It Works
30	convertDigitsToWordsInSubstring	String string = “010h12” startPosition = 5 endPosition = 4	Here I will be testing if the Exception Handling regarding if the starting position is greater than the ending position works.	This method should throw IllegalArgumentException	This method throws IllegalArgumentException It Works