ID	Description	Steps	Expected	Actual	Result	Comment
C1	Test Constructors of the Contract class	1. Create an object using constructor 2. Pass the arguments manually to the constructor 3. Using the toStringmethod check the values are initiated	Object should create and only public methods should be accessible And the private attributes should be initiated with the given values	Object was created. Private attributes can not be accessible. The result has all values are as expected	Passed	In here Contract class has toString override method. This is used only for the testing.
C2	Test getter methods of the Contract class	1. Create an object of the Contract in Main method 2. Using the system print function print the values that will return from each function	Console results should be same as the values that passes to the constructor.	Values are same as the arguments that passes to the constructor	passed	
C3	Test setter methods of the Contract class	1. Follow the steps in the C2 2. The using the setters change the values of the attributes 3. Then again print the results to the console using getters	After changing the values using setters, the next printing stage should print the new values	New printing stage prints the newly changed values	Passed	
U1	Test the method validNextInt(). This method is used to get and validate input of int from the user	1. Give prompt message and min range and max range (Different values)	• When the given input is not in the range it should print an error message.	• It prints the different error message according to the out-of-range criteria and invalid numbers	Passed	This will keep looping until user enters a valid number

		2. Then call method in testing 3. Give invalid numbers, characters Strings 4. Finally give the valid range value and output that value	• If the non -integer value is given again print a message • When a correct value is given it should return that value	• If the Correct number is given that value was returned		
U2	Check the YNtoBoolean(). This will get an character. If it is 'y' or 'Y' then return true false otherwise	1. Call the function with different characters and print the result 2. Give the value 'y' and 'Y' and print and check the results	Given characters except the Y and y Should print false. When it is given Y or y it should print true	Was printed false for any characters except Y or y Print true for Y or y is given	Passed	This method only checks the character Y or y. It will not consider the N or n. Before pass the characters, they should be validated
U3	Check the functionality of the validateStringLength() method. Thins function is to get and validate the string and catch if exceed the given limit Also the validateStringFixedLength()	1. Prompt and set the max length to 25 2. Call the functions with string that includes 26 and more characters and check the output 3. Given string with characters up to	When user give 26 or more characters it should give error message that indicating the maximum length is exceeded. When it is given 25 or less than that it should return that given string excepts the errors. When the max length is change it should print error message when input is exceeded the max length.	When it is given input less or equal to the max length it returned and print the input When any input exceeds the limit, it gave an error message When using fixedStringLength() function it gave true if	Passed	

		24 characters and check the output 4. Given exactly 25 length string and print the result 5. Change the max length value and repeat the process 6. When testing fixed length function same process repeated	Otherwise return the input. validateStringDixedLength() should give true if and only if the input is math to the limit	and only if the input is exactly match with the length		
U4	Test functionality of the characterIsInArray() function. This function will get an character and array of characters as argument. Return true if given character is in the array false otherwise	1. Create an array of characters 2. Call the function with different character and that array 3. Print the output to the console and check the functionality 4. Repeat 1 -3 for different character array	If the given character is contained in the array should return true. Otherwise, false.	When it is given a character which is not inside the array it prints false. When it is given a character which is inside the array it prints true	Passed	
U5	Test the functionality of the validNextChar() function. This function will prompt the user a given message ,get and validate the input. If the given character is not in the expected character list it shows error and prompt again	1. Give the prompt message and the set of characters array 2. Hold the output in a char variable 3. Print the that char variable	Function should print the given prompt string. When the user give a character which is not inside that array this should print an error message and again user is asked to enter a character.	Given prompt message is printed before get the input from the user. When a character from the outside of the array is given it prints an error messagand	Passed	This method is mostly use when There is character input from the user (Y/N)

		4. Then user will prompt to give a character 5. Gave different characters	If the character which is inside the array is given that character should print to the console	again asked to enter a character. If the givena character is correct That character is printed to console		
U6	Test the functionality of the truncateString() Method. This method takes a string and index of the string where the truncate if begin and length to indicate to what length	1. Give a string to the function and start and length value 2. Then store the resultant string to a variable 3. Print that string to the the console 4. Repeat the process for different strings and different start and length values	When the string is given if the value given to the start is negative, length is negative, start + length exceed the given string length, Then it should returns the String as it is. If the start and the length is valid it should print the truncated string of the given string[start , start + length]	When the start or length is invalid it gives the original string that passed to the function. When a valid start and the length is given truncated string is given	Passed	This a Utility function. In here only care about the truncated string when the valid start and length is given.
U7	Test the functionality of appendStringToFile() function	1. Give a filename (both existing and non - existing) and the string that should append to the file 2. Check the file 3. Again call the same filename and different string that need to appnd 4. Check the file is created and results	When the function is called for the first time and If the given file name is not exist after the call a file should be created. The content of that file is the string that passed to the function. When the function call again with different string it should append that string at the end without changing the previous content	When there is no file in the given name it creates a file with the given name and writes the string to that file. When again function is called with same filename the each string when passed tot eh function call is appends to the file.	Passed	This function handle InputOutput exception. When there is an exception occurred it will print a message to the console

<table>
<tr><th></th><th></th><th></th><th></th><th></th><th></th><th></th></tr>
<tr><td>U8</td><td>Test the functionality of the validateReadFile(). This function try to open a file with given name . Then return the scanner object that wraps a file object if the file is found null otherwise</td><td>1. Create a dummy file in the expected path with dummy content.
2. Call the function with different filenames
3. Call the functions with correct filename
4. Check the return value is null or not</td><td>If the filename is valid the return value is not null. It should output the not null value. If the file is not in the path, it should print null</td><td>If the invalid file name is given it prints a error message and return the null. If the file is exits it print not null value</td><td>Passed</td><td>The printed not value is hard to read. Because it returns a Scanner object. It gives some not null string value</td></tr>
<tr><td>U9</td><td>Test the functionality of the splitString()</td><td>1. Call the function with a string and a delimiter as string
2. Hold the results to a String array
3. Print array using to string method
4. Repeat the process for different string and delimiters</td><td>When a string given that consists with the given delimiter result should be printed Strings[words] without the delimiter. If delimiter is not in the given string array should have only one string(Given string). If the string is empty array is empty</td><td>When the string with given delimiter is passed with the delimiter It prints a array , which consists of tokens that separated by the delimiter. If the delimiter is not in the string whole array have one element and that is given sttring</td><td>Passed</td><td></td></tr>
<tr><td>CU1</td><td>Test the functionality of the getDiscount(). This function get the type of the Contract and te int value as period and calculate the discount</td><td>1. Give different combination of types and period
2. Get the result and print it</td><td><table>
<tr><th>Type</th><th>Period</th><th>Expected</th></tr>
<tr><td>Busi.</td><td>12</td><td rowspan="3">10</td></tr>
<tr><td></td><td>18</td></tr>
<tr><td></td><td>24</td></tr>
<tr><td>Non Busi.</td><td>12</td><td rowspan="2">5</td></tr>
<tr><td></td><td>18</td></tr>
<tr><td></td><td>24</td><td>10</td></tr>
<tr><td>Charr.</td><td>-</td><td>30</td></tr>
</table></td><td>For business contracts if the period is greater that or equal to 12 it gives 10 as result.
For non-business type if the period is 12 or 18 it gives 5 and for 24 it gives 10. Charity customer it is given 30.</td><td>Passed</td><td></td></tr>
</table>

				If the invalid duo comes return 0		
CU2	Test the functionality of getMonthlyRate() function. It takes the package and the data bundle and returns the relevant package rate	1. Give every combinations of the package and the data bundle and save the result in each round 2. Print the value that saves to the variable 3. Manually check the value with relevant indexes to the array	If the data bundle is unlimited and package is small or medium it should return 0. If any other combination should return the Relevant value of the array. Value is referred by convert package and data bundle into indexes and return that value. If the invalid duo is given should print an error message	It gives 0 for unlimited data bundle with small or medium packages with and error message. If the relevant packages are correct (have an entry to the array) it gives that value	Passed	In here array is kept in the Class as static array. We can manually give the index and get the values to verify our answer is correct
CU3	Test the functionality of the calculateMonthlyCharge() method. This method will get the details and according to the detail calculate the total charge per month with discounts and taxes	1. Give every combination of the type, package, data bundle and international calls status 2. Print the results of each combination and check	When the parameters are given to the function it should returns the total charge. If the discount is valid, it should be added to the total amount (decrease discount from total). If the international calls are offered (true) it should be added 15% tax to the **curren**t total amount. Total charge should ignore the decimal places(ignore).	It gives correct total charge for every combination. It adds the tax if the international calls are true add 15% tax to the current amount. It returns only integer. Truncate the decimal places (does not round up or down)	Passed	
CU4	Test the functionality of the isContractValid(). It gets the package type and data bundle type and return true if the packages are compatible false otherwise	1. Give every combination of the package and the data bundle and print the results	For package which are small or medium with unlimited data bundle should print the false. Other valid combination should print the true.	When it is given Package = small or medium with data bundle is unlimited plan it prints false to the console. For other	Passed	

				combinations it prints true		
CU5	Test the functionality of the displayContract() , getDataBundle(),getPackage(), intToPackageT, intToDataT	1. Given the relevant details to the displayContract() function. All the inputs are valid. 2. Call the function with those values and check the result. 3. Change the names, packages, data bundle in different types and check the result and repeat	• When a combination of valid inputs is given) it should print the name of the client date reference number. When prints the package value it should prints the type of the package and the value of that also same for the data bundle. If the data bundle is unlimited, it should only print Unlimited in the Data field. • For every Business Contract it should prints Business in the type field, for non-business it should prints normal and charity for Charity contract • According to the given data if it does not have any discount the Discount field should be displayed as Discount: None. Also, if the discount is applied last line should be discounted monthly charge if not just monthly charge	• It prints the names field and date field and ref field correctly • According to the type it prints Business. Normal and charity in the type field. • It prints the Package and Data field correctly according to the description • Prints Discount field as None if no discount is offered and prints the discount value with % symbol if has discount • Prints the Yes in Intl. Calls field if the Boolean value Is true prints No is false	Passed	In here several functions are tested together. getDataBundle(), getPackage(), intToPackageT(), intToDataT() is checked parallel. These functions are used to get the Different field and check the functionality

			• In that monthly charge field, it should print in pounds with 2 decimal places. • If the international calls are true Intl. calls field should Yes otherwise No	• Displayed the monthly charge in 2 decimal places with £ sign. • If discount is applied add a prefix “Discounted” to the last field		
CU6	Test the functionality of the selectDataBundle() and selectPackage(). According to the given value (int) it selects the data bundle. Also, other function it returns selected package according to the data amount and current package	1. First call the selectDataBundle() function with different int values. 2. Then call the other function with different int values and different types of data bundles.	if the estimated data amount is exceed more than 1 GB with standard data bundle(1GB, 4GB, 8GB, Unlimited) then this method select next upper bound data bundle. And the data amount exceeds more than 8Gb package then set it to Unlimited • more than 11 999 MB = Unlimited • 5000 MB <= data <= 11 999MB -> HIGH • 2000 MB <= data <= 4999 MB -> MEDIUM • less than 1999MB = LOW	It prints the nearest upper bound data bundle if the threshold value is exceeded. Otherwise set to the lowest possible package. It gives large package for unlimited data bundle customers	Passed	
			For the selectPackage() Function if the data bundle is unlimited it should returns the Large plan. All inputs are			

			thresholding 200 to the upperbound			
CU7	Test the fucntinality of the saveContract() function. This will get a Contract and save the details to the file.	1. Create dummy objects of Contract 2. Call the function with these contracts 3. Check the file 4. Check the format with archive.txt file	If the file is not created initially a file should be created after the first call.(contracts.txt) Each call of the function contract details should append the line It should gives the integer values to packages (Low=1,..) and data bundle. Should be in the archive.txt format	It creates a file if file is not already in the directory(contracts.txt) For each call it only appends the line data of the contract to the file It saves the package and data bundle data as integers. It follows the archive.txt file format	Passed	
CU8	Test the functionality of the validateMonth()	1. Call the function with a prompt message and get the input for several time 2. Give different kind and length string to the input and check the result 3. Give input with mixed cases characters (“jAn”) 4. Save the result and prints	If the invalid length input is given (not equal exact 3) it should give an error message and again prompt and get the user input. If the Exact3 length characters given as invalid month (not any combination of “jan”,”feb”,…) given it should print different error message and again prompt to get the input If the valid input “Jan”,.. is given that value is printed to the console. If the mixed characters are given it also	Gives different error messages for invalid length and not valid month inputs and asked again to enter the month again. If it is given “Jan” or valid but mixed case characters it prints that mont correctly	Passed	

			prints thee month in lower case.			
CU9	Test the validateRefNumber() function in the ContractUtility class	1. Using appropriate prompt get the input from the user for the reference number 2. Check the printed results is correct 3. Give Different combination of errors 4. Finally enter a valid reference number	- Should print the error message when the input is higher than length 6. - Should print different error message when there is no 3 digits in the middle - Should print another different message if the input does not contain 2 leading letters and one tailing letter (B or N) - Print a reference number if everything is perfect - User should keep asking until enter a valid reference number	- Gave error message when length is not 6 - Gave different types of error when wrong combinations of reference number is entered - Gave the correct output when entered a valid reference number - Keep asking user until enter a valid reference number	Passed	In here this function used different utility function for check fixed length
CU10	Test validatePeriod function of the ContractUtility class. This method is used to get the input for the period from the user and validate it	1. Call the function and assign the return value to int variable 2. Add a print function after the assignment 3. Give dummy valid reference number to the function 4. Enter invalid periods	- Check the user input is within the range and if not give error message - If the user is Business and select 1-month contract, then given different error message - If user any user enters a invalid period except the 1,12,18 or 24 give error message - If everything is correct period should be print	- Gave error message for not in range inputs - Gave error message for Business customer with 1-month contract - Gave error message when enter invalid period	passed	This function only returns the period. In the testing purpose that return value was checked for confirmation

		5. Finally enter a valid period		• Kept asking until enter a valid period • Print the period when it is correct in the test code		
D1	Test the functionality of mainMenu() function	1. Call the method in a while loop and save the returned value from that function 2. If the value is 0 exit from the loop 3. Otherwise print that value into the console	Should Prints the message to the user before getting the input If the invalid range input [other than 0-4] is given it should give an error message. If the input is in given range should print that value to the console	It prints the menu to the console before gets inputs When this is given a value that is not in the range it gives an error message.	Passed	The while loop in the main method is just for the testing this function
D2	Test the functionality of the createContract() function	1. Create while loop for some iteration 2. Call this function inside the while loop 3. Give inputs and check the results 4. Repeats several times with different inputs	• When it gives invalid input for each field, invalid length, or in valid characters it should give error message and prompt until give a correct value for that filed. • User should be asked to choose to automate the package selection or not • If select Y then prompt the user to get the minas	• If the input for any field is wrong or packages are not compatible user gets an error message and asked to enter the relevant field again until entered correctly • User gets option to automate the package selection and if enter yes it asked the amount	Passed	

			and data amount and validate it. • If enter a wrong input again asked to enter the value with error message. • If the entered data amount and mins amount or data package and data bundle (manually selected) is not valid give an error message and prompt again to enter that fields • If the valid details are given it should print the contract in given format and save that details into the contracts.txt file	of the data and minutes • If the inputs are valid then prints the contract and save the data to the file		
D3	Test the displaySummary() and geSummary()[in contractUtility] Function.	1. Call the function 2. Give the input 3. Check the results 4. Try to call the function after data file is not in the directory	• When the option 1 is given it should read and print the data from the contracts.txt • When the option 2 is given it should read from the archive.txt • If the file is not, it should give an error message and exit	Validates the inputs from the user Give and appropriate error message for file not found or invalid input is given If choose correct option display the data according to the	passed	In here displaySummary() method get and validates the input option from the user. Then it calls the getSummary() method of the ContractUtility class with givn

			• Should print the summary in the given format • If one month doesn’t has any contract should print zero • When the input is given if the invalid option it should give an error message again prompt again to enter • Should print values correctly	correct format from the relevant files.		option. All the printing part handle by this function
D4	Test the functionality of getSelectedMonthSummary()	1. Given an input in out of range 2. Give inputs from the valid range and check the results	• If the user gives invalid option it should give an error message and prompt again • If correct option/month given and file is not exist should print error message • If file exist and option is correct and month is not valid prompt again with error message • If the month and the option is correct and file is exist should print the details of the relevant month according to the fromat	When file is exist and options are correct it prints correct data to the console If invalid input for month and option is given appropriate error is displayed. When there is invalid user input it prompts again If the file is not exist give error message and exit	Passed	

CU11	Test the functionality of the findContract()	1. Given a option and search term 2. Check with different combination 3. Check the results	According to the option should search from the correct file If file is not found give an error message and go to main menu If file is existed search against every item refNumber and name for partial or full match and if found should print n the given format If not found should print a message to the user	Read from the correct file if the option is 1 or 2 is given If file is not in there give an error message If the searchTerm is contained any of the refNumber or name that contract data displayed to the console	Passed	