**JUNIT Assignments**

1. Create a class WordCount with **int count(String str)** method which accepts string from the user and returns no of words in a given string. Then create a class with one public method using Junit unit test that test the count(String str) method. Test the method by using any one of the methods of the asserts as per the above scenario and display the output in the Junit framework using Eclipse environment and also check for the number of test cases passed and failed.
2. Create a class ArraySort with **int[] sortNumbers(int [] arr)** method which accepts unsorted array from the user and returns sorted array. Create ArraySortTest class with a public method to test sortNumbers(int [] arr) method. As per the scenario use any one of the assert methods and display the output in the Junit framework using Eclipse environment and also check for the number of test cases passed and failed.
3. Create a TestSuite to test below class methods.
4. AccountDetails class with six methods
5. acctNo ( int)
6. acctName ( string)
7. acctBalance ( float)
8. Deposit()
9. Withdraw()
10. creditCard() (auto generate the credit card number whenever a transactions is made)
11. Create a class DBConnection with **String** **connect(String drivername,String url,String username,String pwd)** method which returns string type of result success/failure. create a test class with drivername,url, uname,pwd variables & two methods setUpBeforeClass() to initialize drivername,url, uname & pwd variables and DBConnectionTest() to test the connect() method by using appropriate assert method which displays the output in the Junit framework using Eclipse environment.
12. Create a class Book with **double discountedPrice(int price,double discount)** method which accepts input from the user and returns total amount to be paid after discount . Test Book class discountedPrice(-,-) method by using Parameterized Test approach with @RunWith(Parameterized.class) annotation.