# TDD Assignments

## JUnit Introduction

**Basics**

Q1: Write a TDD based app to find out the addition of 10 consecutive numbers.
Q2: Write a TDD based app to find out the factorial of given number.
Q3: Write a TDD based app to find out the total even digits and total of odd digits in a long given number. Exm : 542587545 : 4 even digits and 5 odd digits.
Q4: Write a TDD based app to provide an expression and find out the sum of digits in that expression. Exm: “2+3+4” ans :  9
Q5: Write a TDD based app to provide an expression and find out the sum of digits in that expression. Exm: “2+3-4/2*5”
Q6: Write a TDD based app to find the sum of all the integers greater than 100 and less than 200 that are divisible by 7. Check for right and wrong input.
Q7: Accept a number and find its binary equant.

**Intermediate**

Q1: Create StringTest class, this will have multiple test cases like testStringLength(), testIndexOfTest() etc. which will be testing all the String methods. You need to write the test cases for below methods for StringDemo component.
charAt, concat, contains, endsWith, equals, equalsIgnoreCase, indexOf, intern, lastIndexOf, length, replace, split, substring, toLowerCase, toUpperCase,trim, valueOf.

Q2: Recreate the above program, but this time you need to write the logic for each method, you are not supposed to use any of the library method.

Q3: Write a TestClass to evaluate the action from given url.
Suppose you provide the following urls one by one.
www.yash.com/index.jsp
www.yash.com/employees/salary.xhtml
www.google.com/searches/searchdata.jsp

if above urls have been provided as input then in output we should get index, salary, searchdata as actions.
Q4: Create a program for a Car shop to calculate the no of cars sold. Car class has three attributes: speed (int), car owner (String), car type (String).Write methods to get and set above values of the car object. Also devise a way to calculate the number of cars sold by the car shop.

Q5: Create a program for a Hostel to calculate the no of students registered. Student class has three attributes: Student_ID (int), Student_Name (String), Student_class (String).Write methods to get and set above values of the Student object. Also devise a way to calculate the number of students registered in the hostel.

Q6: Create a program for an Author to calculate the number of books he/she has written. Book class has three attributes: Book_Name (String), Book_Price (Double), Book_Type (String). Write methods to get and set above values of the Student object. Also devise a way to calculate the number of books written by the author.

Q7: Create a program for a Company to calculate the number of employees working in that company. Employee class has three attributes: emp_Name (String), emp _Designation(String), emp _ID (Integer).Write methods to get and set above values of the Employee object. Also devise a way to calculate the number of employees working in the company.

**Advance**

Requirement
We have bug tracking system, in this we have a requirement to create application first on which different projects will be created, one Application may have multiple projects.

**Requirement #1**

Application class will have application id and application name.
Create an ApplicationTest class first , and from here we need to add application using addApplication method of ApplicationService class. addApplicationMethod will take application object.
User should be prompted accordingly that application has been created successfully with application detail.

**Requirement #2**
Application can be edited by the application owner only. Assume that you need to test the application editing feature. In the ApplicationService class create public Application updateApplication(int applicationId); method, that will be taking id value and new application name that has to be edited.

**Requirement #3**
Application can be deleted by the application owner only. Assume that you need to test the application deleting feature. In the ApplicationService class create public void deleteApplication(int applicationId); method, that will be taking id value and based on that application must be deleted.

**Requirement #4**
As number of applications can be created, assume that you have an array of size 50, where you can create applications. Show the list of available applications; create a suitable method to test. This method should return the array of Applications.

**Requirement #5**
Update the Application class, so that you can accommodate the creation date of the application as well. Test your application for changes.

**Requirement #6**
We want to get the list of applications based on date range. Test the feature, based on output create a test method and for the same we want you to create an appropriate method in the service class. which will return the array of applications based on date range.

**Requirement #7**
We want you to test a feature to get the list of applications based on free text search. Like, user can provide the application name or date, and it must return the application, this will include everything in the application.

## String Assignment

**Purpose: To demonstrate the creation of String object in different ways.**

Assignment 1:

Create a class Document

Create two fields in this as title and filepath

Create appropriate getter and setter methods

Create showDocumentInformation() method that should return one String value as ***"{title:java basics, filepath:c:/document/corejava/basics/introduction.pdf}"***

- attempt it with Test first approach.

Assignment 2:

- Create a class Member
- Create few fields in this as firstname, lastname, email, password, role
- Create appropriate getter and setter methods
- Create showMemberDetail() method that should return one String value as ***"{name : Pankaj Sharma, email : sharma.pankaj@yash.com, password : 12345, role : Trainer}"***
- attempt it with Test first approach.

Assignment 3:

- Create a class Section
- Create few fields in this as id, name, createdDate
- Create appropriate getter and setter methods
- Create showSectionDetail() method that should return one String value as ***"{id:101,name: core_java, createdDate : 31-01-2017}"***
- attempt it with Test first approach.

Assignment 4:

- Create a class Category
- Create few fields in this as id, name, createdDate
- Create appropriate getter and setter methods

- Create showCategoryDetail() method that should return one String value as ***"{id:101,name: javabasics, createdDate : 31-01-2017}"***

- attempt it with Test first approach.

Assignments for Practice:

Purpose: to demonstrate the String creation methods ie. (using direct initialization in String reference, using new operator, using character array, by taking the part of string from given string.)

**Assignment 1:**

- Create a class StringDemo

- Create a Test class as StringDemoTest

- Create a field as input in StringDemo

- Create appropriate getter and constructor method

- Create getRequiredData(String input) method that should return the provided String unchanged

- For example you have input as "Java Test" then it should return Java Test string as it is, if nothing is provided then "there is no String" should be returned.

**Assignment 2:**

- Create a class Employee

- Create fields like firstName, lastName, company and role

- Create a method, generatePassword(), it should return the generated password

- Create a Test class as EmployeeTest

- If all entries are properly provided, password should be generated as (it should take first two characters from each type, like firstname,lastname,company and role :  example: firstName-pankaj, lastName-sharma, company-yash, role-trainer, then password should be as **pashyatr**)