

## **Core Java Assignments**

### **Lab 2: String Assignment**

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

#### **Exercise 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.

#### **Exercise 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.

#### **Exercise 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.

#### **Exercise 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.

#### **Exercise 5:**

- 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.

#### **Exercise 6:**

- 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)

#### **Exercise 7:**

- Create a NumberToString Class
- Create one double field
- Initialize some data using setting
- Create one numberOutputInString() method and it should return the String output.
- ·For example if you provide input as " 25.4 it should display the output as 2 digits before decimal"

#### **Exercise 8:**

- Finding the repeated character along with its count in a word.
- Create a class DuplicateCharacter
- Create a field as word
- Create getter and setter method
- Create findDuplicateCharacter() method that should return the duplicate character along with count in provided word.