AUTOMATION TESTING EXERCISES

Core Java Concepts

1.) Implement the following logic to arrive at the loan eligibility for an employee:

(a) For an unmarried permanent job holder, if the service has been for 30 years or more,the loan amount is Rs.500,000; otherwise the loan amount is Rs.250,000
(b) For a married permanent job holder, if the service has been for 30 years or more, the loan amount is Rs.600,000; otherwise the loan amount is Rs.350,000
(c) For temporary job holder, the loan amount is Rs.100,000.

Check the loan eligibility with the following input conditions.
(a) a married temporary job holder
(b) an unmarried temporary job holder
(c) a temporary job holder with 30 years of service
(d) a married permanent job holder with 30 years of service
(e) an unmarried permanent job holder with 30 years service
(f) a married permanent job holder with 25 years service.

2.) Prepare the logic for the below series

(a) 1, -3, 5, -7....
(b) 1/1, 1/2, 1/3, 1/4....

3.) Consider an array and perform the below operations

(a) Count the number of odd numbers divisible by 3 in even positions.
(b) Search for a key value in array and print it is available in the array or not.
(c) Count the number of prime numbers.

4.) Write a function validateIPaddress that accepts an ipAddress (string) as an input and does the following validations(Strings & functions)

(a) Length should be minimum 7 and maximum 15
(b) Should not start and end with dot
(c) Should have exactly 3 dots
(d) Dots should not be in consecutive positions.
(e) Should not permit alphabets and special characters

OOPS

1.) Design a class HTCGlobalServices which has the following attributes:
Company ID
Address
EmployeeMap < id, Employee> (Employee is an entity class, key of the map is employee ID and value is Employee object)
HTCglobalservices class should implement ICompanyserviceprovider to facilitate basic CRUD operations on Employee Class.

Prepare three user defined exception class
a) “EmployeeNotAvailableexception”, use it in Read, Update and Delete operations.
b) “DuplicateEmployeeException”, use it in Create operation.
c) “InvalidSalaryException”, use it in create operation, if the salary of an employee is negative.

“HTCLOG.txt”
The Employee entity class should have
a) private attributes
b) public setters,getters,constructors
c) Should override toString(),hashCode(),equals()
d) Should implement serializable marker interface

2. Design a class HTCEmployeeServices which implements IDBservices to facilitate basic CRUD operation on an employee table
Prepare three user defined exception class
a) “EmployeeNotAvailableexception”, use it in Read, Update and Delete operations.
b) “DuplicateEmployeeException”, use it in Create operation.
c) “InvalidSalaryException”, use it in create operation, if the salary of an employee is negative.

“HTCLOG.txt”
The Employee entity class should have
a) private attributes
b) public setters,getters,constructors
c) Should override toString(),hashCode(),equals()
d) Should implement serializable marker interface
(Do not hard code the connection string in the program)
(Update operation should update salary of a give employee)

Connect to a database and implement the basic crud operations against a database . Hardcode the configuration details in a config.xml file. Use stored procedure in place of queries.

UNIT TESTING FRAMEWORK

TestNG

1. Create test suite for testing an E – Shopping site, unit test the website using relevant testcases incorparating testng annotations. Create Testclasses for the below scenarios :
   - Creating a user account
   - Adding products to the cart
   - Adding to the wishlist
   - Checking ordered lists
   - Use different testng annotations and generate a report.

2. Use the above exercise and group the test cases based on regression,sanity,smoke and likewise. Invoke the groups from xml file – (use include and exclude parameters) ,run configuration.

   Create a dependency between relevant testcases. Try to use timeout,dependency method and always run parameters.

3. Apply data driven testing for the above scenario, please hardcode the datasets . Using the concept of parameterization a tester must be able to select the browser of his/her choice from the configuration file.

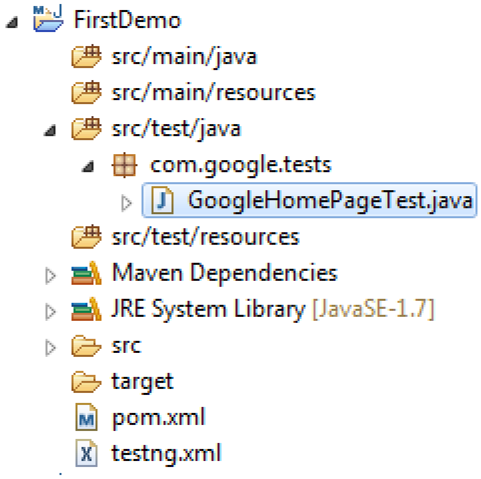
JUNIT

4. Create a JUnit test class,Bank `AccountTest`, and write test cases which exercise all the methods of the BankAccount class.
Compile your classes and and run the JUnit tests. Correct any defects in your tests.
If your tests reveal any defects in the logic of the BankAccount class, repair them.
Write a new test case that reveals a more subtle defect lurking in the Account class. (Hint: Negative balance).

MAVEN

1. In this exercise you create a Java project with the Maven command line and build this project.

    - Create a maven project from command prompt with the required artifact type .
    - Validate that Maven generated a project on your file system similar to the following structure.



```
FirstDemo
    src/main/java
    src/main/resources
    src/test/java
        com.google.tests
            GoogleHomePageTest.java
    src/test/resources
    Maven Dependencies
    JRE System Library [JavaSE-1.7]
    src
    target
    pom.xml
    testng.xml
```

    - Add the relevand plugins and dependencies needed for creating a project in POM.xml file. Try implementing different lifecycle commands . Use the below snippet as a guide

```
<project>
    <modelVersion>4.0.0</modelVersion>
    <groupId>com.vogella.build.maven.java</groupId>
    <artifactId>com.htc.maven.java</artifactId>
    <packaging>jar</packaging>
    <version>1.0-SNAPSHOT</version>
    <name>com.vogella.build.maven.java</name>
    <url>http://maven.apache.org</url>
    <dependencies>
        <dependency>
            <groupId>junit</groupId>
            <artifactId>junit</artifactId>
            <version>3.8.1</version>
            <scope>test</scope>
        </dependency>
    </dependencies>
</project>
```

    - Try to compile, package ,test and build the project with various commands from the command Prompt.

2. Do the above steps in Maven interactive mode(through eclipse)

SELENIUM

1.Create a simple program to visit a website from different browsers.

2. Parameterise the browser type using TestNG parameterisation concepts.

SELENIUM WEBDRIVER

(WebLocating Techniques,Multibrowser,Testng,Textbox & buttons)

3. Automate a test class to check the user credentials for Eportal . Use a combination of locator techniques to locate the elements in the webpage.The test must execute both on Google Chrome & Firefox. Using the capabilites class create a profile in firefox with specific attributes.

(Calendar control,Drop Down box,rich textbox, customise xpath,Parent /child elements)

4. Automate a test to fill in the Employee – edit profile in eportal. In order to locate the elements , use FindElementBy tag and store the values in the parentElement , all the child elements could be retrieved using xpath.

(Customized xpath, Synchronisation,Navigation)

5. Create a automated test to automate reimbursement submission in eportal.

(***DatePicker,Drop down,Customised xpath)***

6. Automate a test to log into eportal and move to ars. Using date picker , choose the month january and check if the the value for 26/01 is 'H' under holidays.

(***JavaScriptExecutor)***

7. Automate a test to open Blossoms page , using java script executor try to choose the article of

(***Alert,Frames,PopUP)***

8. Use the existing sample application in SVN , try to handle frames and popups for the given website.

Data Driven Testing & Report Generation

9. Using data driven testing automate the testing of 10 user id s in eportal , to fill up the Leave form of EPortal

10.Create a webpage which can register new users , after the registration has been completed successfullly the user must be able to log in. Test these functionalities using Selenium.

Generate a report for the above tests using Testng listeners.

11. Create a data driven testing to fill up the contents of a webpage by reading data from the excel sheet , once the user clicks the submit button validates against the database to make sure the data added are not duplicate . Once the new user registration is completed successfully , the user must be able to sign in with his credentials and choose products of his choice. The product summary must be downloaded to the local disk.

Generate a report for the above process using extent test report.

GUI AUTOMATION

12. Use Gui automation techniques(robot,winnium,auto it) to upload a resume to a sample web application.

Page Object Model

13. Create Page object Model for eportal website . Do the below given tasks.

- create a properties file from which the automation script would be reading the element configurations.
- All the database connectivity and loggingdetails must be logged in configuration file.
- Read the test data from excel sheet
- Store the user details into the database after validations
- Automate the signing into eportal by reading data from the database .
- Create an excel report of the same.

Your framework must be designed in such a way that its highly configuratble for any website.

JENKINS

1. Build the Continuous Integration pipeline for the the Page object Model project that you have created:

Use Jenkinsfile for specifying the pipeline

Configure the trigger so that the pipeline runs automatically in case of any commit to the repository

Run the pipeline and observe the results

Try to commit the code that breaks each stage of the pipeline and observe how it is visualized in Jenkins