

# Veeva Integration Program - Assessment 3

Subject: Selenium automation testing

Assessment type: Practical test

## Question1: Write a script to login next generation automation.

Consider a website or web application and our task is to write a java script for a login scenario, the script should demonstrate a generic login process. This can be adapted by replacing the URL and element locators like username, password. with those specific to next generation automation.

## Question2: Write a script to create browser instance based on browser name

## Question3: Search the filter list box, write a script to search for a specified option in the list box.

The script also implement filter the list by creating a search bar the hides non matching items.

## Question4: Create and execute a script to open google.com

Verify the title is google and also verify it is redirected to google.co.in

## Part 2 – Java Programming

Assessment type: Practical test

Question1: Consider storing information about different countries and their capitals. In our problem we deal with different types of countries where some countries have only one capital and there some countries which have two capitals. In such a case the capital can be winter capital or summer capital. Summer capital means capital for both summer and rainy seasons. There some countries for each season winter, summer and rainy. Now store information about N different countries and their capitals and then execute following queries based on request.

1. Print all countries, number of capitals in each country and corresponding capitals along with season also
2. Given a country name and print capitals of that country along with season and number of capitals for that country
3. Print all the countries having two capitals
4. Print all the countries having three capitals
5. Find the countries that does not have summer capital
6. Find and return capitals of the countries with starts with Vowel
7. Print summer capitals of the countries that starts with vowel

Question2: Several students in a school are waiting to be serve. There two types of events that can take place. One ENTER, two SERVED. A student with some priority enters the queue to be priority to. A student with highest priority served (removed from the queue). We assign unique ID to each student enter into the queue. The queue serves the student based on the following priority criteria

1. Student having highest CGPA is served first
2. Any students having same CGPA will be served by name in ascending case sensitive alphabetical order
3. Any students having equal CGPA and name will be served in ascending order of their unique ID.

Now the task to process all the even events and return all the students yet be served in the priority order.

Example1: Input: N = 12

ENTER john 3.75 50

ENTER nort 3.8 24

ENTER shafi 3.7 35

SERVED

SERVED

ENTER shamiha 3.86 36

SERVED

ENTER Ashley 3.9 42

ENTER Meria 3.6 46

ENTER anik 3.95 49

ENTER dan 3.95 50

SERVED

Output:

dan

Ashley

shafi

meria

## **Linux Shell Scripting**

Subject: Linux OS

Assessment type: Practical test

Question1: Linux script to search files

a. Write a script to check if a file exists in a system or not

Check if a file exists in a current directory or not

Check if a file exists at any part of the system or not

b. Write a linux script that searches for files modified within the last three days within the current directory

c. Write a linux script that searches for files larger than 1KB in the current directory

d. Write a linux script that searches for files with a specific extension in the current directory

Question2: Text processing and searching for patterns

a. Write a linux script that searches for specific string in a given text file

b. Consider an html file as input and write a linux script to extract text between specified html tags

Question3: logs

a. Write a script that logs message to a file log.txt

b. Write a script that logs errors to a separate file error.log

Subject: SDLC, Agile Methodology and SCRUM

Assessment type: Viva Voice

Q1: What is the agile manifesto. How it controls the product development

Ans: Agile manifesto is a set of values and principles for software development that emphasizes the team collaboration, flexibility and delivering value to the customers. It guides through principles and practices that create a flexible and iterative process to develop a software product.

Q2: Explain the difference between burn-down and burn-up charts

Ans: A burn-down chart and a burn-up chart are both Agile tools for tracking progress, but they show it differently. A burn-down chart tracks the remaining work over time, starting high and declining toward zero as tasks are completed, making it easy to see if a team is on track to finish a sprint or project. While a burn-up chart tracks completed work, starting at zero and rising as tasks are finished, often including a second line for total scope to show any changes in project requirements. While burn-down charts are useful for monitoring short-term progress within a sprint, burn-up charts are better for visualizing overall project progress and scope changes, making it easier to understand how much work has been done versus how much remains.

Q3: How do you handle changing requirements in agile projects

Ans: Agile methodology is designed to be flexible and adaptive, which allows teams to respond quickly without derailing the project. Agile handles changing requirements by embracing them through collaboration, flexible planning, iterative delivery, and constant feedback, turning what is often a risk in traditional methods into an advantage for delivering value more effectively.

Q4: How do you handle a situation where testing is not completed within sprint

Ans: When testing is not completed within a sprint, the unfinished stories should be moved back to the product backlog, not automatically carried over. The Product Owner (PO) must re-prioritize these items, and the team should not claim velocity points for them. Finally, discuss the root cause during the retrospective to improve future flow.

Q5: How do you estimate your tickets

Ans: Estimating tickets, particularly in software development, involves analyzing the complexity, effort, and uncertainty of tasks rather than just time. Common methods include breaking down tickets into smaller subtasks, using relative sizing like story points (Fibonacci sequence) or t-shirt sizes, and applying three-point estimation (optimistic, pessimistic, and most likely).

Q6: What you do when your automated suite is fail

Ans: When an automated test suite fails, immediately re-run the tests to rule out transient network or environmental issues. If failures persist, analyze logs and reports to distinguish between actual application bugs and faulty, and cross check the scripts (e.g., locator changes, timing issues).