

Java Selenium Interview Programs Part 1 (2026) | Saravana Tech

55 questions • explained solutions • practical examples

About this guide

Examples use Java 8+ and Selenium 4 unless noted. Selenium snippets assume imports and WebDriver setup are
Use URLs, selectors, file paths, and settings configured for your test environment.

Setup notes

- Add selenium-java for Selenium examples; Selenium Manager can resolve drivers in supported environments.
- Add Apache POI (poi and poi-ooxml) for Excel examples and PDFBox for PDF extraction.
- Use the JDBC driver for your database. Keep credentials in environment variables, not source code.
- Run each standalone snippet in a suitable class or adapt it to your existing test framework.

How to run these programs

> Every program is now **complete**: it has its `import` statements, a `public class Main` and a `main` method (Q13 is a TestNG class named `LoginTest`). Paste it into any Java 11+ compiler and run.
>
> **Green badge**: uses only the JDK and runs in any online compiler (Q1–Q4, Q8–Q10, Q14–Q16, Q18–Q19, Q22, Q24–Q29, Q31–Q38, Q40, Q45, Q52–Q54).
>
> **Amber badge**: needs an external library (Selenium with Chrome, Apache POI, PDFBox, a JDBC driver or TestNG). Online compilers cannot run these. Run them in an IDE such as IntelliJ or Eclipse with the Maven dependencies from the Setup notes.

Contents

1 Find duplicate and non-duplicate integers 4
2 Remove duplicate characters from a string 4
3 Check whether a number is prime 5
4 Palindrome string/integer; HashMap/Hashtable; thread 5
5 Take a screenshot using Selenium 6
6 Read Excel values 6
7 Login to a site in Chrome and enter text 7
8 Salary if/else, sum String numbers, character occurrence 8
9 Fibonacci sequence 8
10 Java star patterns 9
11 Window handles 9
12 Right-click using Selenium 10
13 TestNG DataProvider 10
14 Reverse a string 11
15 HashMap definition and program 11
16 Abstract method 12
17 Initialize Chrome browser 12
18 Validate pattern [2312] [932E] [12AB] 13
19 Check whether two strings are anagrams 13
20 Verify a widget loaded correctly 14
21 Verify web-table columns A+B=C 15
22 Count digits in a string 15
23 SendKeys in Selenium WebDriver 16
24 Largest sum from a contiguous subarray 16
25 Count positive or negative values and prepare test data 17
26 Swap lowercase and uppercase 17
27 Find maximum number in an array 18
28 User-defined exception 18
29 Count duplicate string values using HashMap 19
30 Read text from PDF 19
31 Print a pyramid 19
32 Sort a string without built-in sorting 20
33 Singleton with double-checked locking 20
34 Recursive binary search 21
35 Singly linked list: add node and print in reverse 22
36 Serialization and deserialization 23
37 Even/odd threads using wait() and notifyAll() 24
38 Email validation 25
39 Copy contents from one Excel sheet to another 26
40 Separate letters, digits, and special characters 27
41 Switch to the last window 27
42 Connect to a database with JDBC 28
43 Check whether an element is visible 29
44 Count broken links on a page 30
45 Run same code twice and finish within 3 seconds 31
46 Run a package but exclude two TestNG methods 31
47 Interact inside a frame, then click outside it 32
48 Scroll a web page 32
49 Handle alerts 33
50 Check whether a page is loaded 34
51 Upload an image file 35
52 Find the Nth largest number 36
53 Sum the first 1000 prime numbers 36
54 Print all permutations of a string 37
55 Generate a unique email address each run 38

1 Find duplicate and non-duplicate integers

Runs in any online compiler

EXPLANATION Count each number. Values with count > 1 are duplicates; values with count = 1 occur only once. The output contains distinct values in original order.

Java

```java
import java.util.ArrayList;
import java.util.LinkedHashMap;
import java.util.List;
import java.util.Map;

public class Main {
    public static void main(String[] args) throws Exception {
        int[] a = {1, 2, 2, 3, 4, 4, 5, 5, 6, 7};
        Map<Integer, Integer> m = new LinkedHashMap<>();
        for (int n : a)
            m.put(n, m.getOrDefault(n, 0) + 1);
        List<Integer> dup = new ArrayList<>(), unique = new ArrayList<>();
        for (Map.Entry<Integer, Integer> e : m.entrySet())
            if (e.getValue() > 1)
                dup.add(e.getKey());
            else
                unique.add(e.getKey());
        System.out.println(dup);  // [2, 4, 5]
        System.out.println(unique);  // [1, 3, 6, 7]
    }
}
```

2 Remove duplicate characters from a string

Runs in any online compiler

EXPLANATION LinkedHashSet keeps the first occurrence and preserves character order.

Java

```java
import java.util.LinkedHashSet;
import java.util.Set;

public class Main {
    public static void main(String[] args) throws Exception {
        String s = "programming";
        Set<Character> seen = new LinkedHashSet<>();
        for (char c : s.toCharArray())
            seen.add(c);
        StringBuilder out = new StringBuilder();
        for (char c : seen)
            out.append(c);
        System.out.println(out);  // progamin
    }
}
```

3 Check whether a number is prime

Runs in any online compiler

EXPLANATION A prime is greater than 1 and has no divisor except 1 and itself. Check divisors only up to the square root.

Java

```java
public class Main {
    static boolean isPrime(int n) {
        if (n < 2)
            return false;
        for (int d = 2; d <= n / d; d++)
            if (n % d == 0)
                return false;
        return true;
    }

    public static void main(String[] args) throws Exception {
        System.out.println(isPrime(29));  // true
    }
}
```

4 Palindrome string/integer; HashMap/Hashtable; thread

Runs in any online compiler

EXPLANATION Normalize a string before reversing. For integers, reverse digits. HashMap is unsynchronized and allows a null key; Hashtable is synchronized and rejects null keys/values. Runnable is a common way to create a thread.

Java

```java
import java.util.HashMap;
import java.util.Hashtable;
import java.util.Map;

public class Main {
    public static void main(String[] args) throws Exception {
        String s = "Never odd or even".replaceAll("[^A-Za-z0-9]", "").toLowerCase();
        System.out.println(s.equals(new StringBuilder(s).reverse().toString()));
        int n = 12321, original = n, rev = 0;
        while (n > 0) {
            rev = rev * 10 + n % 10;
            n /= 10;
        }
        System.out.println(original == rev);
        Map<String, Integer> hm = new HashMap<>();
        hm.put("QA", 1);
        Map<String, Integer> ht = new Hashtable<>();
        ht.put("Java", 2);
        Thread t = new Thread(() -> System.out.println("Running"));
        t.start();
    }
}
```

> NOTE *For robust integer palindrome code, also handle negative numbers and overflow. Prefer ConcurrentHashMap for many modern concurrent use cases.*

5 Take a screenshot using Selenium

Needs Selenium + Chrome

EXPLANATION Cast WebDriver to TakesScreenshot and write the returned bytes to a file.

Java

```java
import java.nio.file.Files;
import java.nio.file.Paths;
import org.openqa.selenium.OutputType;
import org.openqa.selenium.TakesScreenshot;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            byte[] bytes = ((TakesScreenshot) driver).getScreenshotAs(OutputType.BYTES);
            Files.write(Paths.get("target/screenshot.png"), bytes);
        } finally {
            driver.quit();
        }
    }
}
```

> NOTE *Ensure target/ exists and import java.nio.file.*.*

6 Read Excel values

Needs Apache POI

EXPLANATION Use Apache POI WorkbookFactory and DataFormatter to read cell values in a display-friendly format.

Java

```java
import java.io.FileInputStream;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.DataFormatter;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Main {
    public static void main(String[] args) throws Exception {
        try (FileInputStream in = new FileInputStream("testdata.xlsx");
                Workbook wb = WorkbookFactory.create(in)) {
            Sheet sh = wb.getSheet("Sheet1");
            DataFormatter f = new DataFormatter();
            for (Row row : sh) {
                for (Cell c : row)
                    System.out.print(f.formatCellValue(c) + "\t");
                System.out.println();
            }
        }
    }
}
```

> NOTE *Add Apache POI dependencies. Use a FormulaEvaluator if formula results must be calculated.*

7 Login to a site in Chrome and enter text

Needs Selenium + Chrome

EXPLANATION Open the login page, locate controls with stable selectors, type credentials, click submit, and wait for a post-login element. Do not hard-code credentials.

Java

```java
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com/login");
            driver.findElement(By.id("username")).sendKeys(System.getenv("TEST_USER"));
            driver.findElement(By.id("password")).sendKeys(System.getenv("TEST_PASSWORD"));
            driver.findElement(By.cssSelector("button[type='submit']")).click();
            new WebDriverWait(driver, Duration.ofSeconds(10))
                    .until(ExpectedConditions.visibilityOfElementLocated(By.id("dashboard")));
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *Imports: org.openqa.selenium.*, org.openqa.selenium.support.ui.*, java.time.Duration.*

8 Salary if/else, sum String numbers, character occurrence

Runs in any online compiler

EXPLANATION Use if/else for defined salary rules, parse numeric strings with Integer.parseInt, and count characters in a map. The salary rates below are illustrative only.

Java

```java
import java.util.LinkedHashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) throws Exception {
        double salary = 900000, tax;
        if (salary <= 300000)
            tax = 0;
        else if (salary <= 600000)
            tax = (salary - 300000) * 0.05;
        else if (salary <= 900000)
            tax = 15000 + (salary - 600000) * 0.10;
        else
            tax = 45000 + (salary - 900000) * 0.15;
        String[] nums = {"10", "20", "30"};
        int sum = 0;
        for (String v : nums)
            sum += Integer.parseInt(v);
        String s = "automation";
        Map<Character, Integer> freq = new LinkedHashMap<>();
        for (char c : s.toCharArray())
            freq.put(c, freq.getOrDefault(c, 0) + 1);
        System.out.println(tax + " " + sum + " " + freq);
    }
}
```

NOTE *Do not treat the sample salary thresholds/rates as current tax advice.*

9 Fibonacci sequence

Runs in any online compiler

EXPLANATION Each term is the sum of the prior two; this iterative solution is O(n).

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        int a = 0, b = 1;
        for (int i = 0; i < 10; i++) {
            System.out.print(a + " ");
            int next = a + b;
            a = b;
            b = next;
        }
    }
}
```

10 Java star patterns

Runs in any online compiler

EXPLANATION Nested loops control rows and columns; this prints a left-aligned triangle.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        for (int i = 1; i <= 5; i++) {
            for (int j = 1; j <= i; j++)
                System.out.print("* ");
            System.out.println();
        }
    }
}
```

11 Window handles

Needs Selenium + Chrome

EXPLANATION Switch to a handle after collecting all handles. To identify a newly opened child window reliably, compare handles before and after the click.

Java

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            String last = null;
            for (String h : driver.getWindowHandles())
                last = h;
            if (last != null)
                driver.switchTo().window(last);
        } finally {
            driver.quit();
        }
    }
}
```

12 Right-click using Selenium

Needs Selenium + Chrome

EXPLANATION Use Actions.contextClick. Native browser context menus are not consistently exposed to WebDriver; assert the application's custom menu if there is one.

Java

```java
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.interactions.Actions;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            WebElement target = driver.findElement(By.id("rightClickTarget"));
            new Actions(driver).contextClick(target).perform();
            new WebDriverWait(driver, Duration.ofSeconds(5))
                    .until(ExpectedConditions.visibilityOfElementLocated(By.id("customContextMenu")));
        } finally {
            driver.quit();
        }
    }
}
```

13 TestNG DataProvider

Needs TestNG

EXPLANATION A DataProvider supplies multiple sets of arguments to one test method.

Java

```java
import org.testng.Assert;
import org.testng.annotations.DataProvider;
import org.testng.annotations.Test;

public class LoginTest {

    @DataProvider(name = "users")
    public Object[][] users() {
        return new Object[][] {{"user1", "pass1"}, {"user2", "pass2"}};
    }

    @Test(dataProvider = "users")
    public void loginTest(String user, String password) {
        Assert.assertNotNull(user);
    }
}
```

NOTE *Add TestNG imports and dependency.*

14 Reverse a string

Runs in any online compiler

EXPLANATION StringBuilder has a reverse method; a backward loop is an alternative for interviews.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        String s = "Selenium";
        System.out.println(new StringBuilder(s).reverse());
        StringBuilder r = new StringBuilder();
        for (int i = s.length() - 1; i >= 0; i--)
            r.append(s.charAt(i));
        System.out.println(r);
    }
}
```

15 HashMap definition and program

Runs in any online compiler

EXPLANATION HashMap stores key-value pairs, does not guarantee iteration order, and permits one null key. A repeated key replaces its value.

Java

```java
import java.util.HashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) throws Exception {
        Map<String, Integer> scores = new HashMap<>();
        scores.put("Java", 90);
        scores.put("Selenium", 95);
        scores.put("Java", 92);
        System.out.println(scores.get("Java"));  // 92
    }
}
```

> NOTE *Keys should have stable equals() and hashCode() implementations.*

16 Abstract method

Runs in any online compiler

EXPLANATION An abstract method has no body. A concrete subclass must implement it unless that subclass is also abstract.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        Report report = new PdfReport();
        report.header();
        report.generate();
    }
}

abstract class Report {
    abstract void generate();
    void header() {
        System.out.println("Report");
    }
}

class PdfReport extends Report {
    @Override
    void generate() {
        System.out.println("Generating PDF");
    }
}
```

17 Initialize Chrome browser

Needs Selenium + Chrome

EXPLANATION Instantiate ChromeDriver in Selenium 4, navigate to the URL, then close the driver in a finally block or test teardown.

Java

```java
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://example.com");
        // test steps
        driver.quit();
    }
}
```

18 Validate pattern [2312] [932E] [12AB]

Runs in any online compiler

EXPLANATION Interpreting the sample as three groups of four alphanumeric characters separated by spaces, validate the format only.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        String value = "2312 932E 12AB";
        boolean valid = value.matches("[A-Za-z0-9]{4} [A-Za-z0-9]{4} [A-Za-z0-9]{4}");
        System.out.println(valid);
    }
}
```

> NOTE *This does not validate a real payment card. Clarify whether square brackets are literal in the actual input.*

19 Check whether two strings are anagrams

Runs in any online compiler

EXPLANATION Remove spaces, normalize case, sort both character arrays, and compare.

Java

```java
import java.util.Arrays;

public class Main {
    static boolean anagram(String a, String b) {
        char[] x = a.replaceAll("\\s+", "").toLowerCase().toCharArray();
        char[] y = b.replaceAll("\\s+", "").toLowerCase().toCharArray();
        Arrays.sort(x);
        Arrays.sort(y);
        return Arrays.equals(x, y);
    }

    public static void main(String[] args) throws Exception {
        System.out.println(anagram("Listen", "Silent"));
    }
}
```

> NOTE *Define how punctuation and Unicode characters should be treated.*

20 Verify a widget loaded correctly

Needs Selenium + Chrome

EXPLANATION Wait for visibility, then assert meaningful content or a ready-state value. DOM presence alone does not prove that data loaded.

Java

```java
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            WebElement widget = new WebDriverWait(driver, Duration.ofSeconds(10))
                    .until(ExpectedConditions.visibilityOfElementLocated(By.id("weatherWidget")));
            if (widget.getText().trim().isEmpty())
                throw new AssertionError("Widget has no content");
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *For asynchronous widgets, wait for the spinner to disappear and assert expected data.*

21 Verify web-table columns A+B=C

Needs Selenium + Chrome

EXPLANATION Read the cells from each row, parse their values, and compare the sum with the third column.

Java

```java
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            for (WebElement row : driver.findElements(By.cssSelector("table#results tbody tr"))) {
                List<WebElement> c = row.findElements(By.cssSelector("td"));
                if (c.size() < 3)
                    continue;
                double a = Double.parseDouble(c.get(0).getText().trim());
                double b = Double.parseDouble(c.get(1).getText().trim());
                double sum = Double.parseDouble(c.get(2).getText().trim());
                if (Math.abs(a + b - sum) > 0.000001)
                    throw new AssertionError("Sum mismatch: " + row.getText());
            }
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *Adjust selectors and numeric parsing for currency/commas if present.*

22 Count digits in a string

Runs in any online compiler

EXPLANATION Stream characters and count those recognized as digits.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        String s = "QA2026-Automation55";
        long count = s.chars().filter(Character::isDigit).count();
        System.out.println(count);  // 6
    }
}
```

23 SendKeys in Selenium WebDriver

Needs Selenium + Chrome

EXPLANATION Use WebElement.sendKeys; clear an input first if the old value should be replaced.

Java

```java
import org.openqa.selenium.By;
import org.openqa.selenium.Keys;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            WebElement search = driver.findElement(By.id("searchBox"));
            search.clear();
            search.sendKeys("Selenium WebDriver");
            search.sendKeys(Keys.ENTER);
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *Selenium IDE's recorded type command is not a Java API method.*

24 Largest sum from a contiguous subarray

Runs in any online compiler

EXPLANATION Kadane's algorithm tracks the best sum ending at the current index and the best overall. O(n) time.

Java

```java
public class Main {
    static int maxSum(int[] a) {
        if (a.length == 0)
            throw new IllegalArgumentException("empty array");
        int cur = a[0], best = a[0];
        for (int i = 1; i < a.length; i++) {
            cur = Math.max(a[i], cur + a[i]);
            best = Math.max(best, cur);
        }
        return best;
    }

    public static void main(String[] args) throws Exception {
        System.out.println(maxSum(new int[] {-2, 1, -3, 4, -1, 2, 1, -5, 4}));
    }
}
```

NOTE *Works even if every number is negative.*

25 Count positive or negative values and prepare test data

Runs in any online compiler

EXPLANATION Count positives, negatives, and zeroes separately; test empty, all-positive, all-negative, zero-only, and mixed arrays.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        int[] a = {-3, 0, 7, -1, 5, 0, -8};
        int pos = 0, neg = 0, zero = 0;
        for (int n : a) {
            if (n > 0)
                pos++;
            else if (n < 0)
                neg++;
            else
                zero++;
        }
        System.out.printf("positive=%d negative=%d zero=%d%n", pos, neg, zero);
        System.out.println(pos > neg ? "More positive" : neg > pos ? "More negative" : "Equal");
    }
}
```

26 Swap lowercase and uppercase

Runs in any online compiler

EXPLANATION Convert lowercase characters to uppercase and uppercase to lowercase; leave digits and punctuation unchanged.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        String s = "Qa-Automation 123";
        StringBuilder out = new StringBuilder();
        for (char c : s.toCharArray()) {
            if (Character.isLowerCase(c))
                out.append(Character.toUpperCase(c));
            else if (Character.isUpperCase(c))
                out.append(Character.toLowerCase(c));
            else
                out.append(c);
        }
        System.out.println(out);
    }
}
```

27 Find maximum number in an array

Runs in any online compiler

EXPLANATION Initialize maximum to the first item and compare each remaining item.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        int[] a = {12, 45, 7, 89, 34};
        if (a.length == 0)
            throw new IllegalArgumentException("Empty array");
        int max = a[0];
        for (int n : a)
            if (n > max)
                max = n;
        System.out.println(max);  // 89
    }
}
```

28 User-defined exception

Runs in any online compiler

EXPLANATION Extend Exception for a checked exception and throw it when a rule is violated.

Java

```java
public class Main {
    static void register(int age) throws InvalidAgeException {
        if (age < 18)
            throw new InvalidAgeException("Age must be at least 18");
    }

    public static void main(String[] args) throws Exception {
        try {
            register(16);
        } catch (InvalidAgeException e) {
            System.out.println(e.getMessage());
        }
    }
}

class InvalidAgeException extends Exception {
    InvalidAgeException(String message) {
        super(message);
    }
}
```

29 Count duplicate string values using HashMap

Runs in any online compiler

EXPLANATION Increment the value for each occurrence. The sample output gives the count for every distinct key.

Java

```java
import java.util.LinkedHashMap;
import java.util.Map;

public class Main {
    public static void main(String[] args) throws Exception {
        String[] a = {"111", "222", "333", "444", "555", "222", "111", "333", "111", "222"};
        Map<String, Integer> counts = new LinkedHashMap<>();
        for (String v : a)
            counts.put(v, counts.getOrDefault(v, 0) + 1);
        System.out.println(counts);  // {111=3, 222=3, 333=2, 444=1, 555=1}
    }
}
```

30 Read text from PDF

Needs PDFBox

EXPLANATION PDFBox extracts embedded text. Scanned image-only PDFs need OCR.

Java

```java
import java.io.File;
import org.apache.pdfbox.pdmodel.PDDocument;
import org.apache.pdfbox.text.PDFTextStripper;

public class Main {
    public static void main(String[] args) throws Exception {
        try (PDDocument pdf = PDDocument.load(new File("sample.pdf"))) {
            String text = new PDFTextStripper().getText(pdf);
            System.out.println(text);
        }
    }
}
```

> NOTE *For PDFBox 3, use the Loader API appropriate to that version.*

31 Print a pyramid

Runs in any online compiler

EXPLANATION Print leading spaces and 2*i-1 stars on row i.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        int rows = 5;
        for (int i = 1; i <= rows; i++) {
            for (int s = 0; s < rows - i; s++)
                System.out.print(" ");
            for (int j = 0; j < 2 * i - 1; j++)
                System.out.print("*");
            System.out.println();
        }
    }
}
```

32 Sort a string without built-in sorting

Runs in any online compiler

EXPLANATION Bubble-sort the character array by swapping adjacent characters. This uses character code order, not locale-aware sorting.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        char[] a = "selenium".toCharArray();
        for (int i = 0; i < a.length - 1; i++)
            for (int j = 0; j < a.length - 1 - i; j++)
                if (a[j] > a[j + 1]) {
                    char t = a[j];
                    a[j] = a[j + 1];
                    a[j + 1] = t;
                }
        System.out.println(new String(a));
    }
}
```

33 Singleton with double-checked locking

Runs in any online compiler

EXPLANATION Use a volatile instance and synchronize only during first initialization.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        System.out.println(Singleton.getInstance() == Singleton.getInstance());
    }
}

final class Singleton {
    private static volatile Singleton instance;
    private Singleton() {}
    public static Singleton getInstance() {
        if (instance == null)
            synchronized (Singleton.class) {
                if (instance == null)
                    instance = new Singleton();
            }
        return instance;
    }
}
```

> NOTE *Enum singletons are often simpler; volatile is essential for safe publication in this pattern.*

34 Recursive binary search

Runs in any online compiler

EXPLANATION The array must be sorted. Each recursive call halves the search interval, giving O(log n) time.

Java

```java
public class Main {
    static int binarySearch(int[] a, int low, int high, int target) {
        if (low > high)
            return -1;
        int mid = low + (high - low) / 2;
        if (a[mid] == target)
            return mid;
        return target < a[mid]
                ? binarySearch(a, low, mid - 1, target)
                : binarySearch(a, mid + 1, high, target);
    }

    public static void main(String[] args) throws Exception {
        int[] sorted = {1, 3, 5, 7, 9, 11};
        System.out.println(binarySearch(sorted, 0, sorted.length - 1, 7));
    }
}
```

35 Singly linked list: add node and print in reverse

Runs in any online compiler

EXPLANATION Append a node at the tail. Recursion prints nodes in reverse as calls return.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        MyList list = new MyList();
        list.add(1);
        list.add(2);
        list.add(3);
        list.printReverse(list.head);
        System.out.println();
    }
}

class Node {
    int data;
    Node next;
    Node(int d) {
        data = d;
    }
}

class MyList {
    Node head;
    void add(int v) {
        Node n = new Node(v);
        if (head == null) {
            head = n;
            return;
        }
        Node c = head;
        while (c.next != null)
            c = c.next;
        c.next = n;
    }

    void printReverse(Node n) {
        if (n == null)
            return;
        printReverse(n.next);
        System.out.print(n.data + " ");
    }
}
```

NOTE *Recursive reverse printing uses O(n) stack space.*

36 Serialization and deserialization

Runs in any online compiler

EXPLANATION Write a Serializable object using ObjectOutputStream and read it with ObjectInputStream.

Java

```java
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.io.Serializable;

public class Main {
    public static void main(String[] args) throws Exception {
        try (ObjectOutputStream o = new ObjectOutputStream(new FileOutputStream("employee.ser"))) {
            o.writeObject(new Employee("Ravi", 101));
        }
        try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("employee.ser"))) {
            Employee e = (Employee) in.readObject();
            System.out.println(e.name + " " + e.id);
        }
    }
}

class Employee implements Serializable {
    private static final long serialVersionUID = 1L;
    String name;
    int id;
    Employee(String n, int i) {
        name = n;
        id = i;
    }
}
```

> NOTE *Only deserialize trusted data; native Java deserialization can be unsafe.*

37 Even/odd threads using wait() and notifyAll()

Runs in any online compiler

EXPLANATION Use a shared monitor. wait() releases the lock; use while loops around wait to handle spurious wakeups.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        EvenOdd shared = new EvenOdd();
        Thread oddThread = new Thread(() -> {
            try {
                shared.odd();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        });
        Thread evenThread = new Thread(() -> {
            try {
                shared.even();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        });
        oddThread.start();
        evenThread.start();
        oddThread.join();
        evenThread.join();
    }
}

class EvenOdd {
    int n = 1, max = 10;
    synchronized void odd() throws InterruptedException {
        while (n <= max) {
            while (n % 2 == 0 && n <= max)
                wait();
            if (n <= max) {
                System.out.println("Odd " + n++);
                notifyAll();
            }
        }
    }

    synchronized void even() throws InterruptedException {
        while (n <= max) {
            while (n % 2 != 0 && n <= max)
                wait();
            if (n <= max) {
                System.out.println("Even " + n++);
                notifyAll();
            }
        }
    }
}
```

NOTE *Keep the shared state and methods on the same object.*

38 Email validation

Runs in any online compiler

EXPLANATION A simple regex checks common email formats, but full email syntax is complex. Production systems should confirm addresses via email.

Java

```java
public class Main {
    public static void main(String[] args) throws Exception {
        String email = "qa.user@example.com";
        String regex = "^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\\.[A-Za-z]{2,}$";
        System.out.println(email.matches(regex));
    }
}
```

39 Copy contents from one Excel sheet to another

Needs Apache POI

EXPLANATION Use Apache POI to iterate source rows/cells and create matching target cells. This sample copies common cell values/formulas.

Java

```java
import java.io.FileInputStream;
import java.io.FileOutputStream;
import org.apache.poi.ss.usermodel.Cell;
import org.apache.poi.ss.usermodel.Row;
import org.apache.poi.ss.usermodel.Sheet;
import org.apache.poi.ss.usermodel.Workbook;
import org.apache.poi.ss.usermodel.WorkbookFactory;

public class Main {
    public static void main(String[] args) throws Exception {
        try (FileInputStream in = new FileInputStream("workbook.xlsx");
                Workbook wb = WorkbookFactory.create(in)) {
            Sheet src = wb.getSheet("Source"), dst = wb.getSheet("Target");
            for (Row sr : src) {
                Row dr = dst.getRow(sr.getRowNum());
                if (dr == null)
                    dr = dst.createRow(sr.getRowNum());
                for (Cell sc : sr) {
                    Cell dc = dr.createCell(sc.getColumnIndex(), sc.getCellType());
                    switch (sc.getCellType()) {
                        case STRING:
                            dc.setCellValue(sc.getStringCellValue());
                            break;
                        case NUMERIC:
                            dc.setCellValue(sc.getNumericCellValue());
                            break;
                        case BOOLEAN:
                            dc.setCellValue(sc.getBooleanCellValue());
                            break;
                        case FORMULA:
                            dc.setCellFormula(sc.getCellFormula());
                            break;
                        case BLANK:
                            dc.setBlank();
                            break;
                        default:
                            dc.setCellValue(sc.toString());
                    }
                }
            }
            try (FileOutputStream out = new FileOutputStream("workbook.xlsx")) {
                wb.write(out);
            }
        }
    }
}
```

> NOTE *A full copy should also handle styles, merged cells, widths, heights, and sheet settings.*

40 Separate letters, digits, and special characters

Runs in any online compiler

EXPLANATION Use Character.isLetter and Character.isDigit; other characters go to the special-character bucket.

Java

```
public class Main {
    public static void main(String[] args) throws Exception {
        String s = "QA2026@Chennai#";
        StringBuilder letters = new StringBuilder();
        StringBuilder digits = new StringBuilder();
        StringBuilder special = new StringBuilder();
        for (char c : s.toCharArray()) {
            if (Character.isLetter(c))
                letters.append(c);
            else if (Character.isDigit(c))
                digits.append(c);
            else
                special.append(c);
        }
        System.out.println(letters + " | " + digits + " | " + special);
    }
}
```

41 Switch to the last window

Needs Selenium + Chrome

EXPLANATION Iterate through window handles. For a new child window, compare against the handles collected before clicking.

Java

```
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            String last = null;
            for (String h : driver.getWindowHandles())
                last = h;
            if (last != null)
                driver.switchTo().window(last);
        } finally {
            driver.quit();
        }
    }
}
```

42 Connect to a database with JDBC

Needs database

EXPLANATION Use a PreparedStatement and try-with-resources. Read connection values from environment variables.

Java

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Main {
    public static void main(String[] args) throws Exception {
        String url = System.getenv("DB_URL");
        String user = System.getenv("DB_USER");
        String pass = System.getenv("DB_PASSWORD");
        try (Connection c = DriverManager.getConnection(url, user, pass);
                PreparedStatement ps = c.prepareStatement(
                        "SELECT id,name FROM employee WHERE id=?")) {
            ps.setInt(1, 101);
            try (ResultSet rs = ps.executeQuery()) {
                while (rs.next())
                    System.out.println(rs.getInt("id") + " " + rs.getString("name"));
            }
        }
    }
}
```

> NOTE *Import java.sql.* and add the database vendor's JDBC driver.*

43 Check whether an element is visible

Needs Selenium + Chrome

EXPLANATION findElements avoids a NoSuchElementException when no match exists. isDisplayed checks visibility for a found element.

Java

```java
import java.time.Duration;
import java.util.List;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            List<WebElement> found = driver.findElements(By.id("status"));
            boolean visible = !found.isEmpty() && found.get(0).isDisplayed();
            System.out.println(visible);
            // To wait for visibility:
            new WebDriverWait(driver, Duration.ofSeconds(10))
                    .until(ExpectedConditions.visibilityOfElementLocated(By.id("status")));
        } finally {
            driver.quit();
        }
    }
}
```

44 Count broken links on a page

Needs Selenium + Chrome

EXPLANATION Collect hrefs and check HTTP status codes. Some servers block HEAD requests or require authentication; a 403/405 is not always a broken link.

Java

```java
import java.net.URI;
import java.net.http.HttpClient;
import java.net.http.HttpRequest;
import java.net.http.HttpResponse;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            HttpClient client = HttpClient.newBuilder()
                    .followRedirects(HttpClient.Redirect.NORMAL)
                    .build();
            int broken = 0;
            for (WebElement a : driver.findElements(By.cssSelector("a[href]"))) {
                String href = a.getAttribute("href");
                if (href == null || !(href.startsWith("http://") || href.startsWith("https://")))
                    continue;
                try {
                    HttpRequest req = HttpRequest.newBuilder(URI.create(href))
                            .timeout(Duration.ofSeconds(8))
                            .method("HEAD", HttpRequest.BodyPublishers.noBody()).build();
                    int status = client.send(req, HttpResponse.BodyHandlers.discarding())
                            .statusCode();
                    if (status >= 400)
                        broken++;
                } catch (Exception e) {
                    broken++;
                }
            }
            System.out.println("Potential broken links: " + broken);
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *Java 11+; retry with GET for servers that reject HEAD, and interpret results carefully.*

45 Run same code twice and finish within 3 seconds

Runs in any online compiler

EXPLANATION Run the method twice and measure elapsed time. The time limit cannot be guaranteed if work depends on a slow external system.

Java

```java
public class Main {
    static void runTask() throws InterruptedException {
        Thread.sleep(500); // simulated work
    }

    public static void main(String[] args) throws Exception {
        long start = System.nanoTime();
        for (int i = 0; i < 2; i++)
            runTask();
        long ms = (System.nanoTime() - start) / 1_000_000;
        System.out.println(ms + " ms");
        if (ms > 3000)
            throw new AssertionError("Exceeded 3 seconds");
        // static void runTask() { /* task to test */ }
    }
}
```

> NOTE *For Selenium, avoid fixed sleeps and use explicit waits. Do not use Thread.stop.*

46 Run a package but exclude two TestNG methods

Config file

EXPLANATION Use a TestNG suite XML to control included classes and excluded methods without editing source. Method exclusions are commonly nested under each class.

XML

```xml
<!DOCTYPE suite SYSTEM "https://testng.org/testng-1.0.dtd">
<suite name="Suite">
    <test name="PackageTests">
        <classes>
            <class name="com.example.tests.LoginTest">
                <methods>
                    <exclude name="testToSkipOne"/>
                    <exclude name="testToSkipTwo"/>
                </methods>
            </class>
            <class name="com.example.tests.SearchTest"/>
        </classes>
    </test>
</suite>
```

> NOTE *If you need package scanning, list the classes explicitly when the suite structure/version does not apply excludes at package level.*

47 Interact inside a frame, then click outside it

Needs Selenium + Chrome

EXPLANATION Switch to the frame, interact, then return to defaultContent before locating an outer-page element.

Java

```java
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            driver.switchTo().frame(driver.findElement(By.cssSelector("iframe#detailsFrame")));
            driver.findElement(By.id("insideFrameButton")).click();
            driver.switchTo().defaultContent();
            driver.findElement(By.id("outsideFrameLink")).click();
        } finally {
            driver.quit();
        }
    }
}
```

48 Scroll a web page

Needs Selenium + Chrome

EXPLANATION JavascriptExecutor can scroll by pixels or bring a specific element into view.

Java

```java
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.WebElement;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            JavascriptExecutor js = (JavascriptExecutor) driver;
            js.executeScript("window.scrollBy(0,600);");
            WebElement footer = driver.findElement(By.id("footer"));
            js.executeScript("arguments[0].scrollIntoView({block:'center'});", footer);
        } finally {
            driver.quit();
        }
    }
}
```

49 Handle alerts

Needs Selenium + Chrome

EXPLANATION Wait for the alert, read its text, and accept or dismiss. For prompt alerts, send text before accepting.

Java

```java
import java.time.Duration;
import org.openqa.selenium.Alert;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            Alert alert = new WebDriverWait(driver, Duration.ofSeconds(5))
                    .until(ExpectedConditions.alertIsPresent());
            System.out.println(alert.getText());
            alert.accept();
            // Prompt: alert.sendKeys("input"); alert.accept();
        } finally {
            driver.quit();
        }
    }
}
```

50 Check whether a page is loaded

Needs Selenium + Chrome

EXPLANATION Wait for document.readyState=complete, then wait for a page-specific element because single-page apps may fetch data afterwards.

Java

```java
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.JavascriptExecutor;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(15));
            wait.until(d -> ((JavascriptExecutor) d)
                    .executeScript("return document.readyState").equals("complete"));
            wait.until(ExpectedConditions.visibilityOfElementLocated(By.id("mainContent")));
        } finally {
            driver.quit();
        }
    }
}
```

51 Upload an image file

Needs Selenium + Chrome

EXPLANATION For an HTML file input, send the absolute file path directly; the OS file chooser usually does not need to be automated.

Java

```java
import java.nio.file.Files;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Duration;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.support.ui.ExpectedConditions;
import org.openqa.selenium.support.ui.WebDriverWait;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            Path path = Paths.get("src/test/resources/images/sample.png").toAbsolutePath();
            if (!Files.exists(path))
                throw new IllegalArgumentException("File missing: " + path);
            driver.findElement(By.cssSelector("input[type='file']")).sendKeys(path.toString());
            driver.findElement(By.id("uploadButton")).click();
            new WebDriverWait(driver, Duration.ofSeconds(10))
                    .until(ExpectedConditions.visibilityOfElementLocated(By.id("uploadSuccess")));
        } finally {
            driver.quit();
        }
    }
}
```

52 Find the Nth largest number

Runs in any online compiler

EXPLANATION This solution treats Nth largest as Nth distinct value using a reverse-ordered TreeSet.

Java

```
import java.util.Comparator;
import java.util.TreeSet;

public class Main {
    public static void main(String[] args) throws Exception {
        int[] a = {7, 2, 9, 9, 5, 1};
        int n = 3;
        TreeSet<Integer> sorted = new TreeSet<>(Comparator.reverseOrder());
        for (int x : a)
            sorted.add(x);
        if (n < 1 || n > sorted.size())
            throw new IllegalArgumentException("Invalid N");
        int i = 1, answer = 0;
        for (int x : sorted)
            if (i++ == n) {
                answer = x;
                break;
            }
        System.out.println(answer);  // 5
    }
}
```

NOTE *Clarify whether duplicate values count as separate ranks.*

53 Sum the first 1000 prime numbers

Runs in any online compiler

EXPLANATION Generate candidates until 1000 primes have been found, then add them to a long.

Java

```
public class Main {
    static boolean prime(int n) {
        if (n < 2)
            return false;
        for (int d = 2; d <= n / d; d++)
            if (n % d == 0)
                return false;
        return true;
    }

    public static void main(String[] args) throws Exception {
        long sum = 0;
        int found = 0, candidate = 2;
        while (found < 1000) {
            if (prime(candidate)) {
                sum += candidate;
                found++;
            }
            candidate++;
        }
        System.out.println(sum);  // 3682913
    }
}
```

54 Print all permutations of a string

Runs in any online compiler

EXPLANATION Backtracking swaps each candidate into the current position, recurses, then swaps back. The output count is n!.

Java

```java
public class Main {
    static void permute(char[] a, int index) {
        if (index == a.length) {
            System.out.println(new String(a));
            return;
        }
        for (int i = index; i < a.length; i++) {
            char t = a[index];
            a[index] = a[i];
            a[i] = t;
            permute(a, index + 1);
            t = a[index];
            a[index] = a[i];
            a[i] = t;
        }
    }

    public static void main(String[] args) throws Exception {
        permute("ABC".toCharArray(), 0);
    }
}
```

NOTE *Repeated input characters can produce duplicate permutations; use a set if unique outputs are required.*

55 Generate a unique email address each run

Needs Selenium + Chrome

EXPLANATION Combine a timestamp and UUID to reduce collisions. Use a controlled test domain rather than a real recipient domain.

Java

```java
import java.time.Instant;
import java.util.UUID;
import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class Main {
    public static void main(String[] args) throws Exception {
        WebDriver driver = new ChromeDriver();
        try {
            driver.manage().window().maximize();
            driver.get("https://example.com"); // replace with your application URL
            String email = "qa+" + Instant.now().toEpochMilli() + "-"
                    + UUID.randomUUID().toString().substring(0, 8) + "@example.test";
            driver.findElement(By.id("email")).sendKeys(email);
            System.out.println(email);
        } finally {
            driver.quit();
        }
    }
}
```

NOTE *Imports: java.time.Instant and java.util.UUID. Replace example.test only with a domain your test application accepts.*

Interview reminders

- Explain edge cases such as empty input, duplicates, invalid N, negative values, and timeouts.
- For Selenium, use stable locators, explicit waits, and assertions that validate the business outcome.
- Close WebDriver and resources reliably. Use try-with-resources for files, streams, and JDBC objects.
- Keep passwords, tokens, and environment-specific settings outside source code.