Selenium WebDriver with Java Interview questions:

1. What are the different types of Locator Strategy supported by Selenium?

Solution:

Selenium supports almost all types of locators.

- **Class name:** Locates the elements using Class Name
- **CSS Selector:** Locates the Element using CSS selectors
- **ID:** Locates the Element using the ID attribute
- **Name:** Locates elements whose NAME attribute matches the search value
- **Link Text:** Locates anchor elements whose visible text matches the search value
- **Partial Link Text**: Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
- **Tag Name:** Locates elements whose tag name matches the search value
- **Xpath:** Locates elements matching an XPath expression

2. Explain at least 5 different types of exceptions in Selenium.

Solution:

NoSuchElementException
Reason: The exception occurs when WebDriver is unable to find and locate elements. The incorrect locator can cause this exception in selenium.

TimeoutException
Reason: Selenium waits until a specific time period, If the element could not display or load in specified time.

WebDriverException:
Reason: Webdriver performs actions immediately after 'closing' the browser or it can also occur when our code unable to initialize the webdriver

ElementNotVisibleException
Reason: Selenium tries to perform action on invisible Elements

NoSuchSessionException
Reason: When webdriver cannot execute the commands using driver instance.

NoAlertPresentException
Reason: The user is trying to handle an alert that is not present.

NoSuchWindowException
Reason: WebDriver tries to switch to an invalid window.

3. I have a use case that I need to execute some of the tests only once, I don't have to execute these tests in the future, Should I use selenium WebDriver for this?

Solution:

The Standard Practice for Automation Testing is to Automate only repetitive tasks, as creating and setting up Automation Tests involves cost and effort. In the above use case, the test is performed only once so it is not recommended to use automation.

4. Can we use Selenium for Product Development? If not, what is selenium used for?

Solution:

Selenium is Not a Product Development Framework; Selenium is Automation Testing Framework. Selenium can be used for testing Web Applications.

5. Explain Explicit Implicit and Fluent waits in Selenium?

Solution:

Implicit Wait:

If Selenium could not find the element it throws No Such Element Exception, but we can define, before throwing an exception how long Selenium should wait. This can be done using Selenium Implicit Wait. If we don't define anything Selenium Assumes it as 0.

Example:

//Wait for 10 seconds before throwing exceptions

```
driver.manage().timeouts().implicitlyWait(10,TimeUnit.SECONDS) ;
```

Note: Implicit wait is set as driver level, not specific to any element.

Explicit Wait:

Explicit Wait is used when the specific element in the webpage is taking a long time to load or appear. Usually, this wait is set at the element level. The explicit wait is always used with Expected Conditions.

Selenium First checks if the Expected Condition returns true, if not then wait for a short duration and again checks, if the time duration gets over then Selenium throws the exception.

Example:

```
WebDriverWait wait = new WebDriverWait(driver,30);
wait.until(ExpectedConditions.visibilityOfElementLocated(By.xpath("//div[c
ontains(text(),'Rahul Shetty Academy')]")));
```

Fluent Wait:

Fluent Wait looks for a web element repeatedly at regular intervals until timeout happens or until the element is found. Fluent Wait commands are helpful when interacting with web elements that can take longer durations to load.

Example:

```
// Waiting 30 seconds for an element to be present on the page, checking
// for its presence once every 5 seconds.
Wait<WebDriver> wait = new FluentWait<WebDriver>(driver)
  .withTimeout(Duration.ofSeconds(30))
  .pollingEvery(Duration.ofSeconds(5))
  .ignoring(NoSuchElementException.class);

WebElement foo = wait.until(new Function<WebDriver, WebElement>() {
  public WebElement apply(WebDriver driver) {
    return driver.findElement(By.id("foo"));
  }
});
```

6. What is Selenium IDE? when should we use it?

Solution:

The Selenium-IDE (Integrated Development Environment) is the tool, used to develop your Selenium test cases. Selenium IDE is useful for beginners, where they want to write less code and actions can be done using User Interface (UI).

7. What is Automation Testing?

Solution:

Automation Testing is Software Testing Technique where the actual and expected outcome is validated using Programmatically using Automation Tools.

8. What is Selenese?

Solution:

Selenium is the Web Automation Tool, which provides a set of Selenium commands to test the web applications. These Selenium Commands are often referred to as Selenese.

9. Explain List of Supported Drivers by Selenium?

Solution:

Selenium Communicates with browser using drivers, Each browser has its specific drivers.

- **Chrome Driver:** Chrome/ Chromium based browsers
- **GeckoDriver:** Firefox Browsers
- **InternetExplorerDriver:** Internet Explorer Browser
- **Safari Driver:** Safari Browser
- **Edge Driver:** Edge Browser

10. How to read data from excel in selenium webdriver?

Solution:

Selenium Webdriver doesn't provide any functionality to work with an excel file. However, reading and writing data into excel can be done using Java Utility called POI
Apache POI is an open-source java library used to handle Microsoft Office-based files. This helps the user to perform the operation on excel files such as read, write, etc.

Once you add the Apache POI jar to your project you can use the library utilities to read/write data into Excel File.

Example:

```
FileInputStream fs = new FileInputStream("D:\\Example.xlsx");
XSSFWorkbook workbook = new XSSFWorkbook(fs);
XSSFSheet sheet = workbook.getSheetAt(0);
Row row = sheet.getRow(0);
Cell cell = row.getCell(0);
System.out.println(sheet.getRow(0).getCell(0));
```

11. What is the difference between Assert and Verify.

Solution:

Assert: If the condition fails, then the test will be terminated. There will not be further execution of the test step

Verify: If the condition fails, Execution will continue as normally there will be error logs. Execution will not be terminated.

12. Which method is the overloaded method selenium webdriver?

Solution:

The most used overloaded method in Selenium is:

- **frame() Method:** This method is helpful when we work with iFrame. Based on the parameter we pass Index, WebElement, or name the respective implementations are called.
- Example:
- **By Index**
- driver.switchTo().frame(0);
- **By Name or Id**
- driver.switchTo().frame("iframe1")

- **By Web Element**
- WebElement iframe = driver.findElement(By.id("Frame1"));
- driver.switchTo().frame(iframe);

13. How to handle window dialog such as File Open Dialog in Windows using Selenium?

Solution:

Selenium supports only Web Application Automation, Window-based dialog cannot be automated using Selenium. However integrating with tools like Autoit, Robot Class will be helpful in this case.

14. What is By in selenium?

Solution:

By is the class in Selenium, it provides the locater specific methods and can be used with driver.findElement(s) to get the Element.
List of Important Methods provided by By Class are

Id()
linkText()
partialLinkText()
name()
tagName()
xpath()
className()
cssSelector()

15. What are Hard and Soft Asserts in selenium?

Solution:

Hard Assert: Hard assert throws the exception as soon as the condition fails and execution for the current test case will be stopped and continues with the next test case execution.
Soft Assert: Soft Assert doesn't throw an exception when the condition fails.

16. How to handle hidden element in Selenium?

Solution:

A hidden element means that Element is present in DOM but it is not visible in the browser. To handle the hidden element in Selenium, we need to use the JavaScript Executor.
For Example:
Consider we have a hidden text box with id="hiddenbox1".
JavascriptExecutor executor = (JavascriptExecutor) driver;
String str = (String) executor.executeScript("return document.getElementById('hiddenbox1').value");
Though, the hiddenbox1 is not visible on the browser, since the text box is attached to DOM we can get the value of the text box using the above method.

17. What is the Action Class in Selenium?

Solution:

Action classes provide the ability to simulate mouse and keyboard events. Example Mousehover, Drag and drop, etc.

```
Actions action = new Actions(driver);
action.moveToElement(element).click().perform();
```

18. What are the different types of frameworks?

Solution:

Page Object Model: Page Object Model (POM), is a design pattern in Selenium that creates an object repository for storing all web elements. It is useful in reducing code duplication and maintenance**.** In-Page Object Model, Each web page of an application as a class file. Each class file will contain only corresponding web page elements

Data-Driven Framework: When the entire test data is generated from some external files like Excel, CSV, XML, or some database table, then it is called Data Driven framework

Keyword Driven Framework: When only the instructions and operations are written in a different file like an Excel worksheet, it is called Keyword Driven framework.

Hybrid Framework: Any combination above frameworks like POM, Data-Driven, Keyword Driven is called a Hybrid Framework.

19. Explain Selenium Browser Naivigation commands.

Solution:

driver.get() : Used to navigating to webpage

Example:

driver.get("https://rahulshettyacademy.com/");

driver.getCurrentUrl(): Reads the current URL from the browser's address.

driver.navigate().back(); : Navigates back to previous page. Equivalant to clicking browser back button.

driver.navigate().forward() : Navigates forwards way. Equivalant to clicking browser forward button.

driver.navigate().refresh(); : Refreshs the current page

20. Give an example to perform drag and drop action In Selenium WebDriver?

Solution:

```
Actions actions = new Actions(driver);
WebElement from = driver.findElement(By.id("column-a "));
WebElement to = driver.findElement(By.id("column-b"));
actions.dragAndDrop(from, to).build().perform();
```

21. How can I switch to Multiple Windows in Selenium?

Solution:

Selenium supports working with Multiple Tabs and Windows. switchTo() method is mainly used for Switching between Tabs or windows.

```
ArrayList<String> newTab = new
ArrayList<String>(driver.getWindowHandles());
//switch to new tab
driver.switchTo().window(newTab.get(1));
```

22. How can I Capture Screenshot in Selenium 4?

Solution:

```
    //Taking screenshot of WebPage
WebDriver driver = new ChromeDriver();
 driver.get("http://www.example.com");
 File scrFile = ((TakesScreenshot)driver).getScreenshotAs(OutputType.FILE);
 FileUtils.copyFile(scrFile, new File("./image.png"));
//Taking screenshot of WebElement
WebDriver driver = new ChromeDriver();
driver.get("https://www.example.com");
WebElement element = driver.findElement(By.cssSelector("h1"));
File scrFile = element.getScreenshotAs(OutputType.FILE);
FileUtils.copyFile(scrFile, new File("./image.png"));
```

23. What are the different languages supported by Selenium?

Solution:

Selenium Supports Following Languages

- C#
- Java
- Python
- Ruby
- Javascript

24. What is Selenium Webdriver?

Solution:

Selenium Webdriver is an Open source tool that contains a collection of APIs which is used to verify the result of test cases programmatically.
Or.
Selenium Webdriver is an Open Source Automation Tool, which contains a collection of API's which is used to test the web application using supported programming languages.

25. How to handle alerts pop up in Selenium?

Solution:

driver.switchTo().alert() is used for Switching to Browser Alert.

Accept Alert :
driver.switchTo().alert().accept();
Dismiss Alert:
driver.switchTo().alert().dismiss();
Get Text:
driver.switchTo().alert().getText();
Type Text:
driver.switchTo().alert().sendKeys("Text");

26. Write a Small Code Snippet Launch a browser and Navigate to Webpage and Close the browser.

Solution:

```
System.setProperty("webdriver.chrome.driver","C://chromedriver.exe");
driver= new ChromeDriver();
 driver.get("https://rahulshettyacademy.com/");
```

27. Explain Javascript Executor in Selenium.

Solution:

Javascript executor helps to execute Javascript on the browser. Sometimes the default simulation events might not work click(), getText(), etc. Using Javascript executor we can make it work.

Example:

```
JavascriptExecutor js = (JavascriptExecutor) driver;
js.executeScript("document.getElementByID('element id ').click();");
```

28. Can I Create a Selenium Java framework without using TestNG?

Solution:

TestNG and Selenium are two different things, TestNG helps us to manage Assertions, Reporters and makes things easier for creating the complete framework. We can use other frameworks like Junit to create Selenium Framework.

29. Does Selenium support IFrames, If so please explain.

Solution:

- Selenium Supports Iframe, We can Switch To and From Iframe.
- Example Code
- We can Switch to Iframe using the below techniques.
- **By Index**
- Example:

```
driver.switchTo().frame(0);
```

- **By Name or Id**

```
driver.switchTo().frame("iframe1")
```

- **By Web Element**

```
WebElement iframe = driver.findElement(By.id("Frame1"));
driver.switchTo().frame(iframe);
```

30. I need to press Keys CTRL + SHIFT + S how can I do that in Selenium?

Solution:

```
Actions action = new Actions(driver);

action
.keyDown(Keys.SHIFT).keyDown(Keys.CONTROL).sendKeys("S").build().per
form();
```

31. Write a Java Program to find the longest consecutive occurrence of integers in a given array.

Solution:

In this problem, we need to find the length of the longest consecutive occurrence.

As given in the array below, we have an array of integers. As you can see the longest consecutive occurrence of integers are 6,7,8,9. There is a consecutive increment of 1 . So the output will be 4.

```
package JavaInterviewQuestions;

public class LongestConsecutiveOccurenceExample {

public static void main(String[] args) {

int[] arr = { 4, 3, 25, 6, 7, 8, 9, 2, 3, 10 };

int count = 0;
int max =0;

for(int i = 0; i < arr.length-1; i++) {
if(arr[i] +1== arr[i+1]) {
count++;
}else {
count = 0;
}

max = Math.max(max, count+1);
}
System.out.println(max);
}

}
```

The above program will give the output as 4. We use the Math.max() to obtain the maximum value between the max and the counter variable.

32. Write a Java Program to reverse a String.

Solution:

As asked in the above question, we need to reverse a String. **For Ex:**
The Input is : "RahulShettyAcademy"
Output should be : "ymedacAyttehSluhaR"
This problem can be solved by writing a simple for loop and print the string in the reverse.

```
public class ReverseAStringExample {

public static void main(String[] args) {

String inputString = "RahulShettyAcademy";
String outputString = "";

/*
* Writing a for loop and looping in the reverse order to get the reverse
String
* as output.
*/

for (int i = inputString.length() - 1; i >= 0; i--) {
outputString = outputString + inputString.charAt(i);
}

System.out.println(outputString);
}

}
```

This will print the outputString which will be the reverse of the given inputString.

33. Write a Java Program to get the count of Capitalized words in a String.

Solution:

Capitalized words in a given String which are those words which starts with a Capital Letter.
We need the total count of these words. This problem can be solved by using a for loop and iterate through each character in the String.
We also take a counter variable and increment it when any Capital character is encountered.

```
package JavaInterviewQuestions;

public class CountCapitalizedWordsInString {

public static int getCapsWordsInString(String inputString) {

int counter = 0;
for(int i = 0; i < inputString.length(); i++) {
if(inputString.charAt(i) >= 'A' && inputString.charAt(i) <= 'Z') {
counter++;
}
}
return counter;
}

public static void main(String[] args) {
System.out.println(getCapsWordsInString("RahulShettyAcademy"));
}

}
```

The above program will return the total count of Capitalized String i.e. 3.

34. Write a Java Program to swap two given Strings.

Solution:

As asked in the above question, for swapping of two strings we need to use the substring method of the String class in Java.
Substring method returns part of the string.
We have the start index and end index in the substring method where start index is inclusive and end index is exclusive.
Below is the Java Program to swap two Strings.

```
package JavaInterviewQuestions;

public class StringSwapExample {

public static void main(String[] args) {

// Take two string s1 and s2 which we need to swap.
String s1 = "Rahul";
String s2 = "Shetty";

// Combine both the strings s1 and s2 using the concatenation(+)
operator
s1 = s1 + s2;

// Use the substring method to get the subset of the combined string
s2 = s1.substring(0, s1.length() - s2.length());
s1 = s1.substring(s2.length());

System.out.println("s1 =" + s1);
System.out.println("s2 =" + s2);

}

}
```

The above program will swap the two Strings s1 and s2.

35. Write a Java Program to reverse an array?

Solution:

In this problem, we are given an array of characters as input. This problem can be solved by writing a while loop.
By using a third variable, we reverse the array as below.

```
package JavaInterviewQuestions;

public class ReverseAnArray {

public static void main(String[] args) {

char[] s = { 'a', 'b', 'c', 'd', 'e' };

int right = s.length - 1;
int left = 0;

while (left < right) {
char c = s[left];
s[left] = s[right];
s[right] = c;

// Increment the left by 1 and Decrement the right by 1
left += 1;
right -= 1;
}

for (int i = 0; i < s.length; i++) {
System.out.print(s[i] + " ");
}

}

}
```

The above program will return the total count of Capitalized String i.e. 3.

36. Write a Java Program to convert HashMap to ArrayList.

Solution:

To solve this problem, we can convert the HashMap keys into an ArrayList and then the HashMap values into an ArrayList.

```
package JavaInterviewQuestions;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

public class HashMapToArrayListExample {

public static void main(String[] args) {

HashMap<String, String> hash_map = new HashMap<>();

hash_map.put("India", "New Delhi");
hash_map.put("France", "Paris");
hash_map.put("Germany", "Berlin");
hash_map.put("Australia", "Canberra");

// convert hashmap keys to ArrayList

List<String> countryNameList = new ArrayList<>(hash_map.keySet());
System.out.println("HashMap keys are: ");
for (String s : countryNameList) {
System.out.println(s);
}

System.out.println("**********************");

// convert hashmap values to ArrayList
List<String> capitalNameList = new ArrayList<String>
(hash_map.values());
System.out.println("HashMap values are: ");
for (String s : capitalNameList) {
System.out.println(s);
}

}

}
```

37. Write a Java Program to print the product of an array except self?

Solution:

In this problem, We need the output which is the product of the all the numbers in the array except self.
Here we take two arrays, left_products and right_products in which we store the products of all numbers except self.
Then we multiply the left_products and right_products to get the output.

Below is the Java Program:

```
package JavaInterviewQuestions;

public class ProductArrayExceptSelfExample {

public static void main(String[] args) {

int[] arr = {1,2,3,4};
int N = arr.length;

int[] left_products = new int[N];
int[] right_products = new int[N];

int[] output_array = new int[N];
left_products[0] = 1;
right_products[N-1] = 1;

for(int i = 1; i < N; i++) {
left_products[i] = arr[i-1] * left_products[i-1];
}

for(int i = N-2; i >=0; i--) {
right_products[i] = arr[i+1] * right_products[i+1];
}

for(int i = 0; i < N; i++) {
output_array[i] = left_products[i] * right_products[i];
}

for(int i =0; i < N; i++) {
System.out.print(output_array[i] + " ");
}

}

}
```

38. Write a Java Program to generate Output "aabbbcccc" with the input "a2b3c4"

Solution:

In the above problem, we have the numbers in the input. We need to print the as many characters same the number value.

```
package JavaInterviewQuestions;

public class ConvertNumricToCharExample {

static void convertNumToChar(String s) {

for(int i =0; i < s.length(); i++) {
if(Character.isAlphabetic(s.charAt(i))){
System.out.print(s.charAt(i));
}else {
int a = Character.getNumericValue(s.charAt(i));
for(int j =1; j <a; j++) {
System.out.print(s.charAt(i-1));
}
}
}

}

public static void main(String[] args) {

String str = "a2b3c4"; //output = aabbbcccc
convertNumToChar(str);

}

}
```

39. Write a Java Program to print the second largest number in a given array.

Solution:

In this problem, to find the second largest number in the array, first we sort the array and then using the for loop we find the second largest number in reverse order.

Below is the Java Program:

```
package JavaInterviewQuestions;

public class SecondLargestElementOfArray {

public static void main(String[] args) {

int[] arr = {2,3,1,6,9,9};
int temp;
for (int i = 0; i < arr.length; i++) {
for (int j = i + 1; j < arr.length; j++) {
if (arr[i] > arr[j]) {
temp = arr[i];
arr[i] = arr[j];
arr[j] = temp;
}
}
}

for(int i = arr.length -2; i >=0; i--) {
if(arr[i] != arr[arr.length-1]) {
System.out.println("The second largest element " + arr[i]);
break;
}
}

}

}
```

The output of the above program is 6. In this it also checks that the second largest number is not just based on index. It should be the second largest number in the array.

40. Write a Java Program to print numbers 1 to 100 without using any loop(for/while/do-while).

Solution:

In this problem, to print the numbers between 1 to 100 we will use the if statement and check the condition.
We will check the condition i.e. num value should be less than equal to 100 and then increment it till it reaches to 100 and thus printing each number.

```
package JavaInterviewQuestions;

public class PrintNumWithoutUsingLoopExample {

public static void main(String[] args) {
printnum(1);

}

public static void printnum(int num) {
if (num <= 100) {
System.out.println(num);
num++;
printnum(num);

}

}

}
```

41. What are the limitations of Selenium testing?

Solution:
Unavailability of reliable tech support: Since Selenium is an open-source tool, it does not have dedicated tech support to resolve the user queries.

1. Tests web applications only: Selenium needs to be integrated with third-party tools like Appium and TestNG to test desktop and mobile applications.
2. Limited support for image testing.
3. No built-in reporting and test management facility: Selenium has to be integrated with tools like TestNG, or JUnit among others to facilitate test reporting and management.
4. May require the knowledge of programming languages: Selenium WebDriver expects the user to have some basic knowledge about programming.

42. What are the testing types supported by Selenium?

Solution:
Selenium supports Regression testing and Functional testing.

Regression testing - It is a full or partial selection of already executed test cases that are re-executed to ensure existing functionalities work fine.

The steps involved are -

Re-testing: All tests in the existing test suite are executed. It proves to be very expensive and time-consuming.

Regression test selection: Tests are classified as feature tests, integration tests, and the end to end tests. In this step, some of the tests are selected.

Prioritization of test cases: The selected test cases are prioritized based on business impact and critical functionalities.

Functional testing - Functional Testing involves the verification of every function of the application with the required specification.

The following are the steps involved:

Identify test input
Compute test outcome
Execute test
Compare the test outcome with the actual outcome

43. Mention the types of navigation commands?

Solution:

driver.navigate().to("https://www.rahulshettyacademy.com/"); - Navigates to the provided URL

driver.navigate().refresh(); - This method refreshes the current page

driver.navigate().forward(); - This method does the same operation as clicking on the Forward Button of any browser. It neither accepts nor returns anything.

driver.navigate().back(); - This method does the same operation as clicking on the Back Button of any browser. It neither accepts nor returns anything.

44. What is the major difference between driver.close() and driver.quit()?

Solution:

driver.close()

This command closes the browser's current window. If multiple windows are open, the current window of focus will be closed.

driver.quit()

When quit() is called on the driver instance and there are one or more browser windows open, it closes all the open browser windows.

45. How to type text in an input box using Selenium?

Solution:

sendKeys() is the method used to type text in input boxes

Consider the following example -

WebElement email = driver.findElement(By.id("email")); - Finds the "email" text using the ID locator

email.sendKeys("abcd.efgh@gmail.com"); - Enters text into the URL field

WebElement password = driver.findElement(By.id("Password")); - Finds the "password" text using the ID locator

password.sendKeys("abcdefgh123"); - Enters text into the password field

46. How to click on a hyperlink in Selenium?

Solution:

driver.findElement(By.linkText("Today's deals")).click();

The command finds the element using link text and then clicks on that element, where after the user would be redirected to the corresponding page.

driver.findElement(By.partialLinkText("Service")).click();

The above command finds the element based on the substring of the link provided in the parenthesis and thus partialLinkText() finds the web element.

47. How many types of memory areas are allocated by JVM?

Solution:

Class(Method) Area: Class Area stores per-class structures such as the runtime constant pool, field, method data, and the code for methods.

Heap: It is the runtime data area in which the memory is allocated to the objects

Stack: Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return. Each thread has a private JVM stack, created at the same time as the thread. A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

Program Counter Register: PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.

Native Method Stack: It contains all the native methods used in the application.

48. What is JIT compiler?

Solution:

Just-In-Time(JIT) compiler: It is used to improve the performance. JIT compiles parts of the bytecode that have similar functionality at the same time, and hence reduces the amount of time needed for compilation.

Here the term “compiler” refers to a translator from the instruction set of a Java virtual machine (JVM) to the instruction set of a specific CPU.

49. What is classloader?

Solution:

Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader. There are three built-in classloaders in Java.

Bootstrap ClassLoader: This is the first classloader which is the superclass of Extension classloader. It loads the rt.jar file which contains all class files of Java Standard Edition like java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes, etc.

Extension ClassLoader: This is the child classloader of Bootstrap and parent classloader of System classloader. It loads the jar files located inside $JAVA_HOME/jre/lib/ext directory.

System/Application ClassLoader: This is the child classloader of Extension classloader. It loads the class files from the classpath. By default, the classpath is set to the current directory. You can change the classpath using "-cp" or "-classpath" switch. It is also known as Application classloader.

50. What if I write static public void instead of public static void?

Solution:

The program compiles and runs correctly because the order of specifiers doesn't matter in Java.