CORE JAVA QUESTIONS

1. what is meant by abstraction?

- Hiding the implementation part is called abstraction
- it has 2 types,

1.Partially abstraction(abstract class)

2.Fully abstraction(interface)

Abstract class:

- It is partially abstraction
- It support both abstract method and non-abstract method
- It’s using "extends" keyword
- Here "public abstract" have to mention

Interface:

- It is fully abstraction
- It will support only abstract method
- It’s using "implement" keyword
- "Public Abstract" is default. no need to mention

2. What is the difference between hashtable and hashmap?

Hash table	**Hash map**
1.Both key and value won’t allow null	**1.** key will allow single null and value allow more than one null.
2. Synchronized	**2** Asynchronized
3.It’s a thread safe.	**3**.It’s not a thread safe.

3.Explain about OOPs concept?

- Object Oriented Programing Structure
- OOPS is a method of implementation in which programs are organized as collection of objects, class and methods

Oops principles are

1. Class
2. Method
3. Object
4. Abstraction
5. Encapsulation
6. Inheritance
7. Polymorphism

4. Define interface and Abstract class?

Partial Abstraction (Abstract class):

- It will support abstract method and non-abstract method.
    - We can’t create object for abstract class because in the method signature we didn't mention any business logic. so
    - In abstract method, we only mention abstract signature, won't create business logic

Fully abstraction (interface)

- It will support only abstract method, won't support non abstract method
- In interface "public abstract" is default. we no need to mention
- It using implements keywords

5. Why multiple inheritance does not support in java?

- More than one parent class parallely support into one child class but it won't support in java because
    - Priority problem
    - Compilation error/syntax error

(i.e.) if both parent class having same method name it gets priority problem so it doesn't work in java

- but multiple inheritance support in java using interface
- here we have to create 2 interface(super class) and one sub class(normal). In the sub class we implement both interface

Example Program

```
        interface
        public interface AxisBank {
        public void test();
        }
        public interface HdfcBank {
        public void test();
        }
        sub class(normal class)
        public class Bank implements AxisBank, HdfcBank{
        @Override
        public void test() {
        // TODO Autogenerated method stub
        }
    }
```

6. What is string?

Collections of character or word enclosed with double quotes is called string

7. Difference between list and set and map?

List	**Set**	**Map**
1. It is all insertion order 2. It will allow duplicate value 3. It is index based	1. It's not maintaining any order Hashset→ Random order LinkedHashset→Insertion order Treeset→ Ascending order 2. It does not allows duplicate value 3. It is value based	1. Its follow different order Hashset→ Random order LinkedHashset→Insertion order Treeset→ Ascending order Hashtable→ Random order Concurrent hashmap→ Random order. 2. It doesn't allow duplicate value 3. It is key and value pair

8. Write a code for literal string?

```
    String s1 = "vengat";
    String s2 = "vengat";
    System.out.println(System.identityHashCode(s1));
System.out.println(System.identityHashCode(s2));
```

Output:

```
31168322   // literal string share the memory if same value
31168322
```

9. Difference between final and finally?

Final	**Finally**
1. A final class variable whose value cannot be changed. **2.** A final is declared as class level, they cannot be inherited. **3.** If final is declared as method level, they cannot be override.	**1.** It's a block of statement that definitely executes after the try catch block. **2.** If try block fails means, the final block will executes once.

10. What is method overloading and method overriding?

Method overloading:

- In a same class method name is same and the argument is different is called method overloading
- The argument is depends on
  - data types
  - data types count
  - data type order

Method overriding:

- In a different class , the method name should be same and argument name should be same is called overriding

Selenium and framework questions

1.J unit Annotations and explain?

1. **@Test:** Annotation lets the system know that the method annotated as @Test is a test method. There can be multiple test methods in a single test script.
2. **@Before:** Method annotated as @Before lets the system know that this method shall be executed every time before each of the test method.
3. **@After:** Method annotated as @After lets the system know that this method shall be executed every time after each of the test method.
4. **@BeforeClass:** Method annotated as @BeforeClass lets the system know that this method shall be executed once before any of the test method.
5. **@AfterClass:** Method annotated as @AfterClass lets the system know that this method shall be executed once after any of the test method.
6. **@Ignore:** Method annotated as @Ignore lets the system know that this method shall not be executed.

4. How will you take screenshot & write a code ?

```
TakesScreenshot t=(TakesScreenshot) driver;
File f1 = t.getScreenshotAs(OutputType.FILE);
File f2=new File("D:/screeshot.png");
FileUtils.copyDirectoryToDirectory(f1, f2);
```

5. How will you handle windows based popup , other than robot class?

We can handle window based popup by using AutoIt

6. How will you scroll down and write a code?

Code:

```
 JavascriptExecutor j=(JavascriptExecutor) driver;
WebElement w = driver.findElement(By.xpath("//*[text()='Live Demo']"));
j.executeScript("arguments[0].scrollIntoView(true)", w);
```

7. Explain Robot class and write a code?

We can’t upload a file using selenium.
So we have one class is there for uploading file (i.e.) **Robot class**

```
public static void uploadFiles(File path) {
try {
Robot robot = new Robot();
robot.setAutoDelay(3000);
StringSelection selection = new StringSelection(
path.getAbsolutePath());
Toolkit.getDefaultToolkit().getSystemClipboard()
.setContents(selection, null);
// press ctrl+v
robot.keyPress(KeyEvent.VK_CONTROL);
robot.keyPress(KeyEvent.VK_V);
robot.setAutoDelay(3000);
// release ctrl+v
robot.keyRelease(KeyEvent.VK_CONTROL);
robot.keyRelease(KeyEvent.VK_V);
// press enter
robot.setAutoDelay(3000);
robot.keyPress(KeyEvent.VK_ENTER);
robot.keyRelease(KeyEvent.VK_ENTER);
} catch (AWTException e) {
e.printStackTrace();
}
}
```

Here ,

- Robot → class
- StringSelection → class
- Using this method we can upload any file.

8.Assume you launching IE browser you facing protected mode Exceptions, how you solve that?

Go to Internet option in IE →security→unchecked “**Enable Protected mode”**

9.WebDriver driver=new Firefoxdriver() Explain?

WebDriver –Interface.

FirefoxDriver-class.

Driver-Object.

10.FirefoxDriver is a class or interface, why you don't use interface here?

- FirefoxDriver is a class
- If we use interface we can't create object.

11.Write a code for web table to print all data?

```
public class Dummy {
public static void main(String[] args) {
System.setProperty("webdriver.gecko.driver",
"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
WebDriver driver = new FirefoxDriver();
driver.get("http://toolsqa.com/automation-practice-table/");
List<WebElement> tRows = driver.findElements(By.tagName("tr"));
for(WebElement rows:tRows){
List<WebElement> tData = driver.findElements(By.tagName("td"));
for(WebElement data:tData){
System.out.println(data.getText());
}
}
}
}
```

12.Write a code for dropdown?

```
public class Dummy {
        public static void main(String[] args) {
                System.setProperty("webdriver.gecko.driver",
                                "C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
                WebDriver driver = new FirefoxDriver();
                driver.get("http://ironspider.ca/forms/dropdowns.htm");
                WebElement w = driver.findElement(By.name("coffee"));
                Select s=new Select(w);
                s.selectByIndex(3);
        }
}
```

13.Write a code for maven dependency?

```
 <dependency>
        <groupId>org.seleniumhq.selenium</groupId>
        <artifactId>selenium-java</artifactId>
        <version>3.4.0</version>
</dependency>
```

14. Write a Junit dependency codefor maven?

```
<dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency
```

15.Write a code for multiple windows handling?

```
public class Dummy5 {
            public static void main(String[] args) throws InterruptedException {
                        System.setProperty("webdriver.gecko.driver",
"C:/Users/siva/workspace/Selenium/driver/geckodriver.exe");
                        WebDriver driver =new FirefoxDriver();
                        driver.get("https://www.hdfcbank.com/");
                        driver.findElement(By.xpath(".//*[@id='cee_closeBtn']/img")).click();
                        String p = driver.getWindowHandle();
                        System.out.println(p);
                        driver.findElement(By.xpath(".//*[@id='loginsubmit']")).click();
                        Set<String> all = driver.getWindowHandles();
                        for (String x : all) {
                                    System.out.println(x);
                                    if(!p.equals(x)){
                                                driver.switchTo().window(x);
                                                Thread.sleep(3000);

            driver.findElement(By.xpath("html/body/div[4]/div[2]/div[1]/a")).click();
                                                Thread.sleep(2000);
                                                driver.close();
                                    }
```

16.Uploading a file from local

We can’t upload a file using selenium.
So we have one class is there for uploading file (i.e.) **Robot class**

```
public static void uploadFiles(File path) {
                        try {
                                    Robot robot = new Robot();
                                    robot.setAutoDelay(3000);
                                    StringSelection selection = new StringSelection(
                                                            path.getAbsolutePath());
                                    Toolkit.getDefaultToolkit().getSystemClipboard()
                                                            .setContents(selection, null);
                                    // press ctrl+v
                                    robot.keyPress(KeyEvent.VK_CONTROL);
                                    robot.keyPress(KeyEvent.VK_V);
```

```
            robot.setAutoDelay(3000);
            // release ctrl+v
            robot.keyRelease(KeyEvent.VK_CONTROL);
            robot.keyRelease(KeyEvent.VK_V);
            // press enter
            robot.setAutoDelay(3000);
            robot.keyPress(KeyEvent.VK_ENTER);
            robot.keyRelease(KeyEvent.VK_ENTER);

        } catch (AWTException e) {

            e.printStackTrace();
        }

    }
```

17.Briefly explain pom?

POM:

- Page Object Model
- POM is an object repository design pattern in selenium webdriver
- POM creates our testing code maintainable and reusable
- Page factory is an optimized way to create object repository
- In POM have to create separate class for every web page

18.”System.out.println” explain?

- System → final class
- **out →static Object**
- **println() →method of out object**

19. Explain about maven life cycle?

The following lists all build phases of the default, clean and site lifecycles, which are executed in the order given up to the point of the one specified.

Clean Lifecycle

pre-clean	execute processes needed prior to the actual project cleaning
clean	remove all files generated by the previous build
post-clean	execute processes needed to finalize the project cleaning

Default Lifecycle

- **validate** - validate the project is correct and all necessary information is available
- **compile** - compile the source code of the project
- test - test the compiled source code using a suitable unit testing framework. These tests should not require the code be packaged or deployed
- package - take the compiled code and package it in its distributable format, such as a JAR.
- verify - run any checks on results of integration tests to ensure quality criteria are met
- install - install the package into the local repository, for use as a dependency in other projects locally
- deploy - done in the build environment, copies the final package to the remote repository for sharing with other developers and projects.

Site Lifecycle

pre-site	execute processes needed prior to the actual project site generation
site	generate the project's site documentation
post-site	execute processes needed to finalize the site generation, and to prepare for site deployment
site-deploy	deploy the generated site documentation to the specified web server

20. If u are joined newly here, what are all software and tools u need for automation

1. Eclipse
2. JDK
3. Selenium
4. Drivers for browsers
5. Git url
6. Maven

21. Automate gmail account check the number o messages in inbox

```
String inboxDetails = driver.findElement(By.id(“inbox“));

System.out.println(inboxDetails.getText());
```

22. **why we go for jenkins**

Jenkins triggers a build for every change made in the source code repository for example Git repository. Once the code is built it deploys it on the test server for testing. Concerned teams are constantly notified about build and test results. Finally, Jenkins deploys the build application on the production server.

23. **If I am having 100 test cases I want to execute 50 ly how do u do**

Eg: @Test(enable = false)

By using enable = false attribute, we can skip some test cases in TestNG

24. **How do u order the test cases**

Eg: @Test(priority = 1)

@Test(priority = 2)

By using priority attribute, we can order the test cases in TestNG

25. **Where do you keep your input data**

we keep our input data in excel

26. **Data Driven**

Data Driven automated testing is a method in which the test data set is created in the excel sheet, and is then imported into automation testing tools to provide to the software under test.

Sample code

```
// Import excel sheet.
File src=new File("C:\\Users\\Admin\\Desktop\\TestData.xls");
 // Load the file.
FileInputStream finput = new FileInputStream(src);
// Load the workbook.
XSSWorkbook wb = new XSSWorkbook (finput);
 // Load the sheet in which data is stored.
XSSFsheet sh1 = wb.getSheetAt(0);
System.out.println(sh1.getRow(0).getCell(0).getStringCellValue());
```

27. **When you use list and set in selenium**

While handling multiple windows I will use set
When am using findElements for dropdown, checkbox I will use return type List

28. **Agile methodology**

Backlog grooming
Srint Planning
Scrum meeting
Retrospective - should explain all steps

29. **Did u attend standup call what u discuss**

S I attend, we will discuss about like

1. What did yesterday
2. What have planned today
3. Is any obstacles in project

30. **Regression testing**

REGRESSION TESTING is to confirm that a recent code change has not affected existing features or ffunctionalities.

31. **Create issues in JIRA**

1. Title: Mention the title of an Application

2. Issue Type: Bug

3. Related To: Sprint 5

4. Description: Explain entire flow details

   Eg: I launched the browser

   I enter the URL and then

   I entered Username and password then

   I click the login button

   Expected: Need to navigate Home page

   Actual: Got the error

5. Priority: High

6. Severity: Major

7. Assigned To: Manager

SQL Questions

1. **Write sql query to find maximum 2nd salary of employee**

```
        SELECT MAX(EmpSalary)
        FROM Salary
        WHERE EmpSalary IN(SELECT TOP 2 EmpSalary FROM Salary ORDER BY EmpSalary ASC)
```

Programs

1. **Palindrome Program.**

```
public class PolyndromeNumberCheck {
        public static void main(String[] args) {

                int n, a, i = 0, j = 0;
                Scanner an = new Scanner(System.in);
                System.out.println("Enter a number");
                n = an.nextInt();
                a = n;
                while (a > 0) {
                        i = a % 10;
                        j = (j * 10) + i;
                        a = a / 10;

                }
                if (n == j) {
                        System.out.println("palindrome");
                } else {
                        System.out.println("Not palindrome Number");
                }
        }
}
```

Output:

```
Enter a number

141

Palindrome
```

2. **Assume a string “welcome to Polaris” remove space and print the string.**

```
public class Dummy {

        public static void main(String[] args) {
                String s="Welcome to Polaris";
                String x = s.replace(" ", "");
                System.out.println(x);
        }
  }
```

Output:

```
WelcometoPolaris
```

3. **Write a program to split and then reverse a string.**

Reverse the string:

```
 public class ReverseString {
  public static void main(String args[]) {
             String original, reverse = "";
             Scanner in = new Scanner(System.in);

             System.out.println("Enter a string to reverse");
             original = in.nextLine();

             int length = original.length();

             for (int i = length - 1; i >= 0; i--)
                     reverse = reverse + original.charAt(i);

             System.out.println("Reverse of entered string is: " + reverse);
  }

}
```

Split:

```
public class StringBasic {
        public static void main(String[] args) {
                String s1 = "Hello welcome to java class";
                String[] x = s1.split(" "); // here we split by space
                for(int i=0;i<x.length;i++){
                        System.out.println(x[i]);
                }
        }
}
```

<u>Output:</u>

```
Hello

welcome

to

java

class
```

4.Fibonacci series program.

```
public class Fibonacci {
        public static void main(String[] args) {
                int a = 0, b = 1;
                System.out.println(a);
                System.out.println(b);
                for (int i = 2; i <= 10; i++) {
                        int c = a + b;
                        System.out.println(c);
                        a = b;
                        b = c;
                }}}
```

Output:

```
0

1

1

2

3

5

8

13

21

34
```

5.Write a sample program for list and set?

List:

```
public class ArList {
        public static void main(String[] args) {
                List<Integer> ex=new ArrayList<Integer>();
                ex.add(10);
                ex.add(20);
                ex.add(30);
                ex.add(40);
                ex.add(40);
                ex.add(50);
                System.out.println(ex);
        }
}
```

Output:

```
[10, 20, 30, 40, 40, 50]
```

Set:

```
public class ArList {
        public static void main(String[] args) {
                Set<Integer> ex = new HashSet<Integer>();
                ex.add(10);
                ex.add(20);
                ex.add(30);
                ex.add(40);
                ex.add(50);
                ex.add(50);
                System.out.println(ex);
        }
}
```

Output:

```
[50, 20, 40, 10, 30]
```

6. String reverse with split

```
String original, reverse = "";
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the Original String");
                original = sc.nextLine();
                int length = original.length();
                for (int i = length - 1; i >= 0; i--) {
                        reverse = reverse + original.charAt(i);

                        System.out.println("Reverse of entered string is: " + reverse);
                }

                String[] words = reverse.split("\\s");
                for (String w : words) {
                        System.out.println(w);

                }
```