Level1

What is the difference between the throw and throws keywords in Java?

**Java throw keyword is used throw an exception explicitly in the code, inside the function or the block of code.keyword is followed by an instance of Exception to be thrown,allowed to throw only one exception at a time.*

** Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code. throws is followed by exception class name.throws can declare multiple exceptions*

What is the purpose of the finalize() method in Java? Why is it considered a bad practice to use it?

** The primary purpose of the finalize() method is to allow an object to clean up after itself before it is collected by the garbage collector, such as closing open resources like database connections, network connections, or file descriptors.*

** bad practice to use because it doesn't have any guarantee of execution and adds heavy penalty on performance.*

What is the difference between an abstract class and an interface in Java? When would you use one over the other?

** An abstract class in Java cannot be instantiated and can contain both abstract and non-abstract methods. It can also contain member variables and constructors, which interfaces cannot.*

** An interface defines a set of methods and properties that must be implemented by any class or structure that implements the interface.*

** Abstract classes can have constructors but interfaces can't have constructors.*

** Abstract classes can extend other class and implement interfaces but interface can only extend other interfaces.*

- When there are a lot of methods in the contract, abstract class is more useful.

- Interfaces are a good choice for providing the base for class hierarchy and contract.

interfaces is one of the best practices for coding in java.

- Depends on the specific design needs of the application. Interfaces are generally more flexible and are a good choice for defining a contract that can be implemented by any class. Abstract classes, on the other hand, are useful when you want to provide common behavior or maintain state across subclasses.

What is a Lambda expression in Java? Provide an example to explain its use.

** Lambda expression is important feature in java. It provides a clear and concise way to represent one method interface using an expression. It doesn't have a name, but it has a list of parameters, a body, a return type, and also possibly a list of exceptions that can be thrown.*

** It is very useful in collection library. It helps to iterate, filter and extract data from collection.*

** It saves a lot of code and you don't need to define the method again for providing the implementation.*

What is the difference between the continue and break statements in Java?

** The continue statement, on the other hand, skips the current iteration of a loop and jumps to the next iteration immediately and It does not terminate the loop.*

** The break statement is used to terminate the loop or switch-case statement it is currently in.*

What is the purpose of the volatile keyword in Java? Provide an example to explain its use.

** Volatile keyword is used to modify the value of a variable by different threads. It is also used to make classes thread safe. It means that multiple threads can use a method and instance of the classes at the same time without any problem. The volatile keyword can be used either with primitive type or objects. The volatile keyword cannot be used with classes or methods.*

Example:public class MyRunnable implements Runnable {

```
  private volatile boolean active;
  public void run() {
    active = true;
    while (active) {
    }
  }
  public void stop() {
    active = false;
  }
}
```

What is the difference between ArrayList and LinkedList in Java? When would you use one over the other?

** ArrayList internally uses a dynamic array to store the elements.Manipulation with ArrayList is slow because it internally uses an array. is better for storing and accessing data.*

**LinkedList internally uses a doubly linked list to store the elements. Manipulation with LinkedList is faster. better for manipulating data.*

- ArrayList good when You need fast random access and retrieval of elements by index.

- LinkedList usefull when You need frequent insertions and deletions in the middle of the list.

What is the difference between public, protected, and private access modifiers in Java?

** Public: The access level of a public modifier is everywhere. It can be accessed from within the class, outside the class, within the package and outside the package.*

** Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.*

** Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.*

What is an exception in Java? Provide an example to explain the concept.

** In Java, an exception is an event that disrupts the normal flow of the program. It is an object which is thrown at runtime.*

```
public class ExceptionExample  {
   public static void main (String[] args) {
      int a=5;
      int b=0;
      try{
          System.out.println(a/b);
      }
      catch(ArithmeticException e){
         System.out.println(e.getMessage());
      }
   }
}
```

What is the purpose of the static keyword in Java? Provide an example to explain its use.

** The static keyword in Java is mainly used for memory management and is applied to variables, methods, blocks, and nested classes. It indicates that the member belongs to the class it self rather than to an instance of the class. This means that there is only one copy of that member for all instances of the class .*

```
class Test {
      static void m1() {
         System.out.println("from m1")
}
      public static void main(String[] args) {
         m1();
      }
   }
```

What is a constructor in Java? Provide an example to explain its use.

** A constructor in Java is a special method within a class that is used to initialize objects of that class. It has the same name as the class and it doesn't have a return type. The primary purpose of a constructor is to set default values to the object attributes or properties*

- Example :

```
public class Main {
  int x;
  public Main() {
    x = 5;  //   }
  public static void main(String[] args) {
    Main myObj = new Main();     System.out.println(myObj.x);
```

What is the purpose of the interface keyword in Java? Provide an example to explain its use.

** the interface keyword is used to declare an interface and to achieve abstraction, multiple inheritances in Java. similarly is used to declare a special type of class that only contains abstract methods.*

- Example :

```
interface Animal {
   void eat();
}
class Cat implements Animal {    public void eat() {
      System.out.println("The cat eats");
   }
public class Test {
   public static void main(String[] args) {
      Cat cat = new Cat();
      cat.eat();
```

What is a thread in Java? Provide an example to explain the concept.

** Thread in java is the smallest part of the process that allows a program to operate more efficiently by running multiple tasks simultaneously.*

- Example With threads, you can create two separate threads, one for downloading and one for processing. Each thread operates independently and concurrently. This means that while one thread is downloading the file, the other can start processing the data, and they can run simultaneously.

What is the purpose of the super keyword in Java? Provide an example to explain its use.

** super keyword can be used to refer immediate parent class instance variable.can be used to invoke immediate parent class method and invoke immediate parent class constructor.*

in other hand is used to refer to the superclass (parent class) of a subclass (child class). It is primarily used to access members (fields or method once and use it many times. We do not require to write code again and again. It also provides the easy modification and readability of code.

- Example : name of the method is addNumbers.

```
public int addNumbers(int x, int y) {
 int z = 0;
 z = x + y;
 System.out.println(z);
 return z;
```

What is the difference between a HashMap and a TreeMap in Java? When would you use one over the other?

** A TreeMap can save memory in comparison to a HashMap because it only uses the amount of memory needed to hold its items*

** HashMap is faster than TreeMap. HashMap allows a single null key but TreeMap does not.*

** HashMap uses equals() method of the object class to compare keys and TreeMap uses the compareTo() method to compare keys.*

- Use a TreeMap if you want to keep your entries sorted. TreeMap is a good choice when you need to frequently access elements in a sorted manner.

- Use a HashMap if you prioritize performance over memory consumption. HashMap is a better choice when you need to frequently add, remove, or access elements, and the order of elements is not important.

What is the purpose of the assert keyword in Java? Provide an *example to explain its use.*

** The keyword assert makes the code more readable and helps in code optimization. assert usedto test the assumptions made in the program and verify certain conditions or states.*

- Example : public class Example {

```
public static void main(String[] args)

    int age = 15;

    assert age <= 18 : " you cannot have a license ";

    System.out.println(" Age of the person is " + age);
```

What is the difference between final, finally, and finalize keywords in Java?

** final is used to restrict access and modification of classes, methods, and variables, finally is used in exception handling to ensure certain code is executed regardless of whether an exception is thrown, and finalize is a method called by the garbage collector to perform cleanup activities before an object is destroyed.*

What is polymorphism in Java? Explain with an example.

** Polymorphism in Java is a fundamental concept in Object-Oriented Programming (OOP) that allows a single action to be performed in multiple ways. The term "polymorphism" is derived from two Greek words: "poly", which means many, and "morphs", which means forms.There are two types of polymorphism in Java: compile-time polymorphism(Static) and runtime polymorphism(dynamic).*

- Example : class Helper {

```
  static int Multiply(int a, int b) {
    return a * b;
  }
  static int Multiply(int a, int b, int c) {
    return a * b * c;
  }
}
```

10 Programming Questions

Write a program to implement a binary search tree in Java.

*BinarySearchTree.java

```java
package com.bootcamp.level1;
   // program 1
class Node {
    int key;
    Node left, right;
    public Node(int item) {
        key = item;
        left = right = null;
    }
}
public class BinarySearchTree {
    private Node root;
    public BinarySearchTree() {
        root = null;
    }
    public void insert(int key) {
        root = insertRec(root, key);
    }
    private Node insertRec(Node root, int key) {
        if (root == null) {
            root = new Node(key);
            return root;
        }
        if (key < root.key) {
            root.left = insertRec(root.left, key);
        } else if (key > root.key) {
            root.right = insertRec(root.right, key);
        }
        return root;
    }
    public void delete(int key) {
        root = deleteRec(root, key);
    }
    private Node deleteRec(Node root, int key) {
        if (root == null) {
            return root;
        }
        if (key < root.key) {
            root.left = deleteRec(root.left, key);
        } else if (key > root.key) {
            root.right = deleteRec(root.right, key);
        } else {
            if (root.left == null) {
```

```
            if (root.left == null) {
                return root.right;
            } else if (root.right == null) {
                return root.left;
            }
            root.key = minValue(root.right);
            root.right = deleteRec(root.right, root.key);
        }
        return root;
    }
    private int minValue(Node root) {
        int minValue = root.key;
        while (root.left != null) {
            minValue = root.left.key;
            root = root.left;
        }
        return minValue;
    }
    public boolean search(int key) {
        return searchRec(root, key);
    }
    private boolean searchRec(Node root, int key) {
        if (root == null) {
            return false;
        }
        if (key == root.key) {
            return true;
        } else if (key < root.key) {
            return searchRec(root.left, key);
        } else {
            return searchRec(root.right, key);
        }
    }
    public static void main(String[] args) {
        BinarySearchTree tree = new BinarySearchTree();
        tree.insert(45);
        tree.insert(35);
        tree.insert(25);
        tree.insert(15);
        tree.insert(75);
        tree.insert(65);
        tree.insert(85);
```

```
86          System.out.println("Inorder traversal of the tree:");
87          tree.inorder();
88          System.out.println("Search for 15: " + tree.search(15));
89          tree.delete(30);
90          System.out.println("Inorder traversal after deleting 25:");
91          tree.inorder();
92      }
93      public void inorder() {
94          inorderRec(root);
95      }
96      private void inorderRec(Node root) {
97          if (root != null) {
98              inorderRec(root.left);
99              System.out.print(root.key + " ");
100             inorderRec(root.right);
101         }
102     }
103 }
```

Write a program to implement a doubly-linked list in Java.

*DoublyLinkedList.java

```java
package com.bootcamp.level1;
  // program2
class Node {
    int data;
    Node prev;
    Node next;
    public Node(int data) {
        this.data = data;
        this.prev = null;
        this.next = null;
    }
}
public class DoublyLinkedList {
    private Node head;
    private Node tail;
    public DoublyLinkedList() {
        this.head = null;
        this.tail = null;
    }
    public boolean isEmpty() {
        return head == null;
    }
    public void insertAtEnd(int data) {
        Node newNode = new Node(data);
        if (isEmpty()) {
            head = newNode;
            tail = newNode;
        } else {
            tail.next = newNode;
            newNode.prev = tail;
            tail = newNode;
        }
    }

    public void delete(int data) {
        Node current = head;
        while (current != null) {
            if (current.data == data) {
                if (current.prev != null) {
                    current.prev.next = current.next;
                } else {
                    head = current.next;
                }
```

Write a program to sort an array of objects in Java using the quicksort algorithm.

*QuickSortExample.java

```java
package com.bootcamp.level1;
import java.util.Arrays;
// program3
class Person implements Comparable<Person> {
    private String name;
    private int age;
 Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    @Override
    public int compareTo(Person otherPerson) {
        return this.age - otherPerson.age;
    }
    @Override
    public String toString() {
        return name + " (age: " + age + ")";
    }
}
public class QuickSortExample {
    public static void quickSort(Comparable[] arr, int low, int high) {
        if (low < high) {
            int partitionIndex = partition(arr, low, high);

            quickSort(arr, low, partitionIndex - 1);
            quickSort(arr, partitionIndex + 1, high);
        }
    }
    private static int partition(Comparable[] arr, int low, int high) {
        Comparable pivot = arr[high];
        int i = low - 1;
        for (int j = low; j < high; j++) {
            if (arr[j].compareTo(pivot) < 0) {
                i++;
                swap(arr, i, j);
            }
        }
        swap(arr, i + 1, high);
        return i + 1;
    }
    private static void swap(Comparable[] arr, int i, int j) {
        Comparable temp = arr[i];
```

```
44          arr[i] = arr[j];
45          arr[j] = temp;
46      }
47      public static void main(String[] args) {
48          Person[] people = {
49              new Person("Alice", 30),
50              new Person("Bob", 25),
51              new Person("Charlie", 35),
52              new Person("David", 28),
53              new Person("Eve", 22)
54          };
55          System.out.println("Unsorted Array:");
56          System.out.println(Arrays.toString(people));
57          quickSort(people, 0, people.length - 1);
58          System.out.println("Sorted Array (by age):");
59          System.out.println(Arrays.toString(people));
60      }
61  }
```

Write a program to implement a stack using a linked list in Java.

*Node.java

```java
package com.bootcamp.level1;
// program4
public class Node {
    int data;
    Node next;

    Node(int data) {
        this.data = data;
        this.next = null;
    }
}

public class Stack {
    Node head;
    boolean isempty() {
        return head == null;
    }
    void push(int data) {
        Node newnode = new Node(data);
        newnode.next = head;
        head = newnode;
    }
    int pop() {
        if (isempty()) {
            return -1;
        }
        int poppeddata = head.data;
        head = head.next;
        return poppeddata;
    }
    int peek() {
        if (isempty()) {
            return -1;
        }
        return head.data;
    }
}
```

Write a program to implement a queue using two stacks in Java.

*LCS.java

```java
package com.bootcamp.level1;
 // program 6
public class LCS {
    static void lcs(String S1, String S2, int m, int n) {
        int[][] LCS_table = new int[m + 1][n + 1];

        for (int i = 0; i <= m; i++) {
            for (int j = 0; j <= n; j++) {
                if (i == 0 || j == 0)
                    LCS_table[i][j] = 0;
                else if (S1.charAt(i - 1) == S2.charAt(j - 1))
                    LCS_table[i][j] = LCS_table[i - 1][j - 1] + 1;
                else
                    LCS_table[i][j] = Math.max(LCS_table[i - 1][j], LCS_table[i][j - 1]);
            }
        }

        System.out.println("Length of LCS is " + LCS_table[m][n]);
    }

    public static void main(String[] args) {
        String S1 = "AGGTAB";
        String S2 = "GXTXAYB";
        int m = S1.length();
        int n = S2.length();
        lcs(S1, S2, m, n);
    }
}
```

Write a program to find the maximum subarray sum in an array of integers in Java.

*MaximumSubarraySum.java

```java
package com.bootcamp.level1;
 // program 7
public class MaximumSubarraySum {
    public static int findMaxSubarraySum(int[] nums) {
        int maxSum = Integer.MIN_VALUE;
        int currentSum = 0;
        for (int num : nums) {
            currentSum = Math.max(num, currentSum + num);
            maxSum = Math.max(maxSum, currentSum);
        }
        return maxSum;
    }
    public static void main(String[] args) {
        int[] arr = {-2, 1, -3, 4, -1, 2, 1, -5, 4};
        int maxSubarraySum = findMaxSubarraySum(arr);
        System.out.println("Maximum Subarray Sum: " + maxSubarraySum);
    }
}
```

Write a program to implement a merge sort algorithm in Java.

MergeSort.java

```java
package com.bootcamp.level1;
 // program 8
public class MergeSort {
    public static void mergeSort(int[] array) {
        if (array == null) {
            return;
        }
        int length = array.length;
        if (length <= 1) {
            return;
        }
        int mid = length / 2;
        int[] left = new int[mid];
        int[] right = new int[length - mid];
        for (int i = 0; i < mid; i++) {
            left[i] = array[i];
        }
        for (int i = mid; i < length; i++) {
            right[i - mid] = array[i];
        }
        mergeSort(left);
        mergeSort(right);
        merge(array, left, right);
    }
    private static void merge(int[] array, int[] left, int[] right) {
        int leftLength = left.length;
        int rightLength = right.length;
        int i = 0, j = 0, k = 0;

        while (i < leftLength && j < rightLength) {
            if (left[i] <= right[j]) {
                array[k] = left[i];
                i++;
            } else {
                array[k] = right[j];
                j++;
            }
            k++;
        }
        while (i < leftLength) {
            array[k] = left[i];
            i++;
            k++;
```

Write a program to implement a binary search algorithm for a rotated sorted array in Java.

RotatedBinarySearch.java

```java
package com.bootcamp.level1;
 // program 9
public class RotatedBinarySearch {
    public static int search(int[] nums, int target) {
        int left = 0;
        int right = nums.length - 1;

        while (left <= right) {
            int mid = left + (right - left) / 2;
            if (nums[mid] == target) {
                return mid;
            }
            if (nums[left] <= nums[mid]) {

                if (nums[left] <= target && target < nums[mid]) {
                    right = mid - 1;
                } else {
                    left = mid + 1;
                }
            } else {
                if (nums[mid] < target && target <= nums[right]) {
                    left = mid + 1;
                } else {
                    right = mid - 1;
                }
            }
        }
        return -1;
    }
    public static void main(String[] args) {
        int[] rotatedArray = {4, 5, 6, 7, 0, 1, 2};
        int target = 0;
        int result = search(rotatedArray, target);

        if (result != -1) {
            System.out.println("Target " + target + " found at index " + result);
        } else {
            System.out.println("Target " + target + " not found in the array");
        }
    }
}
```

Write a program to find the shortest path in a weighted graph using Dijkstra's algorithm in Java.

*DijkstraShortestPath.java

```java
package com.bootcamp.level1;
 // program 10
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
     // program10
public class DijkstraShortestPath {
    private int V;
    private List<List<Node>> adj;
    private int[] dist;
    private boolean[] visited;
    class Node {
        int dest;
        int weight;
        Node(int dest, int weight) {
            this.dest = dest;
            this.weight = weight;
        }
    }
    public DijkstraShortestPath(int V) {
        this.V = V;
        adj = new ArrayList<>(V);
        for (int i = 0; i < V; i++) {
            adj.add(new ArrayList<>());
        }
    }
    public void addEdge(int src, int dest, int weight) {
        Node node = new Node(dest, weight);
        adj.get(src).add(node);
    }
    public void findShortestPath(int source) {
        dist = new int[V];
        visited = new boolean[V];
        Arrays.fill(dist, Integer.MAX_VALUE);
        dist[source] = 0;
        for (int count = 0; count < V - 1; count++) {
            int u = minDistance();
            visited[u] = true;
            for (Node node : adj.get(u)) {
                int v = node.dest;
                int weight = node.weight;
                if (!visited[v] && dist[u] != Integer.MAX_VALUE && dist[u] + weight < dist[v]) {
                    dist[v] = dist[u] + weight;
```

```
                }
            }
        }

        printShortestPaths(source);
    }
    private int minDistance() {
        int minDist = Integer.MAX_VALUE;
        int minIndex = -1;
        for (int v = 0; v < V; v++) {
            if (!visited[v] && dist[v] <= minDist) {
                minDist = dist[v];
                minIndex = v;
            }
        }
        return minIndex;
    }
    private void printShortestPaths(int source) {
        System.out.println("Shortest distances from source " + source + " to all other nodes:");
        for (int i = 0; i < V; i++) {
            System.out.println("Node " + i + ": Distance = " + dist[i]);
        }
    }
    public static void main(String[] args) {
        int V = 5;
        DijkstraShortestPath graph = new DijkstraShortestPath(V);
        graph.addEdge(0, 1, 2);
        graph.addEdge(0, 3, 5);
        graph.addEdge(1, 2, 3);
        graph.addEdge(1, 3, 2);
        graph.addEdge(2, 4, 4);
        graph.addEdge(3, 4, 1);
        int source = 0;
        graph.findShortestPath(source);
    }
}
```

3 Selenium Framework Questions:

Create a Page Object Model for the login page of a website. Use Page Factory to initialize the elements. Write a TestNG test to verify successful login with valid credentials.

```java
package com.selenium;

import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

public class addProductToCart {

    public static void main(String[] args) {

        WebDriver driver = new WebDriver();
        driver.get("https://www.amazon.in/");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.findElement(By.xpath("//input[@id=\"twotabsearchtextbox\"]")).sendkeys("samsung s22");
        driver.findElement(By.xpath("//input[@id=\"nav-search-submit-button\"]")).click();
        driver.findElement(By.xpath("//span[@class=\"a-size-medium a-color-base a-text-normal\"]")).click();


        Set<String> s = driver.getWindowHandles();
        ArrayList<E> ar = new ArrayList(s);
        System.out.println();(ar.get(0));
        System.out.println();(ar.get(1));

        driver.switchTo().window.((String)ar.get(1));

        driver.findElement(By.xpath("//input[@id=\"add-to-cart-button\"]")).click();

    }
}
```

Create a Page Object Model for the search page of a website. Use Page Factory to initialize the elements. Write a TestNG test to search for a product, verify that the search results are displayed, and click on a specific product to navigate to its details page.

*verifyCertainElements.java

```java
package com.selenium;

import java.time.Duration;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;
import org.openqa.selenium.chrome.ChromeOptions;
import org.testng.annotations.AfterMethod;
import org.testng.annotations.BeforeMethod;
import org.testng.annotations.Test;
import org.testng.asserts.SoftAssert;

Run All
public class verifyCertainElements {
    WebDriver driver;
    SoftAssert softassert = new SoftAssert();
    public ChromeOptions options;
    @BeforeMethod
    public void setup() {
        driver= new ChromeDriver();
        driver.manage().deleteAllCookies();
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));
        driver.get("https://saucedemo.com");
        System.out.println(driver.getTitle());
    }

    @Test(priority = 1)
    Run | Debug
    public void veriyLoginWithValidCredintial() {

        driver.findElement(By.cssSelector("input#user-name")).sendKeys("standard_user");
        driver.findElement(By.cssSelector("input#password")).sendKeys("secret_sauce");
        driver.findElement(By.cssSelector("input#login-button")).click();
        softassert .assertTrue(driver.findElement(By.cssSelector("span.title")).isDisplayed());
        softassert.assertAll();
        driver.findElement(By.xpath("//div[text()='Sauce Labs Backpack']")).click();

       String actualtextbox = driver.findElement(By.cssSelector("button#add-to-cart-sauce-labs-backpack")).getText();
       String expectedtextbox = "Add to cart";
       softassert.assertTrue(actualtextbox.equals(expectedtextbox));
       if (actualtextbox.equals(expectedtextbox)) {
            System.out.println("page element exists");

        }else

        System.out.println("page element don't exists");
    }
    @AfterMethod
    public void tearDown() {
    }
}
```

Level2

What is the difference between an abstract class and an interface in Java?

** An abstract class is a class that cannot be instantiated and can contain both abstract and non-abstract methods. An interface, on the other hand, is a contract that specifies a set of methods that a class must implement. All methods in an interface are by default abstract.*

** abstract classes are used to provide a base class for concrete subclasses to inherit from, while interfaces are used to define a set of methods that a class must implement. Abstract classes can have implemented and abstract methods, while interfaces can only have abstract methods. Classes can inherit from only one abstract class, but can implement multiple interfaces.*

What is the purpose of the static keyword in Java?

** The static keyword in Java is mainly used for memory management and is applied to variables, methods, blocks, and nested classes. It indicates that the member belongs to the class it self rather than to an instance of the class. This means that there is only one copy of that member for all instances of the class .*

What is the purpose of the final keyword in Java?

** The final keyword in Java is used to restrict the user. It can be applied to variables, methods, and classes, and it indicates that the variable, method, or class cannot be modified or extended and its value cannot be changed once it has been initialized.*

What is the difference between a checked and an unchecked exception in Java?

** Checked exceptions are checked at compile-time, while unchecked exceptions are checked at runtime.*

Checked exceptions represent predictable, external erroneous conditions, while unchecked exceptions represent programming errors.

Checked exceptions must be declared in the method signature or handled with a try-catch block, while unchecked exceptions do not need to be declared or handled.

What is the difference between a stack and a queue data structure in Java?

- Stack follows the LIFO (Last-In-First-Out) principle, while Queue follows the FIFO (First-In-First-Out) principle.

- In a Stack, elements are added and removed from the same end, while in a Queue, elements are added at one end and removed from the other end.

- Stack provides the push() method to add an element and the pop() method to remove an element. Queue provides the add() method to add an element and the remove() method to remove an element

that multiple threads can use a method and instance of the classes at the same time without any problem.

What is the purpose of the transient keyword in Java?

** The transient keyword can be used with the data members of a class in order to avoid their serialization. For example, if a program accepts a user's login details and password. But we don't want to store the original password in the file. Here, we can use transient keyword and when JVM reads the transient keyword it ignores the original value of the object and instead stores the default value of the object.*

What is the difference between the equals() and hashCode() methods in Java?

*e variables. * To Call Another Constructor.*

** To Create and Return the Current Object. * To pass as an argument in the method.*

** To pass the current class instance as a parameter.*

What is the difference between a superclass and a subclass in Java?

** In Java, a superclass is a class from which other classes are derived. It is also known as a base class or parent class. The class that is derived from the superclass is known as a subclass or child class. The subclass inherits all the public and protected members of its parent, no matter what package the subclass is in. If the subclass is in the same package as its parent, it also inherits the package-private members of the parent.*

What is the purpose of the package keyword in Java?

** Preventing naming conflicts. * Making searching/locating and usage of classes, interfaces, enumerations, and annotations easier. * Providing controlled access to classes and their members. * Data encapsulation which allow to group related classes together.*

What is the difference between an instance variable and class variable in java?

** The difference between instance variables and class variables is that instance variables belong to the instance of a class and every instance of that class has its own copy of that variable. On the other hand, class variables are shared among all instances of the class. Therefore, changes to a class variable will affect all instances of the class, while changes to an instance variable will only affect the instance where the change was made.*

What is the purpose of the instanceof operator in Java?

* The instanceof operator in Java is used to check if an object is an instance of a specific class, subc lass, or interface. It's a binary operator that compares an instance with a type and returns either true or false. in other hand is used when working with polymorphism, handling different object types, or responding to objects of different classes or interfaces in a flexible and controlled manner.

What is the purpose of the super keyword in Java?

** super keyword can be used to refer immediate parent class instance variable.can be used to invoke immediate parent class method and invoke immediate parent class constructor.*

in other hand is used to refer to the superclass (parent class) of a subclass (child class). It is primarily used to access members (fields or methods) of the superclass, resolve name conflicts between superclass and subclass.

What is the purpose of the abstract keyword in Java?

** The abstract keyword in Java is used to declare abstract classes and abstract methods or to achieve abstraction, which is one of the pillars of Object-Oriented Programming (OOP). It is a non-access modifier that can be used for classes and methods, but not for variables.*

What is the purpose of the interface keyword in Java?

** The interface keyword in Java is used to declare an interface. It provides total abstraction, meaning all the methods in an interface are declared with an empty body, and all the fields are public, static, and final by default. in other hand The primary purpose of the interface keyword is to declare a contract or a set of methods that classes must adhere to, without providing any implementation details.*

What is the difference between a private and a protected access modifier in Java?

** Private: The access level of a private modifier is only within the class. It cannot be accessed from outside the class.*

** Protected: The access level of a protected modifier is within the package and outside the package through child class. If you do not make the child class, it cannot be accessed from outside the package.*

What is the purpose of the throws keyword in Java?

** Java throws keyword is used in the method signature to declare an exception which might be thrown by the function while the execution of the code. throws is followed by exception class name.throws can declare multiple exceptions.*

10 Java Programming Questions:

Write a Java program to find the factorial of a number using recursion.

*program1.java

```java
package com.bootcamp.level2;

public class program1 {
        public static int factorial(int n) {
            if (n == 0) {
                return 1;
            } else {
                return n * factorial(n - 1);
            }
        }

        public static void main(String[] args) {
            int num = 5;  // Number for which factorial is to be found
            int result = factorial(num);
            System.out.println("The factorial of " + num + " is " + result);
        }
    }
```

Write a Java program to check if a given number is a palindrome or not.

program2.java

```java
package com.bootcamp.level2;

public class program2 {
        public static void main(String[] args) {
            int num = 12321, reversedNum = 0, remainder;
            int originalNum = num;
            while (num != 0) {
                remainder = num % 10;
                reversedNum = reversedNum * 10 + remainder;
                num /= 10;
            }
            if (originalNum == reversedNum) {
                System.out.println(originalNum + " is a Palindrome.");
            } else {
                System.out.println(originalNum + " is not a Palindrome.");
            }
        }
    }
```

Write a Java program to check if a given string is a palindrome or not.

Write a Java program to reverse a given string without using any built-in functions.

Write a Java program to find the second highest number in an array.

program5.java

```java
package com.bootcamp.level2;
// SecondHighestInArray
public class program5{
    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5, 6, 7, 8, 9};
        int highest = Integer.MIN_VALUE;
        int secondHighest = Integer.MIN_VALUE;

        for (int i = 0; i < arr.length; i++) {
            if (arr[i] > highest) {
                secondHighest = highest;
                highest = arr[i];
            } else if (arr[i] > secondHighest && arr[i] < highest) {
                secondHighest = arr[i];
            }
        }

        System.out.println("The second highest number in the array is: " + secondHighest);
    }
}
```

Write a Java program to implement bubble sort algorithm.

Write a Java program to implement selection sort algorithm.

SelectionSort.java

```java
package com.bootcamp.level2;
 // program 7
  public class SelectionSort{
void sort(int arr[]) {
    int n = arr.length;

    for (int i = 0; i < n - 1; i++) {
        int min_idx = i;
        for (int j = i + 1; j < n; j++)
            if (arr[j] < arr[min_idx])
                min_idx = j;

        int temp = arr[min_idx];
        arr[min_idx] = arr[i];
        arr[i] = temp;
    }
}

void printArray(int arr[]) {
    int n = arr.length;
    for (int i = 0; i < n; ++i)
        System.out.print(arr[i] + " ");
    System.out.println();
}

public static void main(String args[]) {
    SelectionSort ob = new SelectionSort();
    int arr[] = {62, 23, 10, 20, 9,43};
    ob.sort(arr);
    System.out.println("Sorted array");
    ob.printArray(arr);
}
}
```

Write a Java program to implement insertion sort algorithm.

InsertionSort.java

```java
package com.bootcamp.level2;
// program 8
public class InsertionSort {
    void sortArray(int arr[]) {
        int n = arr.length;
        for (int i = 1; i < n; i++) {
            int key = arr[i];
            int j = i - 1;

            while (j >= 0 && arr[j] > key) {
                arr[j + 1] = arr[j];
                j = j - 1;
            }
            arr[j + 1] = key;
        }
    }

    void printArray(int arr[]) {
        int n = arr.length;
        for (int i = 0; i < n; ++i)
            System.out.print(arr[i] + " ");
        System.out.println();
    }

    public static void main(String args[]) {
        InsertionSort ob = new InsertionSort();
        int arr[] = {17, 3, 23, 9, 14};
        ob.sortArray(arr);
        System.out.println("Sorted array");
        ob.printArray(arr);
    }
}
```

Write a Java program to implement binary search algorithm.

BinarySearch.java

```java
package com.bootcamp.level2;
 // program 9
public class BinarySearch {
    int binarySearch(int arr[], int l, int r, int x) {
        while (l <= r) {
            int mid = l + (r - l) / 2;

            if (arr[mid] == x)
                return mid;

            if (arr[mid] > x)
                r = mid - 1;
            else
                l = mid + 1;
        }

        return -1;
    }

    public static void main(String args[]) {
        BinarySearch bs = new BinarySearch();
        int arr[] = {12, 17, 29, 23, 34,49};
        int n = arr.length;
        int x = 29;
        int result = bs.binarySearch(arr, 0, n - 1, x);
        if (result == -1)
            System.out.println("Element not present");
        else
            System.out.println("Element found at index " + result);
    }
}
```

Write a Java program to count the number of words in a given string.

*wordCount.java

```java
package com.bootcamp.level2;
 // program 10
public class wordCount {
    public static void main(String[] args) {
        String inputString = "Java is my favorite programming language.";

        inputString = inputString.trim();

        String[] words = inputString.split("\\s+");

        int wordCount = words.length;
        System.out.println("Number of words: " + wordCount);
    }
}
```

3 Selenium Framework Questions:

Create a Page Object Model for the registration page of a website. Use Page Factory to initialize the elements. Write a TestNG test to verify successful registration with valid credentials.

Create a Page Object Model for the checkout page of a website. Use Page Factory to initialize the elements. Write a TestNG test to add a product to the cart, navigate to the checkout page, and verify that the correct product is being purchased.

```java
package com.selenium;

import java.util.ArrayList;
import java.util.concurrent.TimeUnit;

public class addProductToCart {

    public static void main(String[] args) {

        WebDriver driver = new WebDriver();
        driver.get("https://www.amazon.in/");
        driver.manage().window().maximize();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);

        driver.findElement(By.xpath("//input[@id=\"twotabsearchtextbox\"]")).sendkeys("samsung s22");
        driver.findElement(By.xpath("//input[@id=\"nav-search-submit-button\"]")).click();
        driver.findElement(By.xpath("//span[@class=\"a-size-medium a-color-base a-text-normal\"]")).click();


        Set<String> s = driver.getWindowHandles();
        ArrayList<E> ar = new ArrayList(s);
        System.out.println();(ar.get(0));
        System.out.println();(ar.get(1));

        driver.switchTo().window.((String)ar.get(1));

        driver.findElement(By.xpath("//input[@id=\"add-to-cart-button\"]")).click();

    }
}
```

```java
                Assert.assertTrue(ActualDisplayMessage.contains(expectedDisplayMessage));
            }

            @AfterMethod
            public void tearDown () {
             driver.quit();
            }
        }
```

Level3

What is the purpose of the public keyword in Java?

** The public keyword in Java is an access modifier used for classes, attributes, methods, and constructors, making them accessible by any other class* from any part of your program, whether within the same package or from other packages *It is the most non-restricted type of access modifier*.

What is the purpose of the private keyword in Java?

** private keyword is mostly used to create a fully encapsulated class in Java by making all the data members of that class private which allows you to hide the internal implementation details of a class.*

** Hiding and protect the data within the object from direct external manipulation.*

** Preventing Unauthorized Access of other classes or objects or modifying the private members directly which ensure data integrity and maintain the expected behavior of the class.*

What is the purpose of the protected keyword in Java?

* The protected keyword is useful when you want to expose class members to a subclass or other classes in the same package while still maintaining some level of encapsulation and maintain good coding practices. Avoid exposing sensitive data through protected members, and use descriptive names for protected members to make it easier for other developers to understand their purpose.

What is the purpose of the static keyword in Java?

** The static keyword in Java is mainly used for memory management. The static keyword in Java is used to share the same variable or method of a given class. The static keyword is used for a constant variable or a method that is the same for every instance of a class.*

What is the purpose of the final keyword in Java?

** The final keyword in Java is a non-access modifier that is used to indicate that a variable, method, or class cannot be modified or extended.which makes them non-changeable (impossible to inherit or override). And also guarantees that a variable is assigned only once.*

What is the purpose of the abstract keyword in Java?

** The abstract keyword is useful for building class hierarchies, defining a common interface for related classes, and ensuring that subclasses adhere to certain rules or contracts. It's commonly used in frameworks and libraries to define interfaces that multiple classes must implement.*

** The System.out.println() method in Java is used to print the argument passed to it to the console or the standard output stream. It is a fundamental method for displaying information and messages to the console.*

What is the purpose of the Scanner class in Java?

** It is primarily used to read user input and parse it into primitive data types such as int, double, or default String.and making it a valuable tool for interacting with users and handling data from different sources.*

What is the purpose of the Math class in Java?

** The Java Math class provides a set of methods for performing various mathematical operations. It is part of the java.lang package, which means it is automatically imported into every Java program. The Math class contains methods for performing basic numeric operations such as square roots, cube roots, exponential functions, trigonometric etc..*

What is the purpose of the StringBuilder class in Java?

** StringBuilder provides a more memory-efficient way to build and modify strings compared to using regular String concatenation (e.g., using the + operator). When you concatenate strings with +, a new string is created each time. you can change the contents of a StringBuilder without creating a new object each time.*

** StringBuilder is most valuable when you need to perform multiple modifications to a string or when you want to avoid excessive memory allocation for intermediate string objects during concatenation.*

What is the purpose of the equals() method in java?

** the equals() method in Java is used to compare the state or content of two objects for equality. By overriding equals(), you can customize what it means for two objects to be equal based on your specific needs*.

What is the purpose of the compareTo() method in Java?

** the compareTo() method is a powerful tool in Java that allows for the comparison and sorting of objects. It is particularly useful when you need to sort objects in a specific order or when you need to compare objects based on their natural ordering. The primary purpose of this method is to define a natural ordering or sorting for objects of the class.*

What is the purpose of the toString() method in Java?

* *The primary purpose of the toString() method is to provide a textual representation of an object and is often used for logging, debugging, and representing objects in a user interface.*

10 Java Programming Questions:

Write a Java program to print "Hello, World!" to the console.

program1.java

```
1  package com.bootcamp.level3;
2
3  public class program1{
4
5      public static void main(String[] args) {
6           System.out.println(" Hello, World!");
7            }
8          }
```

Write a Java program to find the sum of two numbers.

program2.java

```
1  package com.bootcamp.level3;
2
3  public class program2{
4
5      public static void main(String[] args) {
6           int num1 = 7;
7              int num2 = 8;
8              int sum = num1 + num2;
9              System.out.println("The sum of " + num1 + " and " + num2 + " is " + sum);
10          }
11        }
```

Write a Java program to find the average of three numbers.

Write a Java program to check if a given number is even or odd.

*program4.java

```java
package com.bootcamp.level3;


        import java.util.Scanner;

        public class program4{
            private static Scanner scanner;

            public static void main(String[] args) {
                scanner = new Scanner(System.in);
                System.out.print("Enter a number: ");
                int num = scanner.nextInt();

                if (num % 2 == 0) {
                    System.out.println(num + " is an even number.");
                } else {
                    System.out.println(num + " is an odd number.");
                }
            }
        }
```

Write a Java program to check if a given number is prime or not.

*program5.java

```java
package com.bootcamp.level3;

        import java.util.Scanner;

        public class program5{
                public static void main(String[] args) {
                    Scanner scanner = new Scanner(System.in);
                    System.out.print("Enter a number: ");
                    int num = scanner.nextInt();
                    if (isPrime(num)) {
                        System.out.println(num + " is a prime number.");
                    } else {
                        System.out.println(num + " is not a prime number.");
                    }
                    scanner.close();
                }
                private static boolean isPrime(int num) {
                    return false;
                }
        }
```

Write a Java program to check if a given string is a palindrome or not.

*program6.java

```java
package com.bootcamp.level3;
public class program6 {

    public static boolean isPalindrome(String str) {

        str = str.replaceAll("\\s", "").toLowerCase();

        int left = 0;
        int right = str.length() - 1;

        while (left < right) {
            if (str.charAt(left) != str.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }

        return true;
    }

    public static void main(String[] args) {
        String inputString = "A man a plan a canal Panama";

        if (isPalindrome(inputString)) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}
```

Write a Java program to implement a simple calculator.

*program8.java

```java
package com.bootcamp.level3;

import java.util.Scanner;
      //SimpleCalculator
public class program8 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the first number:");
        double num1 = scanner.nextDouble();

        System.out.println("Enter the second number:");
        double num2 = scanner.nextDouble();

        System.out.println("Enter an operator (+, -, *, /):");
        char operator = scanner.next().charAt(0);

        scanner.close();
        double output;

        switch(operator)
        {
            case '+':
                output = num1 + num2;
                break;

            case '-':
                output = num1 - num2;
                break;

            case '*':
                output = num1 * num2;
                break;

            case '/':
                if (num2 != 0) {
                    output = num1 / num2;
                } else {
                    System.out.println("Error! Division by zero is not allowed.");
                    return;
                }
                break;

            // operator doesn't match any case constant (+, -, *, /)
            default:
                System.out.println("Error! Invalid operator. Please enter correct operator.");
                return;
        }

        System.out.println(num1 + " " + operator + " " + num2 + " = " + output);
    }
}
```

Write a Java program to convert Fahrenheit to Celsius.

Write a Java program to generate a random number between 1 and 100.

program10.java

```java
package com.bootcamp.level3;

import java.util.Random;
// RandomNumber
public class program10 {
    public static void main(String[] args) {
        Random rand = new Random();
        int randomNum = rand.nextInt(100) + 1;
        System.out.println("Random number between 1 and 100: " + randomNum);
    }
}
```

3 Selenium Framework Questions:

Write a TestNG test to navigate to a website and verify the title of the page.

Write a TestNG test to fill out a login form on a website and verify successful login with valid credentials.

Write a TestNG test to navigate to a website, click on a link to navigate to a different page, and verify the presence of certain elements on the new page.

```java
package com.selenium;

import org.openqa.selenium.WebDriver;

Run All
public class VerifyTitle {


    @Test
    Run | Debug
    public void VerifyApplicationTitle() {

        WebDriver driver = new ChromeDriver();
        driver.manage().window().maximize();
        driver.get("https://www.wholefoodsmarket.com/");

         String my_title = driver.getTitle();

         System.out.println("Title is "+my_title);

         String expected_title = "Whole Foods Market | Weekly Sales | Shop In-Store and Online";

         Assert.assertEquals(my_title, expected_title);

         System.out.println("Test completed");


    }

}
```

Console × Results of running class VerifyTitle

<terminated> VerifyTitle [TestNG] C:\Users\yazid\Downloads\eclipse-jee-2023-06-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.7.v20230425-1

```
Title is Whole Foods Market | Weekly Sales | Shop In-Store and Online
Test completed
PASSED: VerifyApplicationTitle

===============================================
    Default test
    Tests run: 1, Failures: 0, Skips: 0
===============================================
```

```java
            System.out.println("page element exists");

        }else

        System.out.println("page element don't exists");
    }
    @AfterMethod
    public void tearDown() {
    }
}
```