

			Date: 2023-09-13
S.No: 1	Expt. Name: Sample programs on operator precedence and associativity		
All:	Open Interpreter/execute .java		
All:	Java program to demonstrate operators precedence and associativity		
S.No: 2	Expt. Name: Sample programs on joins to demonstrate Cartesian structures.		
All:	Java program that uses <= operator and print the result.		
Source Code:	<pre>import java.util.Scanner; class Operatorprecedence{ public static void main(String args[]){ Scanner sc = new Scanner(System.in); System.out.print("Enter first : "); String s1 = sc.nextLine(); System.out.print("Enter second : "); String s2 = sc.nextLine(); if(s1.equals("y") & s2.equals("x")){ System.out.println("x * y is greater than 20"); } else{ System.out.println("x * y less than 20"); } } }</pre>		
		Execution Results - All test cases have succeeded!	
		Test Case - 1	
User Output		User Output	
Enter a number:	4	Enter a number :	13
The product is 55.00000000000001		Enter second number :	5
result = 5		x * y is less than 20	
		Test Case - 2	
User Output		User Output	
Enter a number:	-3	Enter a number :	12
The product is 36.00000000000001		Enter second number :	13
result = 2		x * y is greater than 20	

S No. 3		Ex- Name: Sample Program to demonstrate constructor	Date: 2023-09-13
		<pre>using System; class Student { int id; string name; public Student(int id, string name) { this.id = id; this.name = name; } public void print() { System.out.println("id=" + id); System.out.println("name=" + name); } } public static void main(String args[]) { Student s1 = new Student(1, "abc"); s1.print(); }</pre>	<pre>using System; class Student { int id; string name; public Student(int id, string name) { this.id = id; this.name = name; } public void print() { System.out.println("id=" + id); System.out.println("name=" + name); } } public static void main(String args[]) { Student s1 = new Student(1, "abc"); s1.print(); }</pre>

Execution Results - All test cases have succeeded!	
	Test Case - 1
User Output	
B : null	
B : null	

S.No:	Exe Name	Program	Date
5	pattern	<pre>public class HalfPyramid { public static void main(String[] args) { Scanner sc = new Scanner(System.in); int n = sc.nextInt(); for (int i = 1; i <= n; i++) { for (int j = 1; j <= i; j++) { System.out.print("*"); } } } }</pre>	2023-09-13

Execution Results - All test cases have succeeded!	
	Test Case - 1
User Output	
Error no. of rows :	
5	
* * *	
* * *	
* * *	
* * *	
	Test Case - 2
User Output	
Error no. of rows :	
3	
* * *	
* * *	
	Test Case - 3

10

Execution Results - All test cases have succeeded!

S.No:	Exp Name:	Write the code to create an exception	Date:
1	22461A0579	public class DivideByZero { public static void main(String[] args) { int a = 10; int b = 0; System.out.println("Division by zero"); System.out.println("Exception caught : division by zero occurred"); } }	2022-11-02

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 26
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 2
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 27
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 28
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 29
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 30
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 31
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 32
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 33
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 34
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

Test Case - 1
User Output
Execution Result

Smritaa Banerjee Institute of Technology Page No: 35
ID: 22461A0579

```
public class DivideByZero {  
    public static void main(String[] args) {  
        int a = 10;  
        int b = 0;  
        System.out.println("Division by zero");  
        System.out.println("Exception caught : division by zero occurred");  
    }  
}
```

S No 20	Exp Name: Write the code for creating your own exception	Date: 2023-1-246
		Swami Bhānugan Institute of Technology 2022-2026-CE-B Page No: 33
	<pre>public class MyException extends Exception{ public void myException(){ System.out.println("MyException"); } } public class demo{ public static void main(String args){ try{ MyException e = new MyException(); e.myException(); } catch(MyException e){ System.out.println(e); } } }</pre>	ID: 22461A0597
	Execution Results - All test cases have succeeded!	
	User Output	%java demo MyException
S No 21	Exp Name: program that takes inputs 5 numbers, each between 10 and 100.	Date: 2023-1-246
		Swami Bhānugan Institute of Technology 2022-2026-CE-B Page No: 34
	<pre>public class demo{ public static void main(String args){ int arr[5]; Scanner sc = new Scanner(System.in); System.out.println("Enter 5 unique values between 10 & 100"); for(int i=0;i<5;i++){ arr[i] = sc.nextInt(); } for(int i=0;i<5;i++){ System.out.print("Entered value is "+arr[i]); if(i!=4){ System.out.print("\n"); } } } }</pre>	ID: 22461A0597
	Execution Results - All test cases have succeeded!	
	User Output	Test Case - 1 10 20 30 40 50
S No 22	Exp Name: A program to illustrate threads	Date: 2023-1-247
		Swami Bhānugan Institute of Technology 2022-2026-CE-B Page No: 35
	<pre>public class RandomThread implements Runnable{ public Thread t; public String threadname; RandomThread(String name){ threadname = name; } public void run(){ System.out.println("Creating "+threadname); for(int i=1;i<10;i++){ System.out.println("Running "+threadname); try{ Thread.sleep(100); } catch(InterruptedException e){ System.out.println("Thread "+threadname+" interrupted."); } } System.out.println("Starting "+threadname); t.start(); if(!t.isAlive()) System.out.println("Thread "+threadname+" exited."); } } public class TestThread{ public static void main(String args){ RandomThread t1 = new RandomThread("T1"); RandomThread t2 = new RandomThread("T2"); RandomThread t3 = new RandomThread("T3"); RandomThread t4 = new RandomThread("T4"); RandomThread t5 = new RandomThread("T5"); t1.start(); t2.start(); t3.start(); t4.start(); t5.start(); System.out.println("All threads suspended"); Thread.currentThread().suspended(); System.out.println("All threads resumed"); Thread.currentThread().resume(); } }</pre>	ID: 22461A0597
	Execution Results - All test cases have succeeded!	
	User Output	Test Case - 1 Test Case - 2 Error - unable to start threads 10 20 30 40 50 RandomThread@348 RandomThread@349 RandomThread@350 RandomThread@351 RandomThread@352
		Swami Bhānugan Institute of Technology 2022-2026-CE-B Page No: 36

Swimlane Baramugan Institute of Technology		ID: 2022-2026-CS-E-8	Page No: 43
User Output	User Output	ID: 2022-2026-CS-E-8	Scheme Baramugan Institute of Technology
<pre>import java.util.*; class Figure{ double base,height; double area; public void area(){ double s; s=(base+height)/2; area=s*s; } } class RectangleFigure extends Figure{ void area(){ System.out.println("Area is "+area); } } class TriangleFigure extends Figure{ void area(){ System.out.println("Area is "+(base*height)/2); } } class SquareFigure extends Figure{ void area(){ System.out.println("Area is "+(base*base)); } }</pre>	<pre>User Output Enter length and breadth of Rectangle : 12 14 Enter height and side of Triangle : 7 5 Enter base and side of Square : 3 4 TriangleFigure Area is 10.5 SquareFigure Area is 12.25</pre>	<pre>Test Case - 2 User Output Enter length and breadth of Rectangle : 4 5 Enter height and side of Triangle : 3 RectangleFigure Area is 12.0 SquareFigure Area is 7.5</pre>	<pre>Test Case - 2 User Output Enter length and breadth of Rectangle : 4 5 Enter height and side of Triangle : 3 RectangleFigure Area is 12.0 SquareFigure Area is 7.5</pre>
<pre>public static void main(String args[]) { System.out.println("Enter length and breadth of Rectangle : "); Scanner sc=new Scanner(System.in); double l=sc.nextDouble(); double b=sc.nextDouble(); double s=(l+b)/2; double area=s*s; System.out.println("Area is "+area); } public static void main(String args[]) { System.out.println("Enter height and side of Triangle : "); Scanner sc=new Scanner(System.in); double h=sc.nextDouble(); double s=sc.nextDouble(); double area=(h*s)/2; System.out.println("Area is "+area); } public static void main(String args[]) { System.out.println("Enter base and side of Square : "); Scanner sc=new Scanner(System.in); double b=sc.nextDouble(); double s=b*b; System.out.println("Area is "+s); }</pre>	<pre>System.out.println("Good morning for every 1 second for 5 times"); for(int i=1;i<=5;i++) { Thread.sleep(1000); System.out.println("Good morning for every 1 second for 5 times"); } public static void main(String args[]) { System.out.println("Good morning for every 1 second for 5 times"); new Thread(new Runnable() { public void run() { for(int i=1;i<=5;i++) { try { Thread.sleep(1000); } catch(InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); }</pre>	<pre>System.out.println("Good morning for every 1 second for 5 times"); for(int i=1;i<=5;i++) { Thread.sleep(1000); System.out.println("Good morning for every 1 second for 5 times"); } public static void main(String args[]) { System.out.println("Good morning for every 1 second for 5 times"); new Thread(new Runnable() { public void run() { for(int i=1;i<=5;i++) { try { Thread.sleep(1000); } catch(InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); }</pre>	<pre>System.out.println("Good morning for every 1 second for 5 times"); for(int i=1;i<=5;i++) { Thread.sleep(1000); System.out.println("Good morning for every 1 second for 5 times"); } public static void main(String args[]) { System.out.println("Good morning for every 1 second for 5 times"); new Thread(new Runnable() { public void run() { for(int i=1;i<=5;i++) { try { Thread.sleep(1000); } catch(InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); }</pre>
<p>Ques No: 25 Explain Write a Java program demonstrating the usage of Threads</p> <p>Aims: Write a Java program that uses three threads to perform the below actions: 1. First thread should print Good morning for every 1 second for 5 times 2. Second thread should print Hello for every 1 second for 5 times 3. Third thread should print Java for every 1 second for 5 times Two threads should run in parallel and the third thread should run sequentially. Write appropriate code in the main() method of the ThreadDemo class.</p> <p>Arguments: message, delay and count of type String, int and int respectively.</p> <p>Notes: If you want to sleep for 2 seconds you should call Thread.sleep(2000), as the ThreadDemo class mentioned in the main() method of the ThreadDemo class.</p> <p>Code:</p> <pre>public class ThreadDemo { public static void main(String[] args) { new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Hello for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Java for every 1 second for 5 times"); } } }).start(); } }</pre>	<p>Ques No: 25 Explain Write a Java program demonstrating the usage of Threads</p> <p>Aims: Write a Java program that uses three threads to perform the below actions: 1. First thread should print Good morning for every 1 second for 5 times 2. Second thread should print Hello for every 1 second for 5 times 3. Third thread should run sequentially. Two threads should run in parallel and the third thread should run sequentially. Write appropriate code in the main() method of the ThreadDemo class.</p> <p>Arguments: message, delay and count of type String, int and int respectively.</p> <p>Notes: If you want to sleep for 2 seconds you should call Thread.sleep(2000), as the ThreadDemo class mentioned in the main() method of the ThreadDemo class.</p> <p>Code:</p> <pre>public class ThreadDemo { public static void main(String[] args) { new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Hello for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Java for every 1 second for 5 times"); } } }).start(); } }</pre>	<p>Ques No: 25 Explain Write a Java program demonstrating the usage of Threads</p> <p>Aims: Write a Java program that uses three threads to perform the below actions: 1. First thread should print Good morning for every 1 second for 5 times 2. Second thread should print Hello for every 1 second for 5 times 3. Third thread should run sequentially. Two threads should run in parallel and the third thread should run sequentially. Write appropriate code in the main() method of the ThreadDemo class.</p> <p>Arguments: message, delay and count of type String, int and int respectively.</p> <p>Notes: If you want to sleep for 2 seconds you should call Thread.sleep(2000), as the ThreadDemo class mentioned in the main() method of the ThreadDemo class.</p> <p>Code:</p> <pre>public class ThreadDemo { public static void main(String[] args) { new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Hello for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Java for every 1 second for 5 times"); } } }).start(); } }</pre>	<p>Ques No: 25 Explain Write a Java program demonstrating the usage of Threads</p> <p>Aims: Write a Java program that uses three threads to perform the below actions: 1. First thread should print Good morning for every 1 second for 5 times 2. Second thread should print Hello for every 1 second for 5 times 3. Third thread should run sequentially. Two threads should run in parallel and the third thread should run sequentially. Write appropriate code in the main() method of the ThreadDemo class.</p> <p>Arguments: message, delay and count of type String, int and int respectively.</p> <p>Notes: If you want to sleep for 2 seconds you should call Thread.sleep(2000), as the ThreadDemo class mentioned in the main() method of the ThreadDemo class.</p> <p>Code:</p> <pre>public class ThreadDemo { public static void main(String[] args) { new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Good morning for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Hello for every 1 second for 5 times"); } } }).start(); new Thread(new Runnable() { public void run() { for (int i = 1; i <= 5; i++) { try { Thread.sleep(1000); } catch (InterruptedException e) { e.printStackTrace(); } System.out.println("Java for every 1 second for 5 times"); } } }).start(); } }</pre>

S.No	Exp. Name	Program to find and replace pattern in a given file	Date				
SNo.26	Edu. Name	Program to find and replace pattern in a given file	Date- 2023-12-25				
		Code:					
		<pre>package edu.java; public class Replacefile { public static void main(String[] args) throws IOException { Thread t1 = new Thread(new Printer("test sentence", 1, 2)); Thread t2 = new Thread(new Printer("test sentence", 1, 3)); t1.start(); t2.start(); t2.join(); t1.join(); System.out.println("All the three threads t1, t2 and t3 have completed execution."); } } class Printer implements Runnable { String str; PrintWriter print; int b, c; String message; boolean done; String name; public void run() { for(int i=0;i<1000;i++) { System.out.print(name); if(i>b) print.write(message); else print.write(" "); if(c>c) done=true; else done=false; if(done) break; } Thread.sleep(5000+((int)(Math.random()*1000))); } catch(InterruptedException e) { System.out.print(e); } } </pre>					
		Source Code:					
		<pre>Q:\2023\Replacefile.java package edu.java; import java.io.*; import java.util.*; class Replacefile { public static void main(String args) { try { File file = new File("file1.txt"); BufferedReader reader = new BufferedReader(new FileReader(file)); String line; while((line = reader.readLine()) != null) { if(line.equals("test")) line = "New"; else line += " "; writer = new PrintWriter(file); writer.write(line); writer.close(); } } catch (IOException e) { e.printStackTrace(); } } }</pre>					
		Output:					
		<pre>Q:\2023\Replacefile.java D:\224G1A0579@DESKTOP-CE8E8 ~ % java Replacefile All the three threads t1, t2 and t3 have completed execution.</pre>					
		Summary Report/Output of Testbedlogy:					
		<pre>D:\224G1A0579@DESKTOP-CE8E8 ~ % java Replacefile D:\224G1A0579@DESKTOP-CE8E8 ~ %</pre>					
		File.txt					
		<pre>This is test string 2000. The test string is replaced with your input string. check the string you entered is available here.</pre>					
		Execution Results - All test cases have succeeded!					
		Test Case - 1					
		<table border="1"> <thead> <tr> <th>User Input</th> <th>Output</th> </tr> </thead> <tbody> <tr> <td>test</td> <td>test</td> </tr> </tbody> </table>	User Input	Output	test	test	
User Input	Output						
test	test						

Execution Results - All test cases have succeeded!	
User Output	Test Case - 1
None	String: This is test string. The test string is replaced with your input.
None	String: This is test string. The test string is replaced with your input.
None	String: This is test string. The test string is replaced with your input.

Siamese Ramangran Institute of Technology 2022-2026-CE-B

S.No: 28 Date: 2023-2-17	Epi. Name: Stock Implementation Altin: Create an interface for stock with push and pop operations. Implements the stock in two ways: fixed-size stack and dynamic stock where size is determined when stack is full. Note: Please do not change the package name.	Source Code: https://github.com/AlbinDobrovolc/CSStack_1.java
---	---	--

Bamangjan Institute of Technology [202-2036-CSE-8]

Execution Results - All test cases have succeeded!

Execution Results - All test cases have succeeded!
Test Case - 1

```

public class Finstack implements Stack {
    private int size;
    private int top;
    private StackNode head;
    class StackNode implements Node {
        StackNode left;
        StackNode right;
        String data;
        StackNode(int data) {
            this.data = data + " ";
            System.out.println("Stack node created with value and increased");
        }
        void push(int item) {
            if (left == null) {
                left = new StackNode(item);
            } else {
                left.push(item);
            }
        }
        void pop() {
            if (right == null) {
                System.out.println("Stack underflow");
                return;
            }
            right = right.left;
        }
        void print() {
            System.out.print(data);
        }
    }
    public void push(int item) {
        head = new StackNode(item);
        size++;
    }
    public int pop() {
        if (size == 0) {
            System.out.println("Stack underflow");
            return -1;
        }
        int item = head.data;
        head = head.right;
        size--;
        return item;
    }
    public void print() {
        StackNode temp = head;
        while (temp != null) {
            temp.print();
            temp = temp.right;
        }
    }
}

```


S.No : 32	Ex- Name: Write java programs that use collection framework classes (linkedHashMap class)	Date: 2023-12-04
-----------	---	------------------

hello
string:
three
corresponding string:
two/two
length/length entries :

S.No:	Ex. Name Write Java program that use collection framework classess.
24	<p>Date: 2023-12-04</p> <pre> public static void main(String[] args) { try { BufferedReader reader = new BufferedReader(new FileReader("C:\\Users\\Administrator\\Desktop\\file.txt")); System.out.println("No. of strings in file: " + reader.readLine()); reader.close(); String str = reader.readLine(); int count = 0; while (str != null) { if (str.startsWith("www.")) { System.out.println(str); count++; } str = reader.readLine(); } System.out.println("Count of URLs: " + count); } catch (IOException e) { e.printStackTrace(); } } </pre>

Summary Report	
Test Case ID	Description
TC-001	Test Case 1: Verify successful login with valid credentials.
TC-002	Test Case 2: Verify failed login attempt with invalid credentials.
TC-003	Test Case 3: Verify user profile update functionality.
TC-004	Test Case 4: Verify search functionality for products.
TC-005	Test Case 5: Verify cart management and checkout process.
TC-006	Test Case 6: Verify password reset feature.
TC-007	Test Case 7: Verify multi-step form submission.
TC-008	Test Case 8: Verify session timeout behavior.
TC-009	Test Case 9: Verify responsive design across different devices.
TC-010	Test Case 10: Verify end-to-end purchase flow from start to finish.

hello
string:
three
corresponding string:
two/two
length/length entries :

S.No:	Ex. Name Write Java program that use collection framework classess.
24	<p>Date: 2023-12-04</p> <pre> public static void main(String[] args) { try { BufferedReader reader = new BufferedReader(new FileReader("C:\\Users\\Administrator\\Desktop\\file.txt")); System.out.println("No. of strings in file: " + reader.readLine()); reader.close(); String str = reader.readLine(); int count = 0; while (str != null) { if (str.startsWith("www.")) { System.out.println(str); count++; } str = reader.readLine(); } System.out.println("Count of URLs: " + count); } catch (IOException e) { e.printStackTrace(); } } </pre>

```
1. arraylist printing by using Stream();
2.
3.
```

```
S No. 35 Exp. Name: Write java program(s) that use collection framework classes.(ArrayList class)
Date: 2023-1-204
```

```
ArrayListExample.java
import java.util.*;
import java.util.ArrayList;
class ArrayListExample {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<String>();
        list.add("Hello");
        list.add("World");
        System.out.println("Size of the list: " + list.size());
        System.out.println("Elements present in list by using Stream():");
        list.stream().forEach(System.out::println);
    }
}
```

Test Case - 2

Test Case - 1

Test Case - 2

Test Case - 3

Test Case - 4

Test Case - 5

Test Case - 6

Test Case - 7

Test Case - 8

Test Case - 9

Test Case - 10

Test Case - 11

Test Case - 12

Test Case - 13

Test Case - 14

Test Case - 15

Test Case - 16

Test Case - 17

Test Case - 18

Test Case - 19

Test Case - 20

Test Case - 21

Test Case - 22

Test Case - 23

Test Case - 24

Test Case - 25

Test Case - 26

Test Case - 27

Test Case - 28

Test Case - 29

Test Case - 30

Test Case - 31

Test Case - 32

Test Case - 33

Test Case - 34

Test Case - 35

Test Case - 36

Test Case - 37

Test Case - 38

Test Case - 39

Test Case - 40

Test Case - 41

Test Case - 42

Test Case - 43

Test Case - 44

Test Case - 45

Test Case - 46

Test Case - 47

Test Case - 48

Test Case - 49

Test Case - 50

Test Case - 51

Test Case - 52

Test Case - 53

Test Case - 54

Test Case - 55

Test Case - 56

Test Case - 57

Test Case - 58

Test Case - 59

Test Case - 60

Test Case - 61

Test Case - 62

Test Case - 63

Test Case - 64

Test Case - 65

Test Case - 66

Test Case - 67

Test Case - 68

Test Case - 69

Test Case - 70

Test Case - 71

Test Case - 72

Test Case - 73

Test Case - 74

Test Case - 75

Test Case - 76

Test Case - 77

Test Case - 78

Test Case - 79

Test Case - 80

Test Case - 81

Test Case - 82

Test Case - 83

Test Case - 84

Test Case - 85

Test Case - 86

Test Case - 87

Test Case - 88

Test Case - 89

Test Case - 90

Test Case - 91

Test Case - 92

Test Case - 93

Test Case - 94

Test Case - 95

Test Case - 96

Test Case - 97

Test Case - 98

Test Case - 99

Test Case - 100

Test Case - 101

Test Case - 102

Test Case - 103

Test Case - 104

Test Case - 105

Test Case - 106

Test Case - 107

Test Case - 108

Test Case - 109

Test Case - 110

Test Case - 111

Test Case - 112

Test Case - 113

Test Case - 114

Test Case - 115

Test Case - 116

Test Case - 117

Test Case - 118

Test Case - 119

Test Case - 120

Test Case - 121

Test Case - 122

Test Case - 123

Test Case - 124

Test Case - 125

Test Case - 126

Test Case - 127

Test Case - 128

Test Case - 129

Test Case - 130

Test Case - 131

Test Case - 132

Test Case - 133

Test Case - 134

Test Case - 135

Test Case - 136

Test Case - 137

Test Case - 138

Test Case - 139

Test Case - 140

Test Case - 141

Test Case - 142

Test Case - 143

Test Case - 144

Test Case - 145

Test Case - 146

Test Case - 147

Test Case - 148

Test Case - 149

Test Case - 150

Test Case - 151

Test Case - 152

Test Case - 153

Test Case - 154

Test Case - 155

Test Case - 156

Test Case - 157

Test Case - 158

Test Case - 159

Test Case - 160

Test Case - 161

Test Case - 162

Test Case - 163

Test Case - 164

Test Case - 165

Test Case - 166

Test Case - 167

Test Case - 168

Test Case - 169

Test Case - 170

Test Case - 171

Test Case - 172

Test Case - 173

Test Case - 174

Test Case - 175

Test Case - 176

Test Case - 177

Test Case - 178

Test Case - 179

Test Case - 180

Test Case - 181

Test Case - 182

Test Case - 183

Test Case - 184

Test Case - 185

Test Case - 186

Test Case - 187

Test Case - 188

Test Case - 189

Test Case - 190

Test Case - 191

Test Case - 192

Test Case - 193

Test Case - 194

Test Case - 195

Test Case - 196

Test Case - 197

Test Case - 198

Test Case - 199

Test Case - 200

Test Case - 201

Test Case - 202

Test Case - 203

Test Case - 204

Test Case - 205

Test Case - 206

Test Case - 207

Test Case - 208

Test Case - 209

Test Case - 210

Test Case - 211

Test Case - 212

Test Case - 213

Test Case - 214

Test Case - 215

Test Case - 216

Test Case - 217

Test Case - 218

Test Case - 219

Test Case - 220

Test Case - 221

Test Case - 222

Test Case - 223

Test Case - 224

Test Case - 225

Test Case - 226

Test Case - 227

Test Case - 228

Test Case - 229

Test Case - 230

Test Case - 231

Test Case - 232

Test Case - 233

Test Case - 234

Test Case - 235

Test Case - 236

Test Case - 237

Test Case - 238

Test Case - 239

Test Case - 240

Test Case - 241

Test Case - 242

Test Case - 243

Test Case - 244

Test Case - 245

Test Case - 246

Test Case - 247

Test Case - 248

Test Case - 249

Test Case - 250

Test Case - 251

Test Case - 252

Test Case - 253

Test Case - 254

Test Case - 255

Test Case - 256

Test Case - 257

Test Case - 258

Test Case - 259

Test Case - 260

Test Case - 261

Test Case - 262

Test Case - 263

Test Case - 264

Test Case - 265

Test Case - 266

Test Case - 267

Test Case - 268

Test Case - 269

Test Case - 270

Test Case - 271

Test Case - 272

Test Case - 273

Test Case - 274

Test Case - 275

Test Case - 276

Test Case - 277

Test Case - 278

Test Case - 279

Test Case - 280

Test Case - 281

Test Case - 282

Test Case - 283

Test Case - 284

Test Case - 285

Test Case - 286

Test Case - 287

Test Case - 288

Test Case - 289

Test Case - 290

Test Case - 291

Test Case - 292

Test Case - 293

Test Case - 294

Test Case - 295

Test Case - 296

Test Case - 297

Test Case - 298

Test Case - 299

Test Case - 300

Test Case - 301

Test Case - 302