S.NO	Programs	Date	Page No	Signature
Week-1				
1	Write the steps to download and install Java.			
2	Write a java program to print the message “Welcome to java programming”.			
3	Write a java program that prints student details(name, roll number and section of a student).			
Week-2				
1	Write a java program to calculate the area of a rectangle.			
2a)	Write a program to convert temperature from Celsius to Fahrenheit			
2b)	Write a program to convert temperature from Fahrenheit to Celsius.			
3	Write a program to calculate the simple interest			
4	Write a program to find the largest of two numbers using ternary operator			
5	Write a program to find the factorial of a number			
Week-3				

1.	Create a java program with following instructions a)create a class with name car b) Create 4 attributes name car color, car brand, fuel type, milage. c) Create 3 methods named start, stop, services d) Create 3objects named car1, car2, car3. e) Create a constructor which should print “welcome to car garage”			
2.	Write a java program to create a class BackAccount with two methods deposit( ) and withdraw() b) In deposit( ) whenever an amount is deposited it has to be updated with current amount b) In withdraw( ) whenever an amount is withdrawn it has to be less than current amount else print “Insufficient funds”			
Week-4				
1.	Write a java program with class named “Book”. The class should contain various attributes such as “Title of the book , author , year of publication “. It should also contain a constructor with parameters details of the			

	book. i.e. " Title of the book, author and year of publication". Display the details of two books by creating two objects.			
2.	To create a java program with class named Myclass with a staticvariable "Count" of "int type", Initialized to 0 and a constant variable "pi" of type double initialized to 3.1415 as attributes of that class Now, define a constructor for "Myclass" that increments the "Count" variable each that an object of Myclass is created. Finally , print the final values of "Count" and "pi"variables.			
Week-5				
1.	Implement a java program using the below Array list i) Insert an element at particular index in the array list ii) Modify an element in the array list iii)access an element from the array list iv) remove an element from the array list.			
2.	A vehicle rental company wants to develop a system			

	that maintains information about different types of vechicles available for rent the company rents out cars and bikes, and they need a program to store details about each vehicle, such as brand and speed(should be in super class) 1. cars should have an additional property: no.of doors 2. Bikes should have a property indicating whether they have gears or not. 3. The system should also include a function to display details about each vehicle and indicate when a vehicle is starting. 4. Every class should have a constructor **Question:** 1. Which oops concept is used in the above program 2. If the company decides to add a new type of vehicle, Truck, how would you modify the program? a. Truck should include an additional property capacity (in tons) b. Create a showTruckdetails() method to display			

	the truck's capacity. c. Write a constructor for Truck that initializes all properties 3. Implement the truck class and update the main method to create a Truck object and also create an object for car and bike sub classes Finally, display the details.			
Week-6				
1.	Write a java program to create a Vehicle class with displayInfo() method , overridden in Car subclass to provide info about carcompany , model , price ,seating and petrol.			
2.	A college is developing an automated admission system that verifies students eligibility(UG) and postgraduation(PG) programs. Each program has different eligibility criteria based on the students percentage in their previous qualification. 1. UG admission recquire a minimum of 60%.			

	2. PG admission recquire a minimum of 70%.			
3.	Create a calculator class with overloaded methods to perform additions 1.add two integers 2.add two double values 3.add three integers			
4.	Create a shape class with method calculateArea() that is overloaded for different shapes (eg: square, rectangle).Then create a subclass Circle that overrides calculateArea() method for Circle.			
Week-7				
1.	Write a Java program to create an abstract class Animal with an abstract method called sound(). Create subclasses Lion and Tiger that extend the Animal class and implement the sound() method to make a specific sound for each animal			
2.	.Write a Java program to create an abstract class Shape3D with abstract methods calculateVolume() and calculateSurfaceArea(). Create subclasses Sphere and Cube that extend the Shape3D class and implement the respective methods to calculate the volume and surface area of each shape			

3.	write a java program using an abstract class to define a method for pattern printing • Create an abstract class named patternprinter with an abstract method printpattern(int n) and a concrete method to display the pattern title. • Implement two subclasses 1. Star pattern prints a right angle triangle of Star(*) 2. Numberpattern-prints a right angled triangle of increasing numbers • In the main( ) method ,create objects of both subclasses and print the patterns for a given number of rows. Expected output: Pattern 1: * * * * * * * * * * * * * * * Pattern 2: 1 1 2 1 2 3 1 2 3 4 1 2 3 4 5			
Week-8				
1.	Write a Java program to create an interface Shape with the getPerimeter() method. Create three getPerimeter() method for each of the three classes.			

2.	Write a Java program to create an interface Shape with the getPerimeter() method. Create three getPerimeter() method for each of the three classes.			
3.	Write a java program to implement a login system using intefaces.			
Week-9				
1.	Write a java program to create a method that take integer as parameter and throws an example if the number is even			
2.	Write a java program to create a method that reads a file and throws an exception if the file is not found			
3.	Write a java program to handle arithematic exception using try catch and finally .			
4.	Write a java program to simulate a university system using inner classes			
Week-10				
1.	Write a java program to generate a password for a student using his/her intials and age. The password displayed should be the string consists of first character of first name,			

	middle name last name with age.			
2.	Design and implement a java program that will do the following questions to this string "Welcome! You are practicing Strings concept". i) Convert all the alphabets to capital letters and print out the result ii) Convert all alphabets to lower-case letters and print out the result iii) print out the length of the string iv) Print out the index of the concept			
3.	Implement a java program using the below array methods i) Sorting the elements (numbers and strings) of an array ii) Convert the array elements into string iii) Fill the part of an array iv) Copy the elements of one array into another.			
4.	Implement a java program using the below Array list i) Insert an element at particular index in the array list ii) Modify an element in the array list iii)access an element from the array list iv) remove an element from the array list.			

WEEK-09

1. **Write a java program to create a method that take integer as parameter and throws an example if the number is even.**

Class Diagram:

checkNumber
+checkNumber(number:int):void throws Exception
+main(args:String():void

Code:-

```
public class checkNumber {

public static void checkNumber(int number) throws Exception {

if (number % 2 == 0) {
throw new Exception("Even number is not allowed: " + number);
} else {
System.out.println("Valid output number: " + number);
}}
public static void main(String[] args) {
try {
checkNumber(9);
} catch (Exception e) {
System.out.println("Exception caught: " + e.getMessage());
}}}
```

Output:

```
C:\Users\smoha\Java Prgms>javac  checkNumber.java

C:\Users\smoha\Java Prgms>java  checkNumber
Valid output number: 9
```

Error Table:-

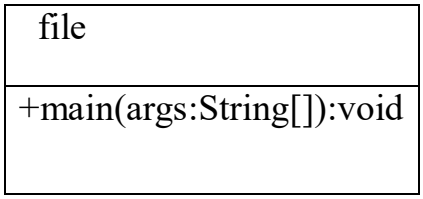
S.no	Expected Error	Error rectification
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

IMPORTANT PONTS:

- **Method Declaration**
  checkNumber(int number) is declared with throws Exception, meaning it can throw an exception that must be handled by the caller.
- 🔹 **Validation Logic**
  The method checks if the number is even (number % 2 == 0). If yes, it throws an exception.
- 🔹 **Use of throw**
  Uses throw new Exception(...) to manually throw an exception when the input is invalid (even number).
- 🔹 **Exception Message**
  The exception includes a custom message: "Even number is not allowed: " + number.

2.Write a java program to create a method that reads a file and throws an exception if the file is not found .

Class Diagram:-



Code:-

```
import java.io.BufferedReader;
 import java.io.FileReader;
 import java.io.IOException;
 public class file {
    public static void main(String[] args) {
        BufferedReader br = null; // Declare BufferedReader outside the try
block
        try {
            br = new BufferedReader(new FileReader("E:/Amrita/example.txt"));
            String line; // Corrected 'string' to 'String'
            while ((line = br.readLine()) != null) {
                System.out.println(line); // Print the actual line instead of the string
"line"
            }
        } catch (IOException e) { // Catch IOException directly
            System.out.println("An error occurred while reading the file: " +
e.getMessage());
        } finally {
            if (br != null) {
                try {
                    br.close();
                } catch (IOException e) {
                    System.out.println("Error closing the BufferedReader: " +
e.getMessage());
                }}}}}
```

Output:

```
C:\Users\smoha\Java Prgms>javac  file.java

C:\Users\smoha\Java Prgms>java  file
An error occurred while reading the file: E:\Amrita\example.txt
(The system cannot find the path specified)
```

Error Table:-

S.no	Expected Error	Error rectification
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

IMPORTANT PONTS:

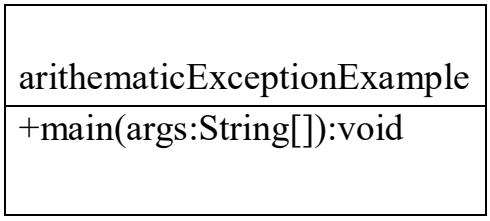
Method Declaration
checkNumber(int number) is declared with throws Exception, meaning it can throw an exception that must be handled by the caller.

Validation Logic
The method checks if the number is even (number % 2 == 0). If yes, it throws an exception.

Try-Catch Block
In the main method, a try-catch block is used to catch the exception thrown by checkNumber.

3.Write a java program to handle arithematic exception using try catch and finally .

Class Diagram:



Code:-

```
import java.util.Scanner;

public class arithematicExceptionExample {
public static void main(String[] args) {
Scanner input = new Scanner(System.in);

try {
System.out.println("Enter first number (numerator):");
int a = input.nextInt();

System.out.println("Enter second number (denominator):");
int b = input.nextInt();
```

```
int result = a / b;
System.out.println("Result of division: " + result);
} catch (ArithmeticException e) {
System.out.println("Error: Cannot divide by zero.");
} catch (Exception e) {
System.out.println("Error: " + e.getMessage());
} finally {
input.close();
}}}
```

Output:

```
C:\Users\smoha\Java Prgms>javac   ArithmeticExceptionExample.java

C:\Users\smoha\Java Prgms>java   ArithmeticExceptionExample
Enter first number (numerator):
8
Enter second number (denominator):
2
Result of division: 4
```

Error Table:-

S.no	Expected Error	Error rectification
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

IMPORTANT PONTS:

- **Class Purpose**
  The program demonstrates handling of arithmetic exceptions during division.
- **User Input with Scanner**
  Scanner is used to take two integer inputs from the user: numerator and denominator.
- **ArithmeticException Handling**
  If the denominator is zero, ArithmeticException is caught and a specific error message is shown.

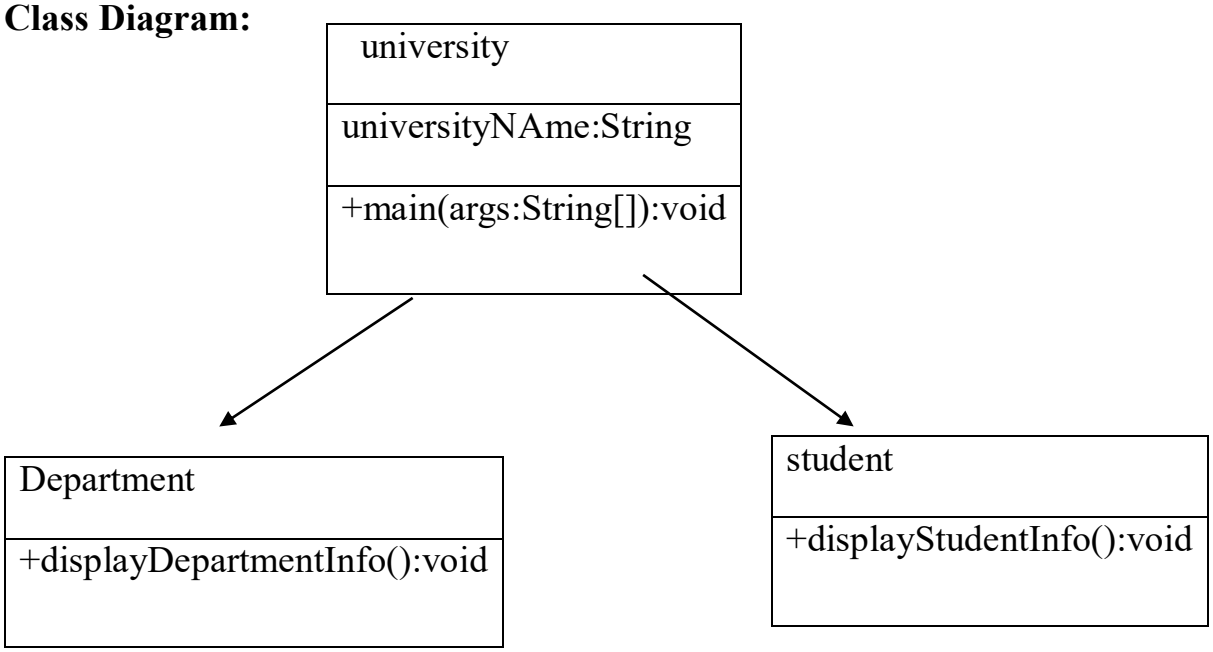
- **Generic Exception Catch**
  A general Exception block is included to catch any unexpected errors (like invalid input).

4. Write a java program to simulate a university system using inner classes

- ✓ Create an outer class namedd University with a variable UniversityName
- ✓ Inside it defgine two non-static in classes

1. Department-With variable like deptName and deptCode and a method to display department details.
2. Student-Variable like stdName and stdCode and a method to display Student details.
3. Create an object for each class and call their methods to display their details and with the university name.

Class Diagram:



Code:-

```
public class university {

    String universityName = "Amrita University";
```

```
    class Department {

        String deptName = "computer science";
        int deptcode = 101;

        void displayDepartmentInfo() {
            System.out.println("department" + deptName);
            System.out.println("department" + deptcode);
        }
    }

    class student {

        String stdname = "Sai Krishna";
        int stdcode = 18977;

        void displayStudentInfo() {
            System.out.println("department" + stdname);
            System.out.println("department" + stdcode);

        }
    }

    public static void main(String[] args) {
        university uni = new university();
        System.err.println("University" + uni.universityName);
        System.err.println("Department__Info");
        university.Department dept = uni.new Department();
        dept.displayDepartmentInfo();

        System.out.println("***** Student Info *****");
        university.student stdent = uni.new student();
        stdent.displayStudentInfo();
    }
}
```

Output:

```
C:\Users\smoha\Java Prgms>javac  university.java

C:\Users\smoha\Java Prgms>java  university
UniversityAmrita University
Department__Info
departmentcomputer science
department101
***** Student Info *****
departmentS Mohana Maheswari
department18777
```

Error Table:

S.no	Expected Error	Error rectification
1	Setting the parameters inside the constructor	We cannot pass the values inside constructor without setting them first
2	}	Ending the class and main method is required

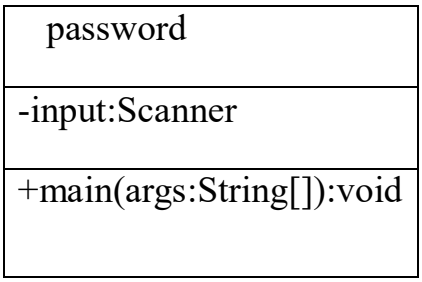
IMPORTANT PONTS:

- Outer Class Definition
  The class university represents a university and contains inner classes for departments and students.
- **Member Variable in Outer Class**
  It has a String variable universityName initialized with "Amrita University".
- **Non-Static Inner Classes**
  Two non-static inner classes are defined: Department and student, each with its own attributes and methods.
- **Department Inner Class**
  Department has deptName and deptcode, and a method displayDepartmentInfo() to print them.
- **Student Inner Class**
  student has stdname and stdcode, and a method displayStudentInfo() to print them.

WEEK-10

1. Write a java program to generate a password for a student using his/her intials and age. The password displayed should be the string consists of first character of first name, middle name last name with age.

Class Diagram:



Code:-

```
import java.lang.String;
import java.util.Scanner;
public class password {
   public static void main(String[] args) {

     Scanner input=new Scanner(System.in);

     System.out.println("enter the first name");
     String FN=input.next();
     System.out.println("enter the last name");
     String LN=input.next();
     System.out.println("enter the age");
     int AGE=input.nextInt();
     String initials=FN.substring(0,1)+LN.substring(0,1)+AGE;
     String PIN=initials.toLowerCase();
    System.out.println(initials+":- is the password created");
}
    }
```

Output:

```
C:\Users\smoha\Java Prgms>javac  password.java

C:\Users\smoha\Java Prgms>java  password
enter the first name
Mohana
enter the last name
Maheswari
enter the age
18
MM18:- is the password created
```

Error Table:-

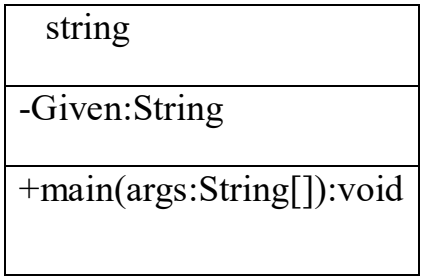
S.no	Expected Error	Error rectification
1	}	Ending the class and main method is required

IMPORTANT PONTS:

- Scanner for Input
  The program uses Scanner to take user input for first name, last name, and age.
- String Manipulation
  Uses substring(0,1) to extract the first character from both the first and last names.
- Password Creation Logic
  Concatenates the initials of the names with the age to form a password.
- Case Formatting
  Converts the generated password to lowercase using toLowerCase() and stores it in PIN.

2. Design and implement a java program that will do the following questions to this string "Welcome! You are practicing Strings concept".
i) Convert all the alphabets to capital letters and print out the result
ii) Convert all alphabets to lower-case letters and print out the result
iii) print out the length of the string
iv) Print out the index of the concept.

Class Diagram:



Code:-

```
import java.lang.String;
public class string{
   public static void main(String[] args) {
      String Given="welcome! You are practicing strings concept";
      System.out.println("Converting into upper case letters :
"+Given.toUpperCase());
      System.out.println("Converting into lower case letters :
"+Given.toLowerCase());
      System.out.println("Resulting the length of the string : "+Given.length());
      System.out.println("Finding the index of the given String
:"+Given.indexOf("concept"));
   }
}
```

Output:

```
C:\Users\smoha\Java Prgms>javac  string.java

C:\Users\smoha\Java Prgms>java  string
Converting into upper case letters : WELCOME! YOU ARE PRACTICING STRINGS CONCEPT
Converting into lower case letters : welcome! you are practicing strings concept
Resulting the length of the string : 43
Finding the index of the given String :36
```

Error:

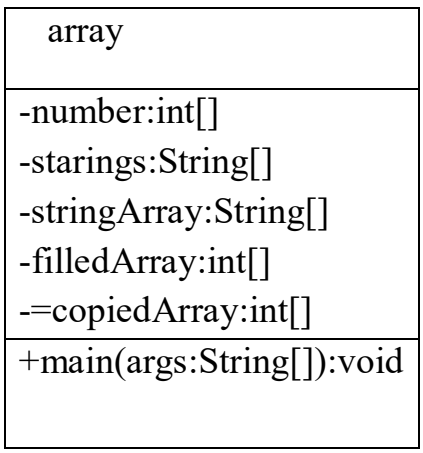
Code Error	error rectification
1. error: ';' expected	1. we must end line with semicolon.

IMPORTANT PONTS:

- String Declaration
  A String variable named Given is initialized with a sentence for manipulation.
- Uppercase Conversion
  toUpperCase() is used to convert the entire string to uppercase letters.
- Lowercase Conversion
  toLowerCase() converts the entire string to lowercase letters.
- Finding String Length
  length() method returns the number of characters in the string.
- Finding Substring Index
  indexOf("concept") finds and returns the starting index of the word "concept".

3.Implement a java program using the below array methods
i) Sorting the elements (numbers and strings) of an array
ii) Convert the array elements into string
iii) Fill the part of an array
iv) Copy the elements of one array into another.

Class Diagram:



Code:-

```
import java.util.Arrays;
public class array {
   public static void main(String[] args) {
      // 1. Sorting the elements (numbers)
      int[] numbers = {5, 3, 8, 1, 2};
      System.out.println("Original numbers array: " +
Arrays.toString(numbers));
      Arrays.sort(numbers);
      System.out.println("Sorted numbers array: " + Arrays.toString(numbers));
      // 1. Sorting the elements (strings)
      String[] strings = {"Banana", "Apple", "Orange", "Mango"};
      System.out.println("Original strings array: " + Arrays.toString(strings));
      Arrays.sort(strings);
      System.out.println("Sorted strings array: " + Arrays.toString(strings));
      // 2. Convert the array elements into strings
      String[] stringArray = Arrays.stream(numbers)
                           .mapToObj(String::valueOf)
                           .toArray(String[]::new);
      System.out.println("Converted numbers array to strings: " +
Arrays.toString(stringArray));
      // 3. Fill part of an array
      int[] filledArray = new int[10];
      Arrays.fill(filledArray, 0, 5, 7); // Fill first 5 elements with 7
      System.out.println("Array after filling part of it: " +
Arrays.toString(filledArray));
      // 4. Copy the elements of one array into another
      int[] copiedArray = new int[numbers.length];
      System.arraycopy(numbers, 0, copiedArray, 0, numbers.length);
      System.out.println("Copied array: " + Arrays.toString(copiedArray));
      // Close the scanner
   }}
```

Output:

```
C:\Users\smoha\Java Prgms>javac   array.java

C:\Users\smoha\Java Prgms>java   array
Original numbers array: [5, 3, 8, 1, 2]
Sorted numbers array: [1, 2, 3, 5, 8]
Original strings array: [Banana, Apple, Orange, Mango]
Sorted strings array: [Apple, Banana, Mango, Orange]
Converted numbers array to strings: [1, 2, 3, 5, 8]
Array after filling part of it: [7, 7, 7, 7, 7, 0, 0, 0, 0, 0]
Copied array: [1, 2, 3, 5, 8]
```

Error:

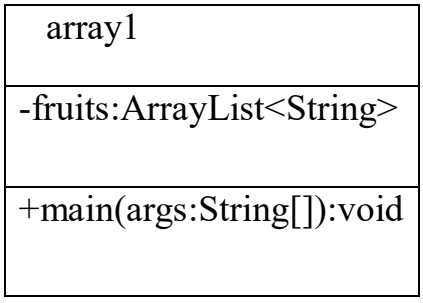
Code Error	error rectification
1. error: ';' expected	1. we must end line with semicolon.

IMPORTANT PONTS:

- Array Sorting (Integers)
  Arrays.sort(numbers) sorts the integer array in ascending order.
- Array Sorting (Strings)
  Arrays.sort(strings) arranges string elements alphabetically (lexicographically).
- Array to String Conversion
  Uses Arrays.stream(numbers).mapToObj(String::valueOf) to convert an integer array to a string array.
- Using Arrays.toString()
  Arrays.toString(array) is used throughout to print array contents in readable format.
- Demonstrates Java Utility Methods
  Showcases various utility methods provided by the java.util.Arrays class.

4.Implement a java program using the below Array list
i) Insert an element at particular index in the array list
ii) Modify an element in the array list
iii)access an element from the array list
iv) remove an element from the array list.

Class Diagram:



Code:-

```
import java.util.ArrayList;
public class array1{
   public static void main(String[] args) {
       ArrayList<String> fruits = new ArrayList<>();
       fruits.add("Apple");
       fruits.add("Banana");
       fruits.add("Orange");
       System.out.println("Original ArrayList: " + fruits);
       fruits.add(1, "Mango");
       System.out.println("After inserting 'Mango' at index 1: " + fruits);
       fruits.set(2, "Grapes");
       System.out.println("After modifying element at index 2: " + fruits);
       String fruitAtIndex3 = fruits.get(3);
       System.out.println("Element at index 3: " + fruitAtIndex3);
       fruits.remove("Banana");
       System.out.println("After removing 'Banana': " + fruits);
       fruits.remove(0);
       System.out.println("After removing element at index 0: " + fruits);
   }
}
```

Output:

```
C:\Users\smoha\Java Prgms>javac  array1.java

C:\Users\smoha\Java Prgms>java  array1
Original ArrayList: [Apple, Banana, Orange]
After inserting 'Mango' at index 1: [Apple, Mango, Banana, Orange]
After modifying element at index 2: [Apple, Mango, Grapes, Orange]
Element at index 3: Orange
After removing 'Banana': [Apple, Mango, Grapes, Orange]
After removing element at index 0: [Mango, Grapes, Orange]
```

Error:

Code Error	**error rectification**
1. error: ';' expected	1. we must end line with semicolon.

IMPORTANT PONTS:

- ArrayListDeclaration
  An ArrayList<String> named fruits is created to store a list of fruit names.
- **AddingElements** fruits.add(...) is used to add elements to the ArrayList; "Apple", "Banana", "Orange" are added initially.
- **InsertingatIndex**
  fruits.add(1, "Mango") inserts "Mango" at index 1, shifting other elements.
- **Dynamic List Behavior**
  The ArrayList resizes automatically as elements are added or removed.
- **No User Input**
  All operations are hardcoded; no input is taken from the user.
- **Demonstrates Core List Operations**
  The program demonstrates key ArrayList methods: add(), set(), get(), and remove().