NIET
Greater Noida
Autonomous Institute

OOTs Using Java Workshop

Name : Kunwar Shaurya Pratap Singh

Course: B.Tech

Branch: Computer Science

Roll number : 2401330120086

Lab : 308B

Day 1: Classwork

Question: Write a Java program to print 'Hello World'.

Solution:

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay1/HelloWorld
Hello World

Process finished with exit code 0
```

Question: Write a Java program to take name and number as input and display them.

Solution:

```
import java.util.Scanner;

public class Name_Number {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name: ");
        String a = sc.nextLine();
        System.out.print("Enter number: ");
        long n = sc.nextLong();
        System.out.println("Name: " + a);
        System.out.println("Number: " + n);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay1/Name_Number
Enter name: Shaurya
Enter number: 1234567890
Name: Shaurya
Number: 1234567890

Process finished with exit code 0
```

Question: Sketch a class diagram containing a class called Employee, which models an employee with an ID, name and salary. Add a method `raiseSalary(percent)` that increases the salary by the given percentage.

Solution:

```
import java.util.Scanner;

public class SalaryHike {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Salary hike of 10%");
        System.out.print("Enter the emp id: ");
        int empId = sc.nextInt();

        switch (empId) {
            case 1:
                String name = "Kunal";
                int salary = 100000;
                double newSalary = hikePercentage(salary) + salary;
                System.out.println("EmpName: " + name);
                System.out.println("New Salary : " + newSalary);
                break;
            case 2:
                name = "Rohit";
                salary = 10000;
                newSalary = hikePercentage(salary) + salary;
                System.out.println("EmpName: " + name);
                System.out.println("New Salary : " + newSalary);
                break;
            case 3:
                name = "Pankaj";
                salary = 1000;
                newSalary = hikePercentage(salary) + salary;
                System.out.println("EmpName: " + name);
                System.out.println("New Salary : " + newSalary);
                break;
            default:
                System.out.println("INVALID ID!");
        }
    }

    public static double hikePercentage(int salary) {
        double hikeOf = (0.1) * salary;
        return hikeOf;
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay1/SalaryHike
Salary hike of 10%
Enter the emp id: 1
EmpName: Kunal
New Salary : 110000.0

Process finished with exit code 0
```

Day 1: Homework

Question: Program to display default value of all Primitive data types.

Solution:

```
public class DefaultValues {
    static byte b;
    static short s;
    static int i;
    static long l;
    static float f;
    static double d;
    static char c;
    static boolean bool;

    public static void main(String[] args) {
        System.out.println("Default value of byte: " + b);
        System.out.println("Default value of short: " + s);
        System.out.println("Default value of int: " + i);
        System.out.println("Default value of long: " + l);
        System.out.println("Default value of float: " + f);
        System.out.println("Default value of double: " + d);
        System.out.println("Default value of char: " + c);
        System.out.println("Default value of boolean: " + bool);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay1/DefaultValues
Default value of byte: 0
Default value of short: 0
Default value of int: 0
Default value of long: 0
Default value of float: 0.0
Default value of double: 0.0
Default value of char:
Default value of boolean: false

Process finished with exit code 0
```

Question: Implement the code using main() method to calculate and print the Total and Average marks scored by a student.

Solution:

```
import java.util.Scanner;

public class AverageCalc {
    public static void main(String[] args) {
        System.out.print("Enter name: ");
        Scanner sc = new Scanner(System.in);
        String name = sc.nextLine();
        System.out.print("Marks1: ");
        int marks1 = sc.nextInt();
        System.out.print("Marks2: ");
        int marks2 = sc.nextInt();
        System.out.print("Marks3: ");
        int marks3 = sc.nextInt();
        int total = marks1 + marks2 + marks3;
        float avg = total / 3;
        System.out.println("Total marks are:" + total);
        System.out.print("Average is: " + avg);
    }
```

```
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay1/AverageCalc
Enter name: Kunwar Shaurya Pratap Singh
Marks1: 95
Marks2: 92
Marks3: 98
Total marks are:285
Average is: 95.0

Process finished with exit code 0
```

Day 2: Classwork

Question: Write code which uses if-then-else statement to check if a given account balance is greater or lesser than the minimum balance.

Solution:

```
import java.util.Scanner;

public class checkBalance {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the balance: ");
        int bal = sc.nextInt();
        if (bal >= 1000) {
            System.out.println("Sufficient balance");
        } else {
            System.out.println("Balance is low");
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay2/checkBalance
Enter the balance:
500
Balance is low

Process finished with exit code 0
```

Question: A class NumberPalindrome with a public method isNumberPalindrome that takes one parameter number of type int. Write a code to check whether the given number is palindrome or not.

Solution:

```
import java.util.Scanner;
```

```
public class NumberPallindrome {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number: ");
        int num = sc.nextInt();
        int n = num;
        int sum = 0;
        while (num > 0) {
            int rem = num % 10;
            sum = sum * 10 + rem;
            num = num / 10;
        }
        if (sum == n) {
            System.out.println("The number is palindrome");
        } else {
            System.out.println("The number is not a Palindrome");
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay2/NumberPallindrome
Enter the number:
12321
The number is palindrome

Process finished with exit code 0
```

Question: Write a class FibonacciSeries with a main method. The method receives one command line argument. Write a program to display fibonacci series.

Solution:

```
import java.util.Scanner;

public class Fibonacci {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number of terms: ");
        int n = sc.nextInt();
        int a = 0;
        int b = 1;
        int count = 2;
        System.out.print(a + " ");
        System.out.print(b + " ");
        while (count <= n) {
            int temp = b;
            b = b + a;
            a = temp;
            count++;
            System.out.print(b + " ");
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay2/Fibonacci
Enter the number of terms:
8
0 1 1 2 3 5 8 13
Process finished with exit code 0
```

Question: Write a Java Program to find the Factorial of a given number.

Solution:

```
import java.util.Scanner;

public class factorial {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number: ");
        int num = sc.nextInt();
        long fact = 1;
        if (num == 0 || num == 1) {
            System.out.println("Factorial: 1");
        } else {
            while (num > 0) {
                fact = fact * num;
                num--;
            }
            System.out.println("Factorial: " + fact);
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay2/factorial
Enter the number:
6
Factorial: 720

Process finished with exit code 0
```

Day 2: Homework

Question: Java Program to create a class, methods and invoke them inside main method.

Solution:

```
public class invoke { // calling a function is known as invoking
    public static void main(String[] args) {
        greeting();
        morning();
        int a = 1;
        int b = 2;
        add(a, b);
    }
```

```
    public static void greeting() {
        System.out.println("Welcome");
    }

    public static void morning() {
        System.out.println("Good Morning Pineapple!");
    }

    public static void add(int a, int b) {
        int c = a + b;
        System.out.println(c);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay2/invoke
Welcome
Good Morning Pineapple!
3

Process finished with exit code 0
```

Question: Write a Java program to illustrate the abstract class concept.

Solution:

```
abstract class Shape {
    public abstract void numberOfSides();
}

class Trapezoid extends Shape {
    @Override
    public void numberOfSides() {
        System.out.println("A trapezoid has 4 sides.");
    }
}

class Triangle extends Shape {
    @Override
    public void numberOfSides() {
        System.out.println("A triangle has 3 sides.");
    }
}

class Hexagon extends Shape {
    @Override
    public void numberOfSides() {
        System.out.println("A hexagon has 6 sides.");
    }
}

public class AbstractShape {
    public static void main(String[] args) {
        Shape trapezoid = new Trapezoid();
        Shape triangle = new Triangle();
        Shape hexagon = new Hexagon();
```

```
        trapezoid.numberOfSides();
        triangle.numberOfSides();
        hexagon.numberOfSides();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay2/AbstractShape
A trapezoid has 4 sides.
A triangle has 3 sides.
A hexagon has 6 sides.

Process finished with exit code 0
```

Question: Java program to illustrate the static field in the class.

Solution:

```
class Student {
    static String college = "My University";
    int studentId;
    String name;

    Student(int id, String n) {
        studentId = id;
        name = n;
    }

    void display() {
        System.out.println("ID: " + studentId + ", Name: " + name + ",
College: " + college);
    }
}

public class StaticField_Illus {
    public static void main(String[] args) {
        Student s1 = new Student(101, "Alice");
        Student s2 = new Student(102, "Bob");

        s1.display();
        s2.display();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay2/StaticField_Illus
ID: 101, Name: Alice, College: My University
ID: 102, Name: Bob, College: My University

Process finished with exit code 0
```

Day 3: Classwork

Question: Write a Java Program to illustrate a static class.

Solution:

```
class University {

    static String universityName = "Global Tech University";

    String establishedYear = "1998";
    static class Department {

        String departmentName;

        public Department(String name) {
            this.departmentName = name;
        }

        public void displayDetails() {
            System.out.println("Department: " + departmentName);

            System.out.println("University: " + universityName);

            // The following line would cause a compile-time error because
a static
            // nested class cannot access instance members of the outer
class.
            // System.out.println("Established In: " + establishedYear);
        }
    }
}


public class StaticClassDemo {
    public static void main(String[] args) {
        University.Department csDept = new University.Department("Computer
Science & Engineering");

        csDept.displayDetails();
    }}
```

Output:

```
Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWor
kDay3/University
Department: Computer Science & Engineering
University: Global Tech University

Process finished with exit code 0
```

Question: Java program to access the class members using super keyword.

Solution:

```
class parent {
    String name = "I am Parent Class";
}

class child extends parent {
    String name = "I am child class";
    void display() {
```

```
        System.out.println(name);
        System.out.println(super.name);
    }
}

public class superKeyword {
    public static void main(String[] args) {
        child c1 = new child();
        c1.display();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay3/superKeyword
I am child class
I am Parent Class

Process finished with exit code 0
```

Question: Java program to access the class members using this keyword.

Solution:

```
class Student {
    String name;
    int age;

    Student(String name, int age) {
        this.name = name;
        this.age = age;
    }

    void display() {
        System.out.println("Name: " + this.name + ", Age: " + this.age);
    }
}

public class thisKeyword {
    public static void main(String[] args) {
        Student S1 = new Student("Xavier", 22);
        S1.display();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay3/thisKeyword
Name: Xavier, Age: 22

Process finished with exit code 0
```

Question: Implement an interface named MountainParts that has a constant named TERRAIN that will store the String value "off-road".

Solution:

```
interface MountainParts {
    String TERRAIN = "off-road";
    void setSuspension(String newValue);
    String getSuspension();
    void setType(String newValue);
    String getType();
}

public class MountainBike implements MountainParts {
    private String suspension;
    private String type;

    public void setSuspension(String newValue) {
        suspension = newValue;
    }

    public String getSuspension() {
        return suspension;
    }

    public void setType(String newValue) {
        type = newValue;
    }

    public String getType() {
        return type;
    }

    public static void main(String[] args) {
        MountainBike bike = new MountainBike();
        bike.setSuspension("Dual");
        bike.setType("Trail");
        System.out.println("Terrain: " + MountainParts.TERRAIN);
        System.out.println("Suspension: " + bike.getSuspension());
        System.out.println("Type: " + bike.getType());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay3/MountainBike
Terrain: off-road
Suspension: Dual
Type: Trail

Process finished with exit code 0
```

Day 3: Homework

Question: Java program to demonstrate nested interface inside a interface.

Solution:

```
interface outerinterface {
    void outermethod();
    interface interinterface {
```

```
        void innermethod();
    }
}

class nestedclass implements outerinterface.interinterface {
    public void innermethod() {
        System.out.println("inner interface method");
    }
}

public class nestedInterface {
    public static void main(String[] args) {
        outerinterface.interinterface obj1 = new nestedclass();
        obj1.innermethod();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay3/nestedInterface
inner interface method

Process finished with exit code 0
```

Question: Java program to demonstrate nested interface inside a class.

Solution:

```
class Shape {
    // Nested static class
    static class TriShape {
        void showSides() {
            System.out.println("A triangle has three sides");
        }
    }
}

public class triangle {
    public static void main(String[] args) {
        Shape.TriShape obj = new Shape.TriShape();
        obj.showSides();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay3/triangle
A triangle has three sides

Process finished with exit code 0
```

Day 4: Classwork

Question: Java program to implement Single Inheritance.

Solution:

```
class Animal1 {
    void eat() {
        System.out.println("This animal eats food");
    }
}

class Dog1 extends Animal1 {
    void bark() {
        System.out.println("Dogs Barks");
    }
}

public class Single_Inheritance {
    public static void main(String[] args) {
        Dog1 d = new Dog1();
        d.bark();
        d.eat();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay4/Single_Inheritance
Dogs Barks
This animal eats food

Process finished with exit code 0
```

Question: Java program to implement multi-level inheritance.

Solution:

```
class Vehicle1 {
    void start() {
        System.out.println("Vehicle is starting...");
    }
}

class Car1 extends Vehicle1 {
    void drive() {
        System.out.println("Car is moving");
    }
}

class ElectricCar extends Car1 {
    void charge() {
        System.out.println("Electric car is charging.");
    }
}

public class multilvl_inheritence {
    public static void main(String[] args) {
        ElectricCar myEv = new ElectricCar();
        myEv.start();
        myEv.drive();
        myEv.charge();
    }
```

```
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay4/multilvl_inheritence
Vehicle is starting...
Car is moving
Electric car is charging.

Process finished with exit code 0
```

Question: Java program to implement constructor and constructor overloading.

Solution:

```
class Box {
    double width, height, depth;

    Box(double w, double h, double d) {
        width = w;
        height = h;
        depth = d;
    }
    Box() {
        width = height = depth = 0;
    }
    Box(double len) {
        width = height = depth = len;
    }
    double volume() {
        return width * height * depth;
    }
}

public class ConstructorOverloading {
    public static void main(String args[]) {
        Box mybox1 = new Box(10, 20, 15);
        Box mybox2 = new Box();
        Box mycube = new Box(7);

        System.out.println("Volume of mybox1 is " + mybox1.volume());
        System.out.println("Volume of mybox2 is " + mybox2.volume());
        System.out.println("Volume of mycube is " + mycube.volume());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay4/ConstructorOverloading
Volume of mybox1 is 3000.0
Volume of mybox2 is 0.0
Volume of mycube is 343.0

Process finished with exit code 0
```

Question: Java program to implement method overloading.

Solution:

```
class Calculator {
    int add(int a, int b) { return a + b; }
    int add(int a, int b, int c) { return a + b + c; }
    double add(double a, double b) { return a + b; }
}

public class MethodOverloadingEx {
    public static void main(String[] args) {
        Calculator calc = new Calculator();
        System.out.println("Sum of 2 and 3 is: " + calc.add(2, 3));
        System.out.println("Sum of 2, 3, and 4 is: " + calc.add(2, 3, 4));
        System.out.println("Sum of 2.5 and 3.5 is: " + calc.add(2.5, 3.5));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay4/MethodOverloadingEx
Sum of 2 and 3 is: 5
Sum of 2, 3, and 4 is: 9
Sum of 2.5 and 3.5 is: 6.0

Process finished with exit code 0
```

Question: Java program to implement method overriding.

Solution:

```
class Vehicle2 {
    void run() {
        System.out.println("Vehicle is running");
    }
}

class Car2 extends Vehicle2 {
    void run() {
        System.out.println("Car is running safely");
    }
}

public class overriding_example {
    public static void main(String args[]) {
        Car2 obj = new Car2();
        obj.run();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay4/overriding_example
Car is running safely
```

```
Process finished with exit code 0
```

Day 5: Classwork

Question: Java program to implement lambda expression without parameter.

Solution:

```
interface MyFunctionalInterface {
    String sayHello();
}

public class LambdaNoParams {
    public static void main(String[] args) {
        MyFunctionalInterface msg = () -> "Hello, World!";
        System.out.println(msg.sayHello());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/LambdaNoParams
Hello, World!

Process finished with exit code 0
```

Question: Java program to implement lambda expression with single parameter.

Solution:

```
interface FuncInterface {
    void abstractfun(int x);
    default void normal() {
        System.out.println("Hello");
    }
}

public class LambdaSingleParam {
    public static void main(String[] args) {
        FuncInterface fobj = (int x) -> System.out.println(2 * x);
        fobj.abstractfun(5);
        fobj.normal();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/LambdaSingleParam
10
Hello

Process finished with exit code 0
```

Question: Java program to define lambda expressions as method parameters.

Solution:

```
import java.util.function.Consumer;

public class LambdaAsMethodParameter {
    public static void processString(String str, Consumer<String>
processor) {
        processor.accept(str);
    }

    public static void main(String[] args) {
        String greeting = "Hello Lambda!";
        processString(greeting, (s) -> System.out.println("Printing: " +
s));
        processString(greeting, (s) -> System.out.println("Length: " +
s.length()));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/LambdaAsMethodParameter
Printing: Hello Lambda!
Length: 13

Process finished with exit code 0
```

Question: Write a class CountOfTwoNumbers with a public method compareCountOf that takes three parameters and returns true if count of arg1 is greater than arg2 in arr.

Solution:

```
public class CountOfTwoNumbers {
    public boolean compareCountOf(int[] arr, int arg1, int arg2) {
        int count1 = 0;
        int count2 = 0;
        for (int num : arr) {
            if (num == arg1) count1++;
            if (num == arg2) count2++;
        }
        return count1 > count2;
    }

    public static void main(String[] args) {
        CountOfTwoNumbers counter = new CountOfTwoNumbers();
        int[] sampleArray = { 1, 2, 3, 1, 1, 4, 5, 2, 1 };
        int num1 = 1;
        int num2 = 2;
        boolean result = counter.compareCountOf(sampleArray, num1, num2);
        System.out.println("Is the count of " + num1 + " greater than the
count of " + num2 + "? " + result);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/CountOfTwoNumbers
Is the count of 1 greater than the count of 2? true

Process finished with exit code 0
```

Question: Java program to show the multiplication of two matrices using arrays.

Solution:

```
public class MatrixMultiplication {
    public static void main(String[] args) {
        int[][] firstMatrix = { { 3, -2, 5 }, { 3, 0, 4 } };
        int[][] secondMatrix = { { 2, 3 }, { -9, 0 }, { 0, 4 } };
        int[][] product = new
int[firstMatrix.length][secondMatrix[0].length];

        for (int i = 0; i < firstMatrix.length; i++) {
            for (int j = 0; j < secondMatrix[0].length; j++) {
                for (int k = 0; k < firstMatrix[0].length; k++) {
                    product[i][j] += firstMatrix[i][k] *
secondMatrix[k][j];
                }
            }
        }
        System.out.println("Product of the matrices is:");
        for (int[] row : product) {
            for (int column : row) {
                System.out.print(column + "    ");
            }
            System.out.println();
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/MatrixMultiplication
Product of the matrices is:
24    29
6    25

Process finished with exit code 0
```

Question: Java Program to search an element using Linear Search.

Solution:

```
import java.util.Scanner;

public class linear_Searching {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = {10, 20, 30, 40, 50};
```

```
        System.out.println("Array: 10 20 30 40 50");
        System.out.print("Enter the target value: ");
        int target = sc.nextInt();
        int result = linear(arr, target);
        if(result != -1)
            System.out.println("Element found at index: " + result);
        else
            System.out.println("Element not found in the array.");
    }

    static int linear(int[] arr, int target) {
        for (int i = 0; i < arr.length; i++) {
            if (arr[i] == target) return i;
        }
        return -1;
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/linear_Searching
Array: 10 20 30 40 50
Enter the target value: 30
Element found at index: 2

Process finished with exit code 0
```

Question: Java program to search an element using Binary Search.

Solution:

```
import java.util.Scanner;
import java.util.Arrays;

public class binary_Searching {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int[] arr = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
        System.out.println("Sorted Array: " + Arrays.toString(arr));
        System.out.print("Enter the target value: ");
        int target = sc.nextInt();
        int res = binary(arr, target);
        if (res == -1) {
            System.out.println("The target doesn't exist.");
        } else
            System.out.println("Target found at index: " + res);
    }

    static int binary(int[] arr, int target) {
        int start = 0, end = arr.length - 1;
        while (start <= end) {
            int mid = start + (end - start) / 2;
            if (target < arr[mid]) end = mid - 1;
            else if (target > arr[mid]) start = mid + 1;
            else return mid;
        }
        return -1;
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay5/binary_Searching
Sorted Array: [2, 5, 8, 12, 16, 23, 38, 56, 72, 91]
Enter the target value: 23
Target found at index: 5

Process finished with exit code 0
```

Day 5: Homework

Question: Java program to implement lambda expression with multi parameter.

Solution:

```
interface StringConcat {
    String concat(String a, String b);
}

public class LambdaMultiParams {
    public static void main(String[] args) {
        StringConcat sc = (str1, str2) -> str1 + str2;
        System.out.println("Result: " + sc.concat("Hello ", "World"));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay5/LambdaMultiParams
Result: Hello World

Process finished with exit code 0
```

Question: Java program to implement lambda expression that iterate list of objects.

Solution:

```
import java.util.ArrayList;
import java.util.List;

public class LambdaList {
    public static void main(String[] args) {
        List<String> fruits = new ArrayList<>();
        fruits.add("Apple");
        fruits.add("Banana");
        fruits.add("Cherry");
        fruits.add("Date");

        System.out.println("Printing list elements:");
        fruits.forEach(System.out::println);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay5/LambdaList
Printing list elements:
Apple
Banana
Cherry
Date

Process finished with exit code 0
```

Question: Java Program to sort element using insertion Sort.

Solution:

```
import java.util.Arrays;

public class insertion_Sort {
    public static void sort(int[] arr) {
        for (int i = 1; i < arr.length; 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;
        }
    }

    public static void main(String[] args) {
        int[] data = { 9, 5, 1, 4, 3 };
        System.out.println("Unsorted Array: " + Arrays.toString(data));
        sort(data);
        System.out.println("Sorted Array : " + Arrays.toString(data));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay5/insertion_Sort
Unsorted Array: [9, 5, 1, 4, 3]
Sorted Array : [1, 3, 4, 5, 9]

Process finished with exit code 0
```

Question: Java Program to sort element using Selection Sort.

Solution:

```
import java.util.Arrays;

public class SelectionSort {
    public static void sort(int[] arr) {
        for (int i = 0; i < arr.length - 1; i++) {
```

```
            int minIndex = i;
            for (int j = i + 1; j < arr.length; j++) {
                if (arr[j] < arr[minIndex]) {
                    minIndex = j;
                }
            }
            int temp = arr[minIndex];
            arr[minIndex] = arr[i];
            arr[i] = temp;
        }
    }

    public static void main(String[] args) {
        int[] data = { 20, 12, 10, 15, 2 };
        System.out.println("Unsorted Array: " + Arrays.toString(data));
        sort(data);
        System.out.println("Sorted Array : " + Arrays.toString(data));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay5/SelectionSort
Unsorted Array: [20, 12, 10, 15, 2]
Sorted Array : [2, 10, 12, 15, 20]

Process finished with exit code 0
```

Question: Java program to Sort elements using Bubble Sort.

Solution:

```
import java.util.Arrays;

public class BubbleSort {
    public static void main(String[] args) {
        int[] arr = { 64, 34, 25, 12, 22, 11, 90 };
        System.out.println("Unsorted array: " + Arrays.toString(arr));
        bubble(arr);
        System.out.println("Sorted array: " + Arrays.toString(arr));
    }

    static void bubble(int[] arr) {
        boolean swapped;
        for (int i = 0; i < arr.length - 1; i++) {
            swapped = false;
            for (int j = 0; j < arr.length - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                    swapped = true;
                }
            }
            if (!swapped) break;
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay5/BubbleSort
Unsorted array: [64, 34, 25, 12, 22, 11, 90]
Sorted array: [11, 12, 22, 25, 34, 64, 90]

Process finished with exit code 0
```

Day 7: Classwork

Question: Java program to create user defined package.

Solution: *File 1: `greetings.java` (inside `com/shaurya` package)*

```
package com.shaurya;

public class greetings {
    public void displayMessage() {
        System.out.println("Hello from the 'shaurya' package!");
    }
}
```

File 2: `Main.java` (inside `com` package)

```
package com;
import com.shaurya.greetings;

public class Main {
    public static void main(String[] args) {
        greetings hello = new greetings();
        hello.displayMessage();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay7/com/Main
Hello from the 'shaurya' package!

Process finished with exit code 0
```

Question: Implement and demonstrate package names collision in java.

Solution:

```
import java.util.*;
import java.awt.*;

public class collison {
    public static void main(String[] args) {
        // Using fully qualified name to avoid ambiguity
        java.util.List<String> stringList = new ArrayList<>();
        stringList.add("No collision here!");
```

```
        java.awt.List awtList = new java.awt.List();
        awtList.add("This one is fine too.");

        System.out.println(stringList.get(0));
        System.out.println(awtList.getItem(0));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay7/collison
No collision here!
This one is fine too.

Process finished with exit code 0
```

Question: Java program to handle an Arithmetic Exception Divided by zero.

Solution:

```
public class ArithmeticExceptionDemo {
    public static void main(String[] args) {
        try {
            int a = 30;
            int b = 0;
            int c = a / b;
            System.out.println("Result: " + c);
        } catch (ArithmeticException e) {
            System.out.println("Caught an exception: Cannot divide by
zero.");
        }
        System.out.println("Program continues after the exception.");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay7/ArithmeticExceptionDemo
Caught an exception: Cannot divide by zero.
Program continues after the exception.

Process finished with exit code 0
```

Question: Java Program to implement User Defined Exception.

Solution:

```
class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
    }
}
```

```
class BankAccount {
    private double balance;

    public BankAccount(double initialBalance) {
        this.balance = initialBalance;
    }

    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount > balance) {
            throw new InsufficientFundsException("Withdrawal amount exceeds
balance.");
        }
        balance -= amount;
        System.out.println("Withdrawal successful. New balance: " +
balance);
    }
}

public class UserDefinedException {
    public static void main(String[] args) {
        BankAccount account = new BankAccount(1000);
        try {
            account.withdraw(500);
            account.withdraw(600);
        } catch (InsufficientFundsException e) {
            System.err.println("Error: " + e.getMessage());
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay7/UserDefinedException
Withdrawal successful. New balance: 500.0
Error: Withdrawal amount exceeds balance.

Process finished with exit code 0
```

Day 7: Homework

Question: Java program to illustrate finally block.

Solution:

```
public class FinallyBlockEx {
    public static void main(String[] args) {
        try {
            System.out.println("Inside the try block.");
            int result = 10 / 0;
            System.out.println("This line will not be executed.");
        } catch (ArithmeticException e) {
            System.out.println("Caught ArithmeticException.");
        } finally {
            System.out.println("Inside the finally block. This always
runs!");
        }
        System.out.println("Program continues...");
    }
```

```
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay7/FinallyBlockEx
Inside the try block.
Caught ArithmeticException.
Inside the finally block. This always runs!
Program continues...

Process finished with exit code 0
```

Question: Java program to illustrate Multiple catch blocks.

Solution:

```
public class MultiCatch {
    public static void main(String[] args) {
        try {
            int[] a = new int[5];
            a[5] = 30 / 0; // This will cause ArithmeticException first
        } catch (ArithmeticException e) {
            System.out.println("Caught an ArithmeticException: Division by
zero.");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Caught an
ArrayIndexOutOfBoundsException.");
        } catch (Exception e) {
            System.out.println("Caught a general exception.");
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay7/MultiCatch
Caught an ArithmeticException: Division by zero.

Process finished with exit code 0
```

Question: Java program for creation of illustrating throw in exception handling.

Solution:

```
public class ExceptionHandling {
    public static void validateAge(int age) {
        if (age < 18) {
            throw new ArithmeticException("Person is not eligible to
vote.");
        } else {
            System.out.println("Person is eligible to vote.");
        }
    }
```

```
    public static void main(String[] args) {
        try {
            validateAge(13);
        } catch (ArithmeticException e) {
            System.out.println("Exception caught: " + e.getMessage());
        }
        System.out.println("Program continues...");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay7/ExceptionHandling
Exception caught: Person is not eligible to vote.
Program continues...

Process finished with exit code 0
```

Question: Implement the concept of Assertion in Java Programming Language.

Solution:

```
import java.util.Scanner;

public class Assertion_illus {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number between 0 and 10: ");
        int value = scanner.nextInt();

        // To run this, you need to enable assertions with the -ea flag
        // Example: java -ea Assertion_illus
        assert (value >= 0 && value <= 10) : "The number is not within the
valid range!";

        System.out.println("You entered: " + value);
        scanner.close();
    }
}
```

Output (with assertions enabled and invalid input):

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay7/Assertion_illus
Enter a number between 0 and 10: 15
Exception in thread "main" java.lang.AssertionError: The number is not
within the valid range!
        at Assertion_illus.main(Assertion_illus.java:10)

Process finished with exit code 1
```

Day 8: Classwork

Question: Implement the concept of Localization in Java Programming Language.

Solution:

```
import java.util.*;

public class LocalizationExample {
    public static void main(String[] args) {
        // Note: Requires Messages_fr.properties and Messages_hi.properties
files
        Locale french = new Locale("fr");
        Locale hindi = new Locale("hi");
        Locale defaultLocale = new Locale("en");

        printMessage(french);
        printMessage(hindi);
        printMessage(defaultLocale);
    }

    public static void printMessage(Locale locale) {
        ResourceBundle bundle = ResourceBundle.getBundle("Messages",
locale);
        System.out.println(locale.getDisplayLanguage() + ": " +
bundle.getString("greeting"));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay8/LocalizationExample
French: Bonjour
Hindi: नमस्ते
English: Hello

Process finished with exit code 0
```

Question: Java program to print the output by appending all the capital letters in the input string.

Solution:

```
public class StringHandling {
    public static void main(String[] args) {
        String input = "Hello World, This Is JAVA";
        StringBuilder capitals = new StringBuilder();

        for (char c : input.toCharArray()) {
            if (Character.isUpperCase(c)) {
                capitals.append(c);
            }
        }
        System.out.println("Original String: " + input);
        System.out.println("Capital letters: " + capitals.toString());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay8/StringHandling
Original String: Hello World, This Is JAVA
Capital letters: HWTIJAVA

Process finished with exit code 0
```

Question: Java program that prints the duplicate characters from the string with its count.

Solution:

```
public class DuplicateCharCount {
    public static void main(String[] args) {
        String s1 = "Beautiful";
        System.out.println("The String is: " + s1);
        System.out.print("The Duplicate characters in a string: ");
        char[] string = s1.toCharArray();
        int count;

        for(int i = 0; i <string.length; i++) {
            count = 1;
            for(int j = i+1; j <string.length; j++) {
                if(string[i] == string[j] && string[i] != ' ') {
                    count++;
                    //Set string[j] to 0 to avoid printing visited
character
                    string[j] = '0';
                }
            }
            //A character is considered as duplicate if count is greater
than 1
            if(count > 1 && string[i] != '0')
                System.out.println(string[i] + ", count = " + count);
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay8/DuplicateCharCount
The String is: Beautiful
The Duplicate characters in a string: u, count = 2

Process finished with exit code 0
```

Question: Java program to check if two strings are anagrams of each other.

Solution:

```
import java.util.Arrays;
import java.util.Scanner;

public class anagram {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = sc.nextLine();
```

```
        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();

        str1 = str1.replaceAll("\\s", "").toLowerCase();
        str2 = str2.replaceAll("\\s", "").toLowerCase();

        if (str1.length() != str2.length()) {
            System.out.println("Not Anagrams");
            return;
        }

        char[] charArray1 = str1.toCharArray();
        char[] charArray2 = str2.toCharArray();

        Arrays.sort(charArray1);
        Arrays.sort(charArray2);

        if (Arrays.equals(charArray1, charArray2)) {
            System.out.println("Strings are anagrams");
        } else {
            System.out.println("Not Anagrams");
        }
        sc.close();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay8/anagram
Enter first string: Listen
Enter second string: Silent
Strings are anagrams

Process finished with exit code 0
```

Day 8: Homework

Question: Java Program to count the total number of characters in a string.

Solution:

```
public class string_length {
    public static void main(String[] args) {
        String a = "Hello World";
        int length = a.length();
        System.out.println("The length of the string is: " + length);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay8/string_length
The length of the string is: 11

Process finished with exit code 0
```

Question: Java Program to count the total number of punctuation characters exists in a String.

Solution:

```
public class punctuation_Count {
    public static void main(String[] args) {
        String p = "Hello! This is a Ball. How are you?";
        int count = 0;
        for (int i = 0; i < p.length(); i++) {
            char ch = p.charAt(i);
            if (ch == '!' || ch == ',' || ch == ';' || ch == '.' || ch ==
'?' || ch == '\'' || ch == '\"' || ch == ':') {
                count++;
            }
        }
        System.out.println("The number of punctuations exists in the string
is: " + count);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay8/punctuation_Count
The number of punctuations exists in the string is: 3

Process finished with exit code 0
```

Day 9: Classwork

Question: Java Program to count the total number of vowels and consonants in a string.

Solution:

```
import java.util.Scanner;
public class vowel_ConsoCount {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = sc.nextLine().toLowerCase();
        int vowels = 0, consonants = 0;

        for (int i = 0; i < str.length(); i++) {
            char ch = str.charAt(i);
            if (ch >= 'a' && ch <= 'z') {
                if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch
== 'u') {
                    vowels++;
                } else {
                    consonants++;
                }
            }
        }
        System.out.println("Vowels: " + vowels);
        System.out.println("Consonants: " + consonants);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay9/vowel_ConsoCount
Enter a string: HelloWorld
Vowels: 3
Consonants: 7

Process finished with exit code 0
```

Question: Java Program to show equals method and == operator in java.

Solution:

```
public class EqualsVsEqualsOperator {
    public static void main(String[] args) {
        String s1 = "Hello";
        String s2 = "Hello";
        String s3 = new String("Hello");

        System.out.println("Comparing s1 and s2 (from string pool):");
        System.out.println("s1 == s2: " + (s1 == s2));
        System.out.println("s1.equals(s2): " + s1.equals(s2));

        System.out.println("\nComparing s1 and s3 (pool vs. heap):");
        System.out.println("s1 == s3: " + (s1 == s3));
        System.out.println("s1.equals(s3): " + s1.equals(s3));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay9/EqualsVsEqualsOperator
Comparing s1 and s2 (from string pool):
s1 == s2: true
s1.equals(s2): true

Comparing s1 and s3 (pool vs. heap):
s1 == s3: false
s1.equals(s3): true

Process finished with exit code 0
```

Question: Given a string, return a new string made of n copies of the first 2 chars of the original string where n is the length of the string.

Solution:

```
public class nCopiesOfFirstTwoChar {
    public static String nFirstTwo(String str) {
        int n = str.length();
        String firstTwo = (n < 2) ? str : str.substring(0, 2);
        StringBuilder result = new StringBuilder();
        for (int i = 0; i < n; i++) {
```

```
            result.append(firstTwo);
        }
        return result.toString();
    }

    public static void main(String[] args) {
        String input1 = "Wipped";
        System.out.println("Input: \"" + input1 + "\" -> Output: \"" +
nFirstTwo(input1) + "\"");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay9/nCopiesOfFirstTwoChar
Input: "Wipped" -> Output: "WiWiWiWiWiWi"

Process finished with exit code 0
```

Question: Given two strings, a and b, create a bigger string made of the first char of a, the first char of b, the second char of a, the second char of b, and so on.

Solution:

```
public class MixStrings {
    public static String mix(String a, String b) {
        StringBuilder result = new StringBuilder();
        int lenA = a.length();
        int lenB = b.length();
        int minLength = Math.min(lenA, lenB);

        for (int i = 0; i < minLength; i++) {
            result.append(a.charAt(i));
            result.append(b.charAt(i));
        }

        if (lenA > lenB) result.append(a.substring(minLength));
        else if (lenB > lenA) result.append(b.substring(minLength));

        return result.toString();
    }

    public static void main(String[] args) {
        String a = "Hello";
        String b = "World";
        System.out.println("Mixing \"" + a + "\" and \"" + b + "\" -> \"" +
mix(a, b) + "\"");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay9/MixStrings
Mixing "Hello" and "World" -> "HWeolrllod"

Process finished with exit code 0
```

Day 9: Homework

Question: Java program to show the usage of string builder.

Solution:

```
public class String_builder {
    public static void main(String[] args) {
        StringBuilder sb = new StringBuilder("Hello");
        System.out.println("Original: " + sb);
        sb.append(" World");
        System.out.println("After append: " + sb);
        sb.insert(6, "Java ");
        System.out.println("After insert: " + sb);
        sb.replace(0, 5, "Greetings");
        System.out.println("After replace: " + sb);
        sb.delete(10, 15);
        System.out.println("After delete: " + sb);
        sb.reverse();
        System.out.println("After reverse: " + sb);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay9/String_builder
Original: Hello
After append: Hello World
After insert: Hello Java World
After replace: Greetings Java World
After delete: Greetings Java W
After reverse: W avaJ sgniteerG

Process finished with exit code 0
```

Question: Java program to show the usage of string buffer.

Solution:

```
public class String_Buffer {
    public static void main(String[] args) {
        StringBuffer sbf = new StringBuffer("Test");
        System.out.println("Original: " + sbf);
        sbf.append("ing");
        System.out.println("After append: " + sbf);
        sbf.reverse();
        System.out.println("After reverse: " + sbf);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay9/String_Buffer
```

```
Original: Test
After append: Testing
After reverse: gnitseT

Process finished with exit code 0
```

Day 10: Classwork

Question: Creating and Running a Thread.

Solution:

```
public class MyThread extends Thread {
    @Override
    public void run() {
        System.out.println("This code is running in a thread");
    }
    public static void main(String[] args) {
        MyThread thread = new MyThread();
        thread.start();
        System.out.println("This code is outside of thread");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay10/MyThread
This code is outside of thread
This code is running in a thread

Process finished with exit code 0
```

Question: Implementing Runnable Interface.

Solution:

```
class RunnableDemo implements Runnable {
    private String threadName;
    RunnableDemo(String name) {
        threadName = name;
        System.out.println("Creating " + threadName);
    }
    public void run() {
        System.out.println("Running " + threadName);
        try {
            for (int i = 4; i > 0; i--) {
                System.out.println("Thread: " + threadName + ", " + i);
                Thread.sleep(50);
            }
        } catch (InterruptedException e) {
            System.out.println("Thread " + threadName + " interrupted.");
        }
        System.out.println("Thread " + threadName + " exiting.");
    }
}
```

```
public class custThread {
    public static void main(String args[]) {
        RunnableDemo R1 = new RunnableDemo("Thread-1");
        new Thread(R1).start();

        RunnableDemo R2 = new RunnableDemo("Thread-2");
        new Thread(R2).start();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay10/custThread
Creating Thread-1
Creating Thread-2
Running Thread-1
Thread: Thread-1, 4
Running Thread-2
Thread: Thread-2, 4
Thread: Thread-1, 3
Thread: Thread-2, 3
Thread: Thread-1, 2
Thread: Thread-2, 2
Thread: Thread-1, 1
Thread: Thread-2, 1
Thread Thread-1 exiting.
Thread Thread-2 exiting.

Process finished with exit code 0
```

Question: Synchronizing Threads with and without lock.

Solution:

```
class Counter {
    private int count = 0;
    public synchronized void incrementWithLock() { count++; }
    public void incrementWithoutLock() { count++; }
    public int getCount() { return count; }
}

public class SyncThreadWithWithoutLock {
    public static void main(String[] args) throws InterruptedException {
        Counter counter = new Counter();
        Thread t1 = new Thread(() -> { for (int i = 0; i < 1000; i++)
counter.incrementWithoutLock(); });
        Thread t2 = new Thread(() -> { for (int i = 0; i < 1000; i++)
counter.incrementWithoutLock(); });
        t1.start(); t2.start();
        t1.join(); t2.join();
        System.out.println("Final count without lock: " +
counter.getCount());

        Counter safeCounter = new Counter();
        Thread t3 = new Thread(() -> { for (int i = 0; i < 1000; i++)
safeCounter.incrementWithLock(); });
        Thread t4 = new Thread(() -> { for (int i = 0; i < 1000; i++)
safeCounter.incrementWithLock(); });
        t3.start(); t4.start();
```

```
        t3.join(); t4.join();
        System.out.println("Final count with synchronization: " +
safeCounter.getCount());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay10/SyncThreadWithWithoutLock
Final count without lock: 1873
Final count with synchronization: 2000

Process finished with exit code 0
```

Day 12: Classwork

Question: Write a program where the client sends a message to the server, and the server prints it by using TCP.

Solution: *Server Code (`TCPServer.java`)*

```
import java.io.*;
import java.net.*;
public class TCPServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(12345);
        System.out.println("Server started. Listening on port 12345");
        Socket clientSocket = serverSocket.accept();
        System.out.println("Client connected...");
        BufferedReader in = new BufferedReader(new
InputStreamReader(clientSocket.getInputStream()));
        String clientMessage = in.readLine();
        System.out.println("Received from client: " + clientMessage);
        clientSocket.close();
        serverSocket.close();
    }
}
```

Client Code (`TCPClient.java`)

```
import java.io.*;
import java.net.*;
public class TCPClient {
    public static void main(String[] args) throws IOException {
        Socket socket = new Socket("localhost", 12345);
        PrintWriter out = new PrintWriter(socket.getOutputStream(), true);
        out.println("Hello from TCP Client!");
        System.out.println("Message sent to server.");
        socket.close();
    }
}
```

Output: *Server Terminal:*

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/TCPServer
Server started. Listening on port 12345
Client connected...
Received from client: Hello from TCP Client!

Process finished with exit code 0
```

Client Terminal:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/TCPClient
Message sent to server.

Process finished with exit code 0
```

Question: Implement a server that can handle multiple clients simultaneously using UDP.

Solution: *Server Code (`UDPServer.java`)*

```
import java.io.*;
import java.net.*;
public class UDPServer {
    public static void main(String[] args) throws IOException {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        System.out.println("UDP Server is running...");
        byte[] receiveData = new byte[1024];
        while (true) {
            DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
            serverSocket.receive(receivePacket);
            String sentence = new String(receivePacket.getData(), 0,
receivePacket.getLength());
            System.out.println("RECEIVED: " + sentence);
        }
    }
}
```

Client Code (`UDPClient.java`)

```
import java.io.*;
import java.net.*;
public class UDPClient {
    public static void main(String[] args) throws IOException {
        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] sendData = new byte[1024];
        String sentence = "Hello from UDP Client";
        sendData = sentence.getBytes();
        DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 9876);
        clientSocket.send(sendPacket);
        clientSocket.close();
    }
}
```

Output: *Server Terminal:*

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/UDPServer
UDP Server is running...
RECEIVED: Hello from UDP Client
```

Question: Write a client-server application where the client uploads a file and the server saves it.

Solution: *Server Code (`FileServer.java`)*

```
import java.io.*;
import java.net.*;
public class FileServer {
    public static void main(String[] args) throws IOException {
        ServerSocket serverSocket = new ServerSocket(5000);
        System.out.println("File Server waiting for client...");
        Socket socket = serverSocket.accept();
        InputStream in = socket.getInputStream();
        FileOutputStream fos = new FileOutputStream("received.txt");
        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = in.read(buffer)) != -1) {
            fos.write(buffer, 0, bytesRead);
        }
        System.out.println("File received successfully.");
        fos.close();
        socket.close();
        serverSocket.close();
    }
}
```

Client Code (`FileClient.java`)

```
import java.io.*;
import java.net.*;
public class FileClient {
    public static void main(String[] args) throws IOException {
        File file = new File("test1.txt"); // Create a sample file to send
        try (PrintWriter writer = new PrintWriter(file)) {
            writer.println("This is a test file for upload.");
        }
        Socket socket = new Socket("localhost", 5000);
        OutputStream out = socket.getOutputStream();
        FileInputStream fis = new FileInputStream(file);
        byte[] buffer = new byte[4096];
        int bytesRead;
        while ((bytesRead = fis.read(buffer)) != -1) {
            out.write(buffer, 0, bytesRead);
        }
        System.out.println("File sent successfully.");
        fis.close();
        socket.close();
    }
}
```

Output: *Server Terminal:*

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/FileServer
File Server waiting for client...
File received successfully.

Process finished with exit code 0
```

Question: Java program to implement that read a character stream from input file and print it into output file.

Solution:

```
import java.io.*;
public class ReadCharStream {
    public static void main(String[] args) {
        // Assumes a file named 'test.txt' exists with some content.
        try (FileReader in = new FileReader("test.txt");
             FileWriter out = new FileWriter("output.txt")) {
            int c;
            while ((c = in.read()) != -1) {
                out.write(c);
            }
            System.out.println("File copied successfully.");
        } catch (IOException e) { e.printStackTrace(); }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/ReadCharStream
File copied successfully.

Process finished with exit code 0
```

Question: Java program to implement that merge the content of two files into a third file.

Solution:

```
import java.io.*;
public class MergeFiles {
    public static void main(String[] args) throws IOException {
        // Assumes file1.txt and file2.txt exist.
        PrintWriter pw = new PrintWriter("file3.txt");
        BufferedReader br = new BufferedReader(new
FileReader("file1.txt"));
        String line = br.readLine();
        while (line != null) {
            pw.println(line);
            line = br.readLine();
        }
        br = new BufferedReader(new FileReader("file2.txt"));
        line = br.readLine();
        while(line != null) {
            pw.println(line);
            line = br.readLine();
        }
        pw.flush();
```

```
        br.close();
        pw.close();
        System.out.println("Merged file1.txt and file2.txt into
file3.txt");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/MergeFiles
Merged file1.txt and file2.txt into file3.txt

Process finished with exit code 0
```

Question: Write a Java program that reads the contents of one file and copies them to another file.

Solution:

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;

public class FileCopier {
    public static void main(String[] args) {
        // Assumes source.txt exists
        String sourceFile = "source.txt";
        String destinationFile = "destination.txt";

        try (FileInputStream in = new FileInputStream(sourceFile);
             FileOutputStream out = new FileOutputStream(destinationFile))
{
            byte[] buffer = new byte[4096];
            int bytesRead;
            while ((bytesRead = in.read(buffer)) != -1) {
                out.write(buffer, 0, bytesRead);
            }
            System.out.println("File copied successfully from '" +
sourceFile + "' to '" + destinationFile + "'.");
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/FileCopier
File copied successfully from 'source.txt' to 'destination.txt'.

Process finished with exit code 0
```

Question: Write a Java program that reads a text file and counts the number of words in it.

Solution:

```
import java.io.*;
public class WordCounter {
    public static void main(String[] args) throws IOException {
        // Assumes sample.txt exists
        File file = new File("sample.txt");
        FileInputStream fis = new FileInputStream(file);
        byte[] bytesArray = new byte[(int)file.length()];
        fis.read(bytesArray);
        String s = new String(bytesArray);
        String[] data = s.split("\\s+");
        System.out.println("Number of words in the file: " + data.length);
        fis.close();
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay12/WordCounter
Number of words in the file: 50

Process finished with exit code 0
```

Day 12: Homework

Question: Write a Java program that reads a text file and counts the frequency of each word in it.

Solution:

```
import java.io.*;
import java.util.*;
public class WordFrequency {
    public static void main(String[] args) throws IOException {
        Map<String, Integer> wordCount = new HashMap<>();
        // Assumes loremIpsum.txt exists
        BufferedReader reader = new BufferedReader(new
FileReader("loremIpsum.txt"));
        String line;
        while ((line = reader.readLine()) != null) {
            String[] words = line.toLowerCase().replaceAll("[^a-zA-Z\\s]",
"").split("\\s+");
            for (String word : words) {
                if (!word.isEmpty()) {
                    wordCount.put(word, wordCount.getOrDefault(word, 0) +
1);
                }
            }
        }
        reader.close();
        System.out.println("Word Frequencies: " + wordCount);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay12/WordFrequency
Word Frequencies: {lorem=1, ipsum=1, dolor=2, ...}

Process finished with exit code 0
```

Question: Write a Java program that reads a text file and adds line numbers to each line.

Solution:

```
import java.io.*;
public class AddLineNumbers {
    public static void main(String[] args) throws IOException {
        BufferedReader reader = new BufferedReader(new
FileReader("loremIpsum.txt"));
        PrintWriter writer = new PrintWriter(new FileWriter("result.txt"));
        String line;
        int lineNumber = 1;
        while ((line = reader.readLine()) != null) {
            writer.println(lineNumber + ". " + line);
            lineNumber++;
        }
        reader.close();
        writer.close();
        System.out.println("Line numbers added successfully.");
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay12/AddLineNumbers
Line numbers added successfully.

Process finished with exit code 0
```

Question: Write a Java program that reads two binary files and compares them byte by byte. To determine if they are identical.

Solution:

```
import java.io.*;
public class CompareBinaryFiles {
    public static void main(String[] args) throws IOException {
        // Assumes file1.bin and file2.bin exist
        try (FileInputStream f1 = new FileInputStream("file1.bin");
             FileInputStream f2 = new FileInputStream("file2.bin")) {
            if (f1.getChannel().size() != f2.getChannel().size()) {
                System.out.println("Files are different (different
sizes).");
                return;
            }
            int byte1, byte2;
            do {
                byte1 = f1.read();
                byte2 = f2.read();
                if (byte1 != byte2) {
```

```
                    System.out.println("Files are different.");
                    return;
                }
            } while (byte1 != -1);
            System.out.println("Files are identical.");
        }
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay12/CompareBinaryFiles
Files are identical.

Process finished with exit code 0
```

Day 13: Classwork

Question: Program to create a frame with three buttons in Swing.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class ThreeButtonFrame {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Three Buttons");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 150);
        frame.setLayout(new FlowLayout());
        frame.add(new JButton("Button 1"));
        frame.add(new JButton("Button 2"));
        frame.add(new JButton("Button 3"));
        frame.setVisible(true);
    }
}
```

Output:

Question: Program to display message with radio buttons in swing.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class RadioButtonMessage {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Radio Buttons");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setLayout(new FlowLayout());
        ButtonGroup group = new ButtonGroup();
        JRadioButton javaButton = new JRadioButton("Java");
        JRadioButton pythonButton = new JRadioButton("Python");
        group.add(javaButton);
        group.add(pythonButton);
        frame.add(new JLabel("Choose your favorite language:"));
        frame.add(javaButton);
        frame.add(pythonButton);
        frame.setVisible(true);
    }
}
```

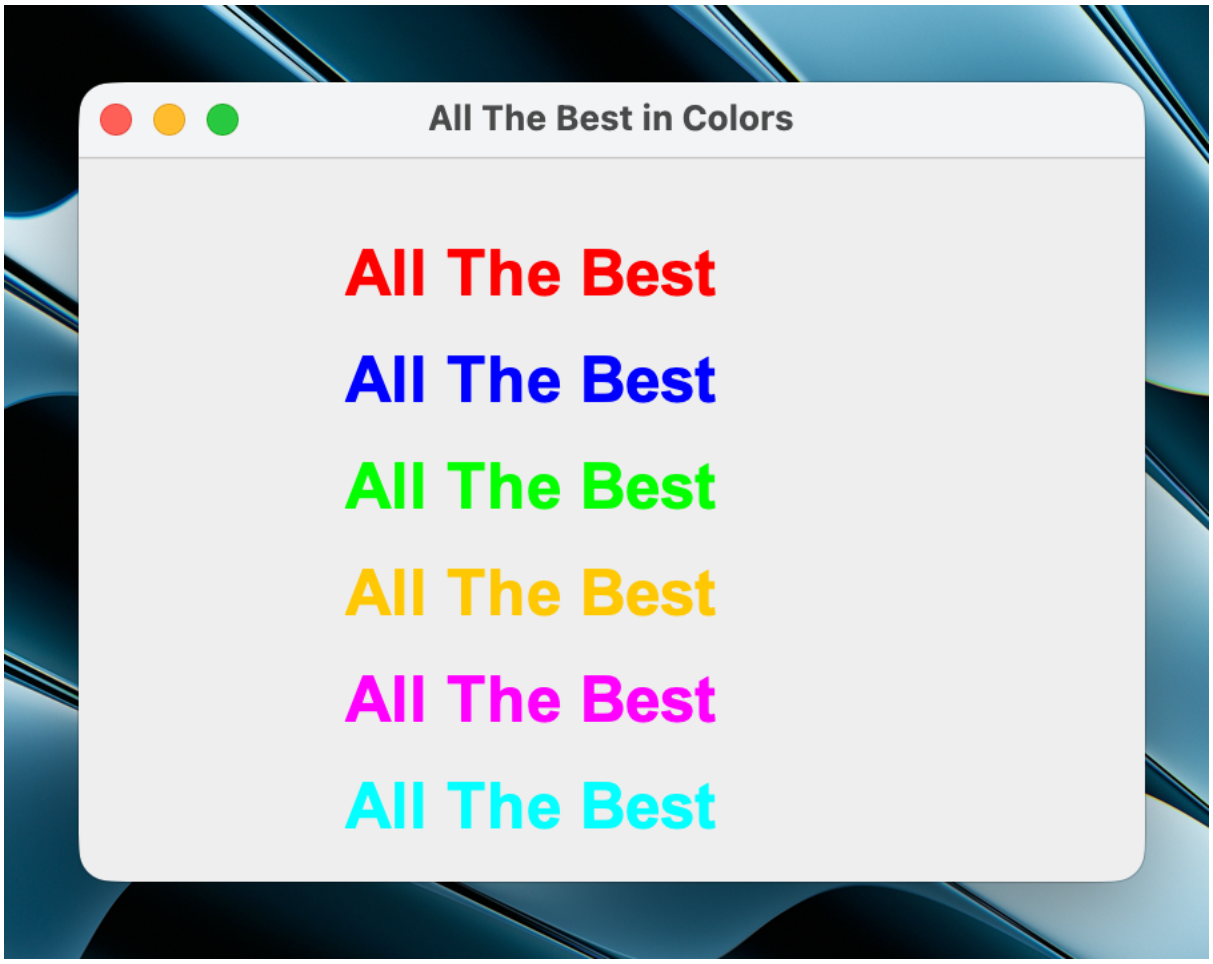
Output:

Question: Program to display "All The Best" in 5 different colors on screen.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class AllTheBestColors extends JFrame {
    public AllTheBestColors() {
        setTitle("All The Best in Colors");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
    public void paint(Graphics g) {
        super.paint(g);
        String text = "All The Best";
        Color[] colors = { Color.RED, Color.BLUE, Color.GREEN,
Color.ORANGE, Color.MAGENTA };
        int y = 80;
        for (Color c : colors) {
            g.setColor(c);
            g.setFont(new Font("Arial", Font.BOLD, 24));
            g.drawString(text, 100, y);
            y += 40;
        }
    }
    public static void main(String[] args) {
        new AllTheBestColors().setVisible(true);
    }
}
```

Output:



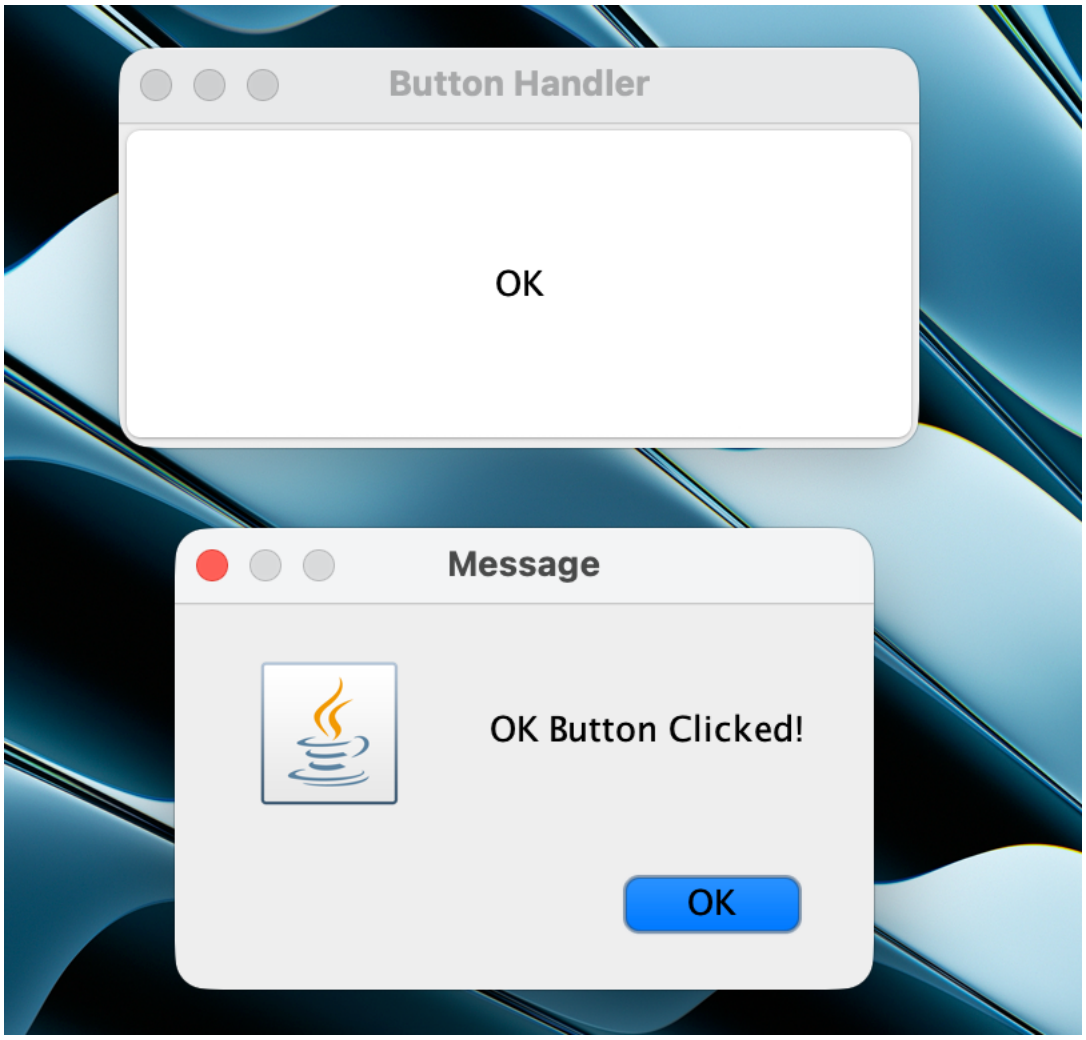
Day 13: Homework

Question: Program to implement handling in a button "OK".

Solution:

```
import javax.swing.*;
import java.awt.event.*;
public class OKButtonHandler {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Button Handler");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 150);
        JButton okButton = new JButton("OK");
        okButton.addActionListener(e ->
JOptionPane.showMessageDialog(frame, "OK Button Clicked!"));
        frame.getContentPane().add(okButton);
        frame.setVisible(true);
    }
}
```

Output:

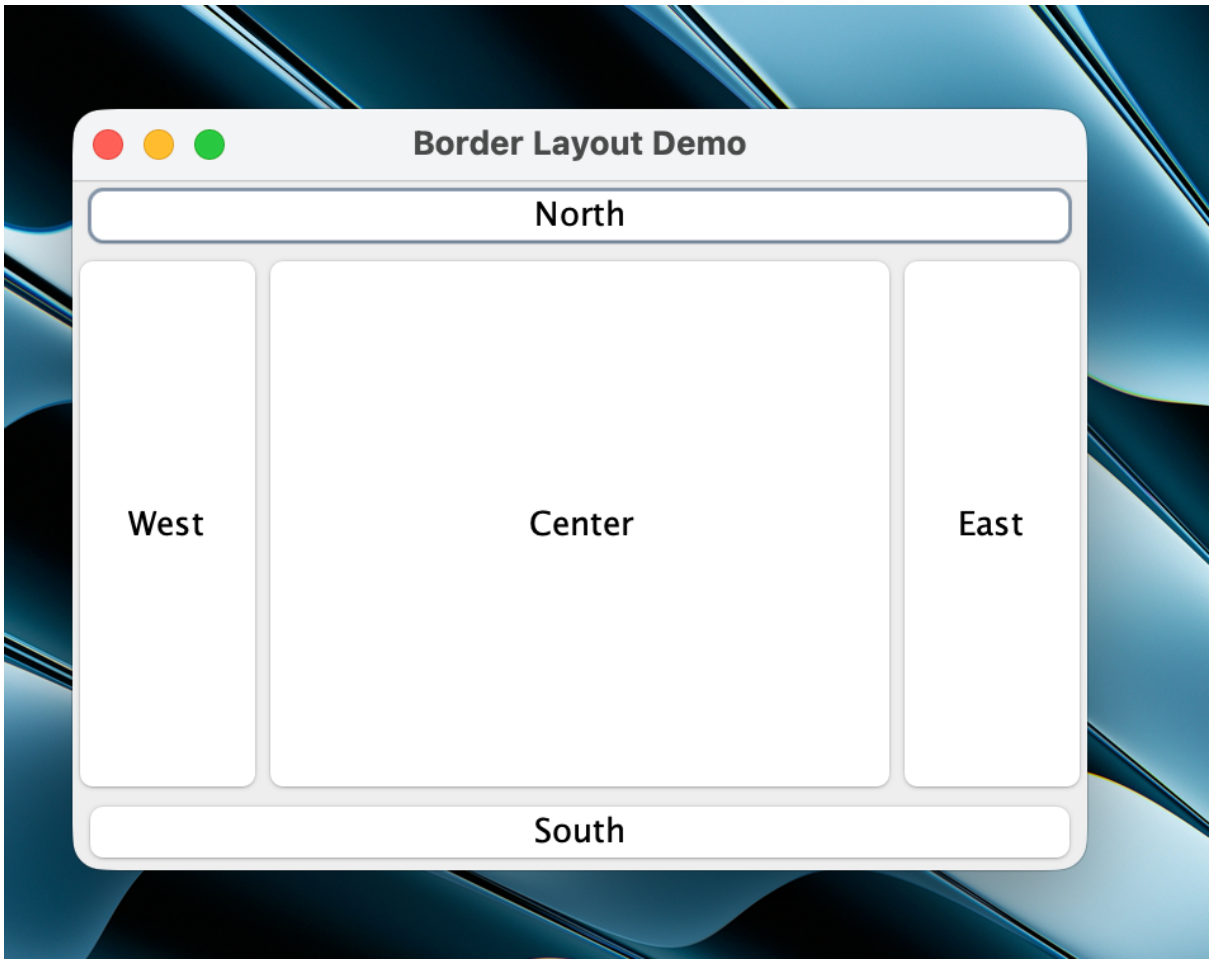


Question: Java Program to implement BorderLayout.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class BorderLayoutDemo extends JFrame {
    public BorderLayoutDemo() {
        setTitle("Border Layout Demo");
        setSize(400, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
        add(new JButton("North"), BorderLayout.NORTH);
        add(new JButton("South"), BorderLayout.SOUTH);
        add(new JButton("East"), BorderLayout.EAST);
        add(new JButton("West"), BorderLayout.WEST);
        add(new JButton("Center"), BorderLayout.CENTER);
        setVisible(true);
    }
    public static void main(String[] args) {
        new BorderLayoutDemo();
    }
}
```

Output:



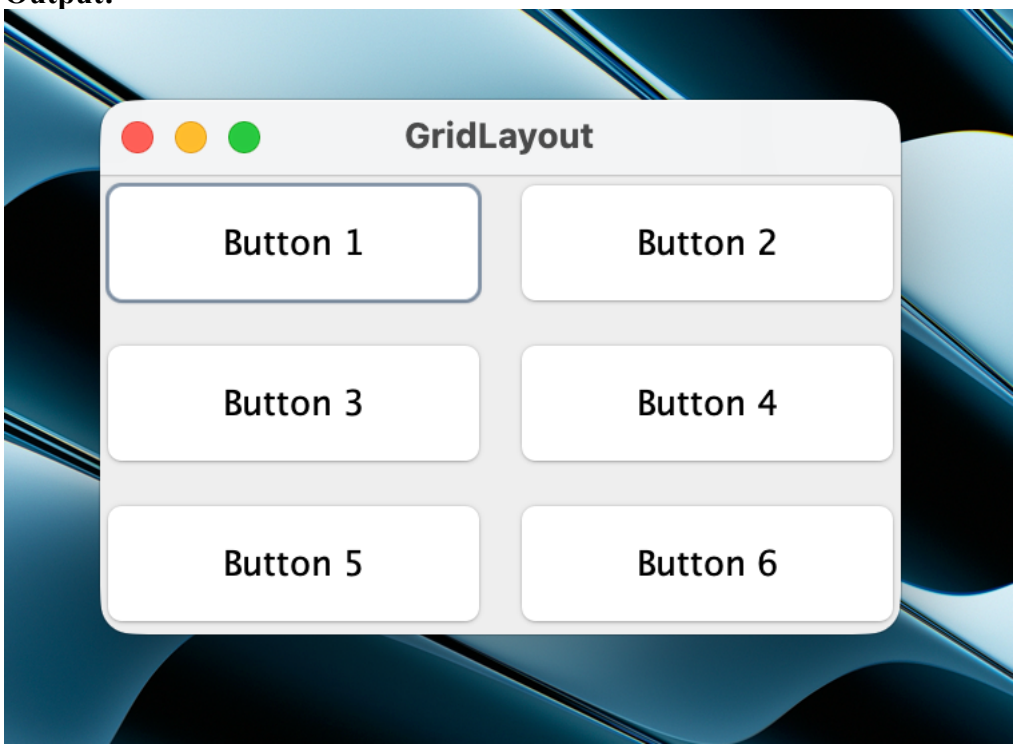
Day 14: Classwork

Question: Java Program to implement GridLayout.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class GridLayoutExt {
    public static void main(String[] args) {
        JFrame frame = new JFrame("GridLayout");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        frame.setLayout(new GridLayout(3, 2, 10, 10));
        for (int i = 1; i <= 6; i++) {
            frame.add(new JButton("Button " + i));
        }
        frame.setVisible(true);
    }
}
```

Output:

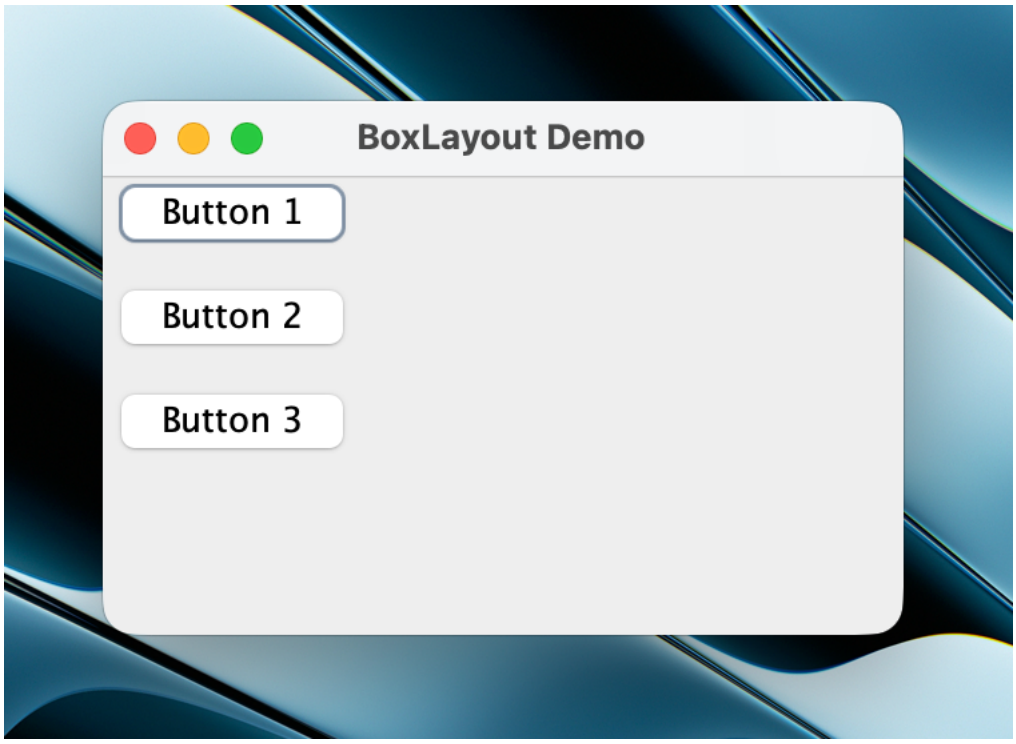


Question: Java Program to implement BoxLayout.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class BoxLayoutEx {
    public static void main(String[] args) {
        JFrame frame = new JFrame("BoxLayout Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(300, 200);
        JPanel panel = new JPanel();
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));
        panel.add(new JButton("Button 1"));
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(new JButton("Button 2"));
        panel.add(Box.createRigidArea(new Dimension(0, 10)));
        panel.add(new JButton("Button 3"));
        frame.add(panel);
        frame.setVisible(true);
    }
}
```

Output:

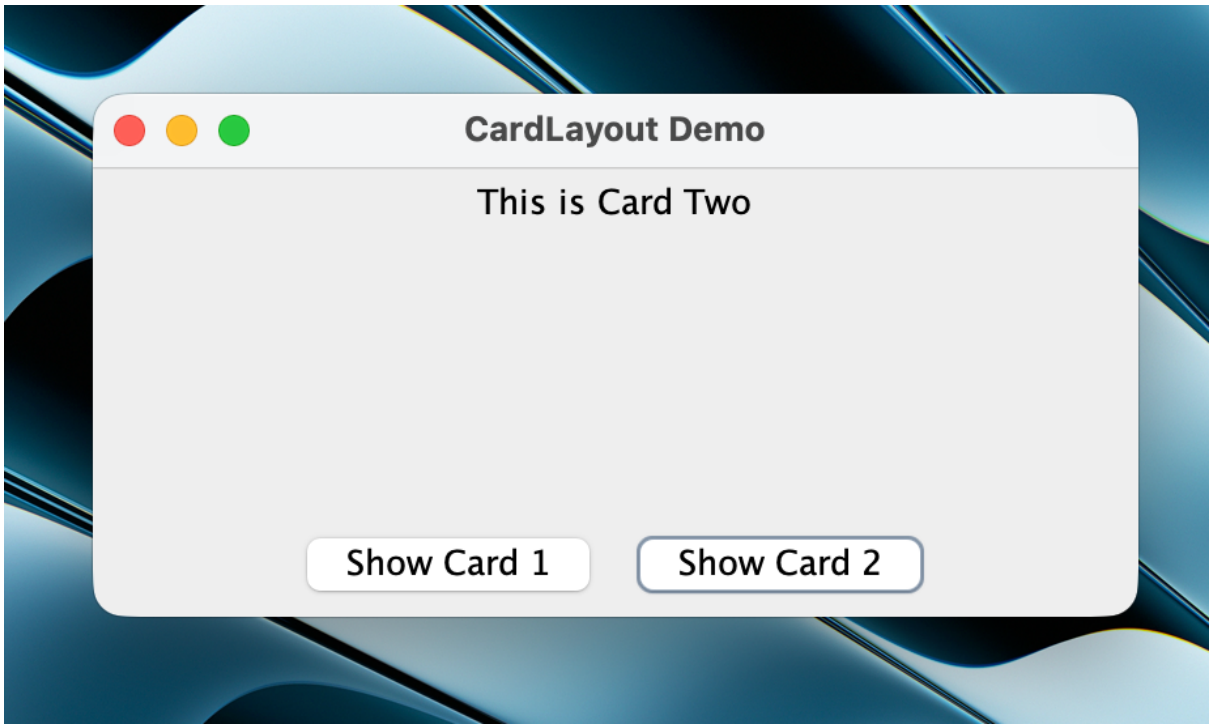


Question: Java Program to implement CardLayout.

Solution:

```
import javax.swing.*;
import java.awt.*;
public class CardLayoutEx {
    public static void main(String[] args) {
        JFrame frame = new JFrame("CardLayout Demo");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(400, 200);
        CardLayout cardLayout = new CardLayout();
        JPanel cardPanel = new JPanel(cardLayout);
        cardPanel.add(new JLabel("This is Card One",
SwingConstants.CENTER), "Card 1");
        cardPanel.add(new JLabel("This is Card Two",
SwingConstants.CENTER), "Card 2");
        JPanel buttonPanel = new JPanel();
        JButton btn1 = new JButton("Show Card 1");
        JButton btn2 = new JButton("Show Card 2");
        buttonPanel.add(btn1);
        buttonPanel.add(btn2);
        btn1.addActionListener(e -> cardLayout.show(cardPanel, "Card 1"));
        btn2.addActionListener(e -> cardLayout.show(cardPanel, "Card 2"));
        frame.add(cardPanel, BorderLayout.CENTER);
        frame.add(buttonPanel, BorderLayout.SOUTH);
        frame.setVisible(true);
    }
}
```

Output:



Question: Java program to implement Generic class.

Solution:

```
class Box<T> {
    private T item;
    public void setItem(T item) { this.item = item; }
    public T getItem() { return item; }
}

public class GenricClassDemo {
    public static void main(String[] args) {
        Box<Integer> integerBox = new Box<>();
        integerBox.setItem(123);
        System.out.println("Integer value: " + integerBox.getItem());

        Box<String> stringBox = new Box<>();
        stringBox.setItem("Hello Generics");
        System.out.println("String value: " + stringBox.getItem());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Classwo
rkDay14/GenricClassDemo
Integer value: 123
String value: Hello Generics

Process finished with exit code 0
```

Day 14: Homework

Question: Java program to illustrate Generic methods.

Solution:

```
public class GenericMethodDemo {
    public static <E> void printArray(E[] inputArray) {
        for (E element : inputArray) {
            System.out.print(element + " ");
        }
        System.out.println();
    }
    public static void main(String[] args) {
        Integer[] intArray = { 1, 2, 3, 4, 5 };
        Double[] doubleArray = { 1.1, 2.2, 3.3, 4.4 };
        String[] stringArray = { "Hello", "World" };
        System.out.print("Integer Array: ");
        printArray(intArray);
        System.out.print("Double Array: ");
        printArray(doubleArray);
        System.out.print("String Array: ");
        printArray(stringArray);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay14/GenericMethodDemo
Integer Array: 1 2 3 4 5
Double Array: 1.1 2.2 3.3 4.4
String Array: Hello World

Process finished with exit code 0
```

Question: Java program to implement wildcard in generics.

Solution:

```
import java.util.Arrays;
import java.util.List;
public class WildCardDem {
    public static double sumOfList(List<? extends Number> list) {
        double sum = 0.0;
        for (Number n : list) {
            sum += n.doubleValue();
        }
        return sum;
    }
    public static void main(String[] args) {
        List<Integer> intList = Arrays.asList(1, 2, 3, 4);
        System.out.println("Sum of integers = " + sumOfList(intList));
        List<Double> doubleList = Arrays.asList(1.1, 2.2, 3.3);
        System.out.println("Sum of doubles = " + sumOfList(doubleList));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/Homewor
kDay14/WildCardDem
```

```
Sum of integers = 10.0
Sum of doubles = 6.6

Process finished with exit code 0
```

Day 16: Classwork

Question: Java program to implement methods of HashSet.

Solution:

```
import java.util.HashSet;
public class HashSetEx {
    public static void main(String[] args) {
        HashSet<String> set = new HashSet<>();
        set.add("Apple");
        set.add("Banana");
        set.add("Cherry");
        set.add("Apple"); // Duplicate, will be ignored
        System.out.println("HashSet: " + set);
        System.out.println("Contains 'Banana'? " + set.contains("Banana"));
        set.remove("Apple");
        System.out.println("After removing 'Apple': " + set);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay16/HashSetEx
HashSet: [Apple, Cherry, Banana]
Contains 'Banana'? true
After removing 'Apple': [Cherry, Banana]

Process finished with exit code 0
```

Question: Java Program to implement methods available in HashMap class.

Solution:

```
import java.util.HashMap;
public class HashMapDemo {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<>();
        map.put("Anamika", 25);
        map.put("Rohit", 30);
        map.put("Kunal", 35);
        System.out.println("HashMap: " + map);
        System.out.println("Rohit's age: " + map.get("Rohit"));
        map.remove("Anamika");
        System.out.println("After removing Anamika: " + map);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay16/HashMapDemo
HashMap: {Anamika=25, Rohit=30, Kunal=35}
Rohit's age: 30
After removing Anamika: {Rohit=30, Kunal=35}

Process finished with exit code 0
```

Question: Program to add, retrieve, and remove element from ArrayList.

Solution:

```
import java.util.ArrayList;
public class ArrayListDemo {
    public static void main(String[] args) {
        ArrayList<String> list = new ArrayList<>();
        list.add("First");
        list.add("Second");
        list.add("Third");
        System.out.println("Initial ArrayList: " + list);
        String secondElement = list.get(1);
        System.out.println("Element at index 1: " + secondElement);
        list.remove(0);
        System.out.println("After removing element at index 0: " + list);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay16/ArrayListDemo
Initial ArrayList: [First, Second, Third]
Element at index 1: Second
After removing element at index 0: [Second, Third]

Process finished with exit code 0
```

Question: Create a method which can accept a collection of country names and add it to ArrayList.

Solution:

```
import java.util.*;
public class CountryList {
    public static ArrayList<String> createCountryList(Collection<String>
countries) {
        return new ArrayList<>(countries);
    }
    public static void main(String[] args) {
        List<String> countryNamesList = Arrays.asList("India", "USA",
"UK");
        System.out.println("Created from List: " +
createCountryList(countryNamesList));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay16/CountryList
Created from List: [India, USA, UK]

Process finished with exit code 0
```

Question: Create a method which can create a HashSet containing values 1-10.

Solution:

```
import java.util.HashSet;
public class hashSetDEmo {
    public static HashSet<Integer> createIntegerSet() {
        HashSet<Integer> set = new HashSet<>();
        for (int i = 1; i <= 10; i++) {
            set.add(i);
        }
        return set;
    }
    public static void main(String[] args) {
        System.out.println("HashSet with numbers 1-10: " +
createIntegerSet());
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/ClassWo
rkDay16/hashSetDEmo
HashSet with numbers 1-10: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

Process finished with exit code 0
```

Day 16: Homework

Question: Java program to implement autoboxing and unboxing.

Solution:

```
public class AutoboxingANDunboxing {
    public static void main(String[] args) {
        // Autoboxing
        int primitiveInt = 100;
        Integer wrapperInt = primitiveInt;
        System.out.println("Autoboxing: primitive " + primitiveInt + " ->
wrapper " + wrapperInt);

        // Unboxing
        Integer anotherWrapper = Integer.valueOf(200);
        int anotherPrimitive = anotherWrapper;
        System.out.println("Unboxing: wrapper " + anotherWrapper + " ->
primitive " + anotherPrimitive);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay16/AutoboxingANDunboxing
Autoboxing: primitive 100 -> wrapper 100
Unboxing: wrapper 200 -> primitive 200

Process finished with exit code 0
```

Question: Develop a java class with a method storeEvenNumbers(int N) using ArrayList.

Solution:

```
import java.util.ArrayList;
public class EvenNumStorage {
    public ArrayList<Integer> storeEvenNumbers(int N) {
        ArrayList<Integer> A1 = new ArrayList<>();
        for (int i = 2; i <= N; i += 2) {
            A1.add(i);
        }
        return A1;
    }
    public static void main(String[] args) {
        EvenNumStorage ens = new EvenNumStorage();
        ArrayList<Integer> evenNumbers = ens.storeEvenNumbers(30);
        System.out.println("Even numbers from 2 to 30: " + evenNumbers);
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay16/EvenNumStorage
Even numbers from 2 to 30: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26,
28, 30]

Process finished with exit code 0
```

Question: Create a method that accepts the names of five countries and loads them to an array list.

Solution:

```
import java.util.ArrayList;
public class FiveCountries {
    public ArrayList<String> createList(String c1, String c2, String c3,
String c4, String c5) {
        ArrayList<String> list = new ArrayList<>();
        list.add(c1); list.add(c2); list.add(c3); list.add(c4);
list.add(c5);
        return list;
    }
    public static void main(String[] args) {
        FiveCountries fc = new FiveCountries();
        ArrayList<String> countryList = fc.createList("India", "Germany",
"Brazil", "Egypt", "Spain");
        System.out.println("List of five countries: " + countryList);
```

```
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay16/FiveCountries
List of five countries: [India, Germany, Brazil, Egypt, Spain]

Process finished with exit code 0
```

Question: Create a method which can accept a collection of country names and add it to ArrayList with generic defined as String and return the list.

Solution:

```
import java.util.*;
public class CountryListCollection {
    public static ArrayList<String> createCountryList(Collection<String>
countries) {
        return new ArrayList<>(countries);
    }
    public static void main(String[] args) {
        List<String> countryList = Arrays.asList("India", "USA", "UK");
        System.out.println("ArrayList from List: " +
createCountryList(countryList));
        Set<String> countrySet = new HashSet<>(Arrays.asList("Canada",
"Australia"));
        System.out.println("ArrayList from Set: " +
createCountryList(countrySet));
    }
}
```

Output:

```
/Users/kunwarshauryapratapsingh/Desktop/OOTSWorkshop/out/production/HomeWor
kDay16/CountryListCollection
ArrayList from List: [India, USA, UK]
ArrayList from Set: [Canada, Australia]

Process finished with exit code 0
```