

1.1) Program to remove all repeated elements from an array

```
public class Main{
    public static void main(String [] args){
        int []abc={2,5,6,4,2,5};
        int b=abc.length;
        for (int i=0;i<b;i++){
            for (int k=i+1;k<b;k++){
                if (abc[i]==abc[k]){
                    abc[i]=-1;
                }
            }
        }
        for (int a=0;a<b;a++){
            if(abc[a]!=-1){
                System.out.print(abc[a]+" ");
            }
        }
    }
}
```

1.2) Write a Java program to find the common elements between two arrays of integers.

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

        // Input array A
        System.out.print("Enter the size of Array A: ");
        int sizeA = scanner.nextInt();
        int[] arrayA = new int[sizeA];
        System.out.println("Enter the elements of Array A:");
        for (int i = 0; i < sizeA; i++) {
            arrayA[i] = scanner.nextInt();
        }

        // Input array B
        System.out.print("Enter the size of Array B: ");
        int sizeB = scanner.nextInt();
        int[] arrayB = new int[sizeB];
        System.out.println("Enter the elements of Array B:");
        for (int i = 0; i < sizeB; i++) {
            arrayB[i] = scanner.nextInt();
        }

        // Find common elements
        System.out.println("Common elements between Array A and Array B:");
        for (int i = 0; i < sizeA; i++) {
            for (int j = 0; j < sizeB; j++) {
```

```

        if (arrayA[i] == arrayB[j]) {
            System.out.println(arrayA[i]);
            break;
        }
    }
}
}
}
}

```

2.1) Java Program to Count Number of Duplicate Words in String

```

import java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String s = sc.nextLine();
        s=s.toLowerCase();
        String[] words = s.split(" ");

        int d=0;
        for(int i=0;i<words.length;i++){
            int count=0;
            for(int j=i+1;j<words.length;j++){
                if(words[i].equals(words[j])){
                    count++;
                    words[j]="0";
                }
            }

            if(count>0 && !words[i].equals("0")){
                System.out.println(words[i]);
                d++;
            }
        }

        System.out.println(d);
    }
}

```

2.2) How to Check if the String Contains 'e' in umbrella

```

public class Main{

    public static void main(String [] args){
        String a="chtan";
        boolean c=a.contains("e");
        if(c){
            System.out.println("ffffff");
        }
        else{
            System.out.println("zzzzzzzz");
        }
    }
}

```

```

    }
}
}

```

3.

3.1) Java Program to Reverse a String.

```

import java.util.Scanner;
public class Main{
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("enter string");
        String x = scanner.nextLine();
        StringBuilder y = new StringBuilder();
        for (int i = x.length() - 1; i >= 0; i--) {
            y.append(x.charAt(i));
        }
        System.out.println(y);
    }
}

```

3.2) Write a Java program to check that String is palindrome or not.

```

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

        System.out.println("Enter a string:");
        String input = scanner.nextLine();
        int left = 0;
        int right = input.length() - 1;
        int x=0;
        while (left < right) {
            if (input.charAt(left) != input.charAt(right)) {
                x=1;
            }
            left++;
            right--;
        }
        if (x==0) {
            System.out.println("The string is a palindrome.");
        } else {
            System.out.println("The string is not a palindrome.");
        }
    }
}

```

4. A Company manufactures Vehicles, which could be a Helicopter, a Car, or a Train depending on the customer's demand. Each Vehicle instance has a method called move, which prints on the console the nature of movement of the vehicle. For example, the Helicopter Flies in Air, the Car Drives on Road and the Train Runs on Track. Write a program that accepts input from the user on the kind of vehicle the user wants to order, and the system should print out nature of movement. Implement all Java coding best practices to implement this program.

```

import java.util.*;

class vehicle{
    void move(){
        System.out.println("runs");
    }
}

class helicopter extends vehicle{
    void move(){
        System.out.println("the Helicopter Flies in Air");
    }
}

class car extends vehicle{
    void move(){
        System.out.println("the Car Drives on Road ");
    }
}

class train extends vehicle{
    void move(){
        System.out.println("the Train Runs on Track");
    }
}

public class lab_4_vehicle {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Which kind of vehicle u want to order?");
        System.out.println("Enter\n 1 for Helicopter \n 2 for car \n 3 for train");
        vehicle vehicle;
        int n = sc.nextInt();
        if(n==1){
            // vehicle = new helicopter();
            // vehicle.move();
            vehicle v = new helicopter();
            v.move();
        }
        else if(n==2){
            vehicle = new car();
            vehicle.move();
        }
        else if(n==3){
            vehicle = new train();
            vehicle.move();
        }
        else{
            System.out.println("Enter right choice");
        }
    }
}

```

5. We have to calculate the percentage of marks obtained in three subjects (each out of 100) by student A and in four subjects (each out of 100) by student B. Create an abstract class 'Marks' with an abstract method 'getPercentage'. It is inherited by two other classes 'A' and 'B' each having a method with the same name which returns the percentage of the students. The constructor of student A takes the marks in three subjects as its parameters and the marks in four subjects as its parameters for student B. Create an object for each of the two classes and print the percentage of marks for both the students.

```
package labexam;
import java.util.*;

abstract class Marks{
    abstract double getPercentage();
}

class A extends Marks{
    double p;

    public A(int s1, int s2, int s3){
        p = (s1+s2+s3)/3;
    }

    double getPercentage(){
        return p;
    }
}

class B extends Marks{
    double p;

    public B(int s1, int s2, int s3,int s4){
        p = (s1+s2+s3+s4)/4;
    }

    double getPercentage(){
        return p;
    }
}

public class lab_5_marks {
    public static void main(String[] args) {

        Marks stud1 = new A(70,80,90);
        double p1 = stud1.getPercentage();
        System.out.println("Percentage of student 1 = "+p1);

        Marks stud2 = new B(70,80,90,99);
        double p2 = stud2.getPercentage();
        System.out.println("Percentage of student 2 = "+p2);
    }
}
```

6. Write the following code in your editor below:

A class named Arithmetic with a method named add that takes 2 integers as parameters and returns an integer denoting their sum.

A class named Adder that inherits from a superclass named Arithmetic. The main method in the Tester class should print the following: SAMPLE O/P:My superclass is: Arithmetic

42 13 20

```
import java.util.*;

class Arithmetic{
    int add(int n1, int n2){
        int sum = n1+n2;
        return sum;
    }
}

class Adder extends Arithmetic{

}

public class lab_6_arithmetic {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        int n2 = sc.nextInt();
        Adder add1 = new Adder();
        int sum = add1.add(n1,n2);
        System.out.println("My superclass is: Arithmetic");
        System.out.println(sum+" " + n1 + " " + n2);
    }
}
```

7. You are required to compute the power of a number by implementing a calculator. Create a class My Calculator which consists of a single method long power (int, int). This method takes two integers n and p, as parameters and finds (n)^p. If either or is negative, then the method must throw an exception which says "n or p should not be negative". Also, if both are zero, then the method must throw an exception which says "n or p should not be negative".

```
import java.util.Scanner;

class MyCalculator{
    public long power(int n,int p) throws Exception{
        if(n==0||p==0){
            throw new Exception ("zzzzzzzz");
        }
        else if(n<1||p<1){
            throw new Exception("ffffffffff");
        }
        else{
            return (long) Math.pow(n, p);
        }
    }
}
```

```

    }
}

public class Main{
    public static void main(String [] args){
        Scanner scanner=new Scanner(System.in);
        int n=scanner.nextInt();
        int p=scanner.nextInt();
        MyCalculator abc=new MyCalculator();
        try{
            long ans=abc.power( n, p);
            System.out.println(ans);

        }
        catch(Exception e){
            System.out.println(e.getMessage());
        }
    }
}

```

8. You are given a phone book that consists of people's names and their phone number. After that you will be given some person's name as query. For each query, print the phone number of that person. Use HashMap to implement it. The first line will have an integer denoting the number of entries in the phone book. Each entry consists of two lines: a name and the corresponding phone number. After these, there will be some queries. Each query will contain a person's name. Read the queries until end-of-file.

Constraints:

A person's name consists of only lower-case English letters and it may be in the format 'first-name last-name' or in the format 'first-name'. Each phone number has exactly 8 digits without any leading zeros. For each case, print "Not found" if the person has no entry in the phone book. Otherwise, print the person's name and phone number.

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

```

public class lab_8_phonebook {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of entries in the phone book: ");
        int n = sc.nextInt();
        sc.nextLine(); // Consume the newline character

        // Create a HashMap to store the phone book entries
        Map<String, String> phoneBook = new HashMap<>();

        // Read the phone book entries
        for (int i = 0; i < n; i++) {
            System.out.print("Enter the name: ");
            String name = sc.nextLine();
            System.out.print("Enter the phone number: ");
            String phoneNumber = sc.nextLine();

```

```

        phoneBook.put(name, phoneNumber);
    }

    System.out.print("Enter the name to search (or type 'exit' to quit): ");
    while (sc.hasNextLine()) {
        String query = sc.nextLine();
        if (query.equals("exit")) {
            break;
        }

        String phoneNumber = phoneBook.get(query);

        if (phoneNumber != null) {
            System.out.println(query + " " + phoneNumber);
        } else {
            System.out.println("Not found");
        }

    }

    sc.close();
}
}

```

9. Gui-chat frame

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class lab_9_gui {
    public static void main(String[] args) {

        JFrame jf = new JFrame();
        jf.setSize(500, 500);
        JPanel panel = new JPanel();

        JMenuBar menuBar = new JMenuBar();
        JMenu file = new JMenu("File");
        JMenu help = new JMenu("Help");

        menuBar.add(file);
        menuBar.add(help);

        JMenuItem open = new JMenuItem("Open");
        JMenuItem saveAs = new JMenuItem("Save as");

        file.add(open);
        file.add(saveAs);

        JTextArea textArea = new JTextArea();

        JTextField text = new JTextField(15);
        JButton reset = new JButton("Clear");
    }
}

```



```

JButton send = new JButton("Send");
JLabel label = new JLabel("Please Enter Text");
panel.add(label);
panel.add(text);
panel.add(send);
panel.add(reset);

// Action listener for the send button
send.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String inputText = text.getText();
        textArea.append(inputText + "\n");
        text.setText("");
    }
});

// Action listener for the clear button
reset.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        textArea.setText("");
    }
});

jf.getContentPane().add(menuBar, BorderLayout.NORTH);
jf.getContentPane().add(panel, BorderLayout.SOUTH);
jf.getContentPane().add(textArea, BorderLayout.CENTER);

jf.setVisible(true);
}
}

```

10. Gui number addition

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Main extends JFrame {
    private JTextField num1Field;
    private JTextField num2Field;
    private JLabel resultLabel;

    public Main() {
        setTitle("Simple Calculator");
        setSize(300, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);

        // Create components
        num1Field = new JTextField(10);

```

```

num2Field = new JTextField(10);
JButton addButton = new JButton("Add");
resultLabel = new JLabel("Result:");

// Create panel and add components
JPanel panel = new JPanel();
panel.setLayout(new GridLayout(4, 1));
panel.add(new JLabel("Number 1:"));
panel.add(num1Field);
panel.add(new JLabel("Number 2:"));
panel.add(num2Field);
panel.add(addButton);
panel.add(resultLabel);

// Add panel to the frame
add(panel);

// Register ActionListener for the addButton
addButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        try {
            int num1 = Integer.parseInt(num1Field.getText());
            int num2 = Integer.parseInt(num2Field.getText());
            int result = num1 + num2;
            resultLabel.setText("Result: " + result);
        } catch (NumberFormatException ex) {
            resultLabel.setText("Invalid input");
        }
    }
});
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            new Main().setVisible(true);
        }
    });
}
}

```

11.2) Write a java program to check that given number is prime or not.

```

public class lab_11_prime {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int count=0;

        for(int i=2;i<=n/2;i++){
            if(n%i==0){
                count++;
            }
        }
    }
}

```

```

    }

    if(count>0){
        System.out.println("No");
    }
    else{
        System.out.println("Yes");
    }
}
}
}

```

12. Write a Java program to display the pattern like a diamond.
Input number of rows (half of the diamond) :7 Expected Output :

```

*
***
*****
*****
*****
*****
*****
*****
*****
*****
*****
***
*

```

```

import java.util.Scanner;
public class Main {
    public static void main(String[] args)
    {
        int i,j,inp;
        System.out.print("Input number of rows (half of the diamond) : ");
        Scanner in = new Scanner(System.in);
        inp = in.nextInt();
        for(i=0;i<=inp;i++){
            for(j=1;j<=inp-i;j++){
                System.out.print(" ");
            }
            for(j=1;j<=2*i-1;j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }
        for(i=inp-1;i>=1;i--){
            for(j=1;j<=inp-i;j++){
                System.out.print(" ");
            }
            for(j=1;j<=2*i-1;j++){
                System.out.print("*");
            }
            System.out.print("\n");
        }
    }
}

```

```

    }
}
}

```

13.1) Write Java Program to find the transpose of a given matrix .

```

import java.util.*;

public class lab_13_transpose {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter size of matrix");
        int m = sc.nextInt();
        int n = sc.nextInt();

        int mat[][] = new int[m][n];

        System.out.println("Enter Elements");
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                mat[i][j]=sc.nextInt();
            }
        }

        System.out.println("Matrix before transpose : ");
        for(int i=0;i<m;i++){
            for(int j=0;j<n;j++){
                System.out.print(mat[i][j]+" ");
            }
            System.out.println();
        }

        System.out.println("Matrix after transpose : ");
        for(int i=0;i<n;i++){
            for(int j=0;j<m;j++){
                System.out.print(mat[j][i]+" ");
            }
            System.out.println();
        }
    }
}

```

13.2) Write Java Program to find the number of the words in the given text file.

```

import java.util.*;
import java.io.*;

public class lab_13_wordcount {
    public static void main(String[] args) {
        try {
            File file = new File("C:\\Users\\abhis\\Basic1\\src\\textfile.txt");

```

```

Scanner sc = new Scanner(file);
int wordCount = 0;

while (sc.hasNext()) {
    sc.next();
    wordCount++;
}

System.out.println(wordCount);
} catch (FileNotFoundException e) {
    System.out.println("File not found.");
}
}
}

```

14. Gui calculator

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class Calculator extends JFrame implements ActionListener {
    private JTextField textField;
    private JButton[] buttons;
    private String[] buttonLabels = {
        "7", "8", "9", "/",
        "4", "5", "6", "*",
        "1", "2", "3", "-",
        "0", "C", "=", "+"
    };
};

private double firstNumber;
private String operator;

public Calculator() {
    setTitle("Calculator");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setResizable(false);
    setSize(300, 400);
    setLocationRelativeTo(null);

    textField = new JTextField();
    textField.setPreferredSize(new Dimension(300, 50));
    textField.setFont(new Font("Arial", Font.PLAIN, 24));
    textField.setHorizontalAlignment(JTextField.RIGHT);
    textField.setEditable(false);
    add(textField, BorderLayout.NORTH);

    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(4, 4, 10, 10));

```

```

        buttons = new JButton[buttonLabels.length];
        for (int i = 0; i < buttonLabels.length; i++) {
            buttons[i] = new JButton(buttonLabels[i]);
            buttons[i].setFont(new Font("Arial", Font.PLAIN, 24));
            buttons[i].addActionListener(this);
            panel.add(buttons[i]);
        }

        add(panel, BorderLayout.CENTER);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String command = e.getActionCommand();

        if (command.matches("[0-9]")) {
            textField.setText(textField.getText() + command);
        } else if (command.equals("=")) {
            double secondNumber = Double.parseDouble(textField.getText());
            double result = calculate(firstNumber, secondNumber, operator);
            textField.setText(String.valueOf(result));
        } else if (command.equals("C")) {
            textField.setText("");
        } else {
            firstNumber = Double.parseDouble(textField.getText());
            operator = command;
            textField.setText("");
        }
    }
}

private double calculate(double firstNumber, double secondNumber, String operator) {
    double result = 0;

    switch (operator) {
        case "+":
            result = firstNumber + secondNumber;
            break;
        case "-":
            result = firstNumber - secondNumber;
            break;
        case "*":
            result = firstNumber * secondNumber;
            break;
        case "/":
            result = firstNumber / secondNumber;
            break;
    }

    return result;
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            Calculator calculator = new Calculator();

```

```

        calculator.setVisible(true);
    }
});
}
}

```

15. Write a Java Program to iterate ArrayList using for-loop, iterator, and advance for-loop. Insert 3 Array List. Input 20 30 40 Output:

iterator Loop:

```

20
30
40

```

Advanced For Loop:

```

20
30
40

```

For Loop:

```

20
30
40

```

```

import java.util.*;

```

```

public class lab_15_arraylist {
    public static void main(String[] args) {
        ArrayList<Integer> a = new ArrayList<Integer>();
        a.add(20);
        a.add(30);
        a.add(40);

        System.out.println("iterator Loop:");
        Iterator<Integer> iterator = a.iterator();
        while(iterator.hasNext()){
            System.out.println(iterator.next());
        }

        System.out.println("Advanced For Loop:");
        for(int i:a){
            System.out.println(i);
        }

        System.out.println("For Loop:");
        for (int i = 0; i < a.size(); i++) {
            System.out.println(a.get(i));
        }

    }
}

```

16. Write a Java Program to count the number of words in a string using HashMap.Output:

Input :Enter String: "This this is is done by Saket Saket";

{Saket=2, by=1, this=1, This=1, is=2, done=1}

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

```
public class lab_16_hashmap {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter String: ");
        String input = sc.nextLine();

        String[] words = input.split(" ");

        HashMap<String,Integer> wordCount = new HashMap<>();

        for(int i=0;i<words.length;i++){
            String word = words[i];
            if (wordCount.containsKey(word)) {
                int count = wordCount.get(word);
                wordCount.put(word, count + 1);
            } else {
                wordCount.put(word, 1);
            }
        }
        System.out.println(wordCount);
    }
}
```

17. Write a program to read 10 string from console and then print the sorted strings on console (Use String Class).2) combine two string 3)reverse first string and display it .

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

```
public class lab_17_string {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String[] strings = new String[10];

        for(int i=0;i<10;i++){
            System.out.print("Enter string " + (i + 1) + ": ");
            strings[i] = sc.nextLine();
        }

        Arrays.sort(strings);

        System.out.println("Sorted strings:");
        for(int i=0;i<10;i++){
            System.out.println(strings[i]);
        }

        //2
        String cString = strings[0] + strings[1];
        System.out.println("Combined string: " + cString);
    }
}
```



```

//3
String s1 = strings[0];
String rs1 = "";
for(int i=s1.length()-1;i>=0;i--){
    rs1 = rs1 + s1.charAt(i);
}
System.out.println("Reversed string: " + rs1);
}
}

```

18. Write a program to implement following inheritance. Accept data for 5 persons and display the name of employee having salary greater than 5000.

Class Name: Person

Member variables:

Name, age

Class Name: Employee

Member variables:

Designation, salary

```

class Person{
    String name;
    int age;
    Person(int age,String name){
        this.name=name;
        this.age=age;
    }
}
class Employee extends Person{
    String post;
    int salary;

    Employee(String post,String name,int age,int salary){
        super(age,name);
        this.post=post;
        this.salary=salary;
        if (salary>5000){
            System.out.println(name);
        }
    }
}
public class Main{
    public static void main (String [] args){
        Employee emp = new Employee("Developer","Naman",19,45000);
        Employee emp1 = new Employee("Manager","Karan",25,4000);
        Employee emp2 = new Employee("Accountant","Akash",26,5000);
        Employee emp3 = new Employee("Developer","Yash",22,6000);
        Employee emp4 = new Employee("Data Scientist","Sahil",23,2000);
    }
}

```

19. Implementing "Multiple Inheritance". Create a two interfaces Account containing methods set() and display() And interface Person containing methods store() and disp(). Derive a class Customer from Person and Account. Accept the name, account number, balance and display all the information related to account along with the interest.

```
package labexam;
import java.util.*;

interface Account{
    void set();
    void display();
}

interface Person{
    void store();
    void disp();
}

class Customer implements Person, Account{
    String name;
    int accountNumber;
    double balance;

    public void set(){
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter name: ");
        name = sc.nextLine();
        System.out.print("Enter account number: ");
        accountNumber = sc.nextInt();
        System.out.print("Enter balance: ");
        balance = sc.nextDouble();
    }

    public void display(){
        System.out.println("Account Information:");
        System.out.println("Name: " + name);
        System.out.println("Account Number: " + accountNumber);
        System.out.println("Balance: " + balance);
        System.out.println("Interest: " + (balance * 0.05)); // Assuming 5% interest rate
    }

    public void store(){

    }

    public void disp() {

    }
}

public class lab_19_multipleinherit {
    public static void main(String[] args) {
        Customer customer = new Customer();
        customer.set();
    }
}
```

```
        customer.display();
    }
}
```

20. "Write a program, to implement the following hierarchy. Displays information of each class the rectangle represents the classes. The classes Movie and MusicVideo inherits all the members of the class Video Tape."

```
class VideoTape {
    public String title;
    public int length;
    public boolean available;

    public VideoTape(String title, int length, boolean available) {
        this.title = title;
        this.length = length;
        this.available = available;
    }

    public void show() {
        System.out.println("Video Tape Information:");
        System.out.println("Title: " + title);
        System.out.println("Length: " + length + " minutes");
        System.out.println("Available: " + (available ? "Yes" : "No"));
    }
}

class Movie extends VideoTape {
    public String director;
    public double rating;

    public Movie(String title, int length, boolean available, String director, double rating) {
        super(title, length, available);
        this.director = director;
        this.rating = rating;
    }

    public void show() {
        super.show();
        System.out.println("Director: " + director);
        System.out.println("Rating: " + rating);
    }
}

class MusicVideo extends VideoTape {
    public String artist;
    public String category;

    public MusicVideo(String title, int length, boolean available, String artist, String category) {
        super(title, length, available);
        this.artist = artist;
        this.category = category;
    }
}
```

```

    public void show() {
        super.show();
        System.out.println("Artist: " + artist);
        System.out.println("Category: " + category);
    }
}

```

```

public class lab_20_inheritance {
    public static void main(String[] args) {
        VideoTape videoTape = new VideoTape("Generic Video", 120, true);
        Movie movie = new Movie("The Matrix", 150, true, "The Wachowskis", 8.7);
        MusicVideo musicVideo = new MusicVideo("Shape of You", 180, false, "Ed Sheeran", "Pop");

        System.out.println("Video Tape Information:");
        videoTape.show();
        System.out.println();

        System.out.println("Movie Information:");
        movie.show();
        System.out.println();

        System.out.println("Music Video Information:");
        musicVideo.show();
    }
}

```

21. Write a Java program to create a class called "Student" with a name, grade, and courses attributes, and methods to add and remove courses.

```

package labexam;
import java.util.*;

class Student{
    public String name;
    public int grade;
    List<String> courses;

    public Student(String name,int grade){
        this.name = name;
        this.grade = grade;
        this.courses = new ArrayList<>();
    }

    public String getName(){
        return name;
    }

    public int getGrade(){
        return grade;
    }

    public List<String> getCourses(){

```

```

        return courses;
    }

    public void add(String c){
        courses.add(c);
    }

    public void remove(String c){
        courses.remove(c);
    }
}

public class lab_21_student {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter name");
        String name = sc.nextLine();

        System.out.println("Enter grades");
        int grade = sc.nextInt();

        System.out.println("How many courses u want to enter");
        int n = sc.nextInt();
        sc.nextLine();

        Student stud = new Student(name, grade);

        System.out.println("Enter courses name");
        for(int i=0;i<n;i++){
            stud.add(sc.nextLine());
        }

        System.out.println("Student Name: " + stud.getName());
        System.out.println("Grade: " + stud.getGrade());
        System.out.println("Courses: " + stud.getCourses());

        System.out.println("Which course u want to remove");
        stud.remove(sc.nextLine());

        System.out.println("Updated courses: "+stud.getCourses());

    }
}

```

22. Write a Java program to create a class known as Person with methods called getFirstName() and getLastName(). Create a subclass called Employee that adds a new method named getEmployeeId() and overrides the getLastName() method to include the employee's job title.

```

import java.util.*;

class Person{
    String fName;
    String lName;

```

```

public Person(String fName, String lName){
    this.fName = fName;
    this.lName = lName;
}

public String getFirstName(){
    return fName;
}

public String getLastName(){
    return lName;
}
}

class Employee extends Person{
    int employeeId;
    String jobTitle;

    public Employee(String fName,String lName,int employeeId,String jobTitle){
        super(fName,lName);
        this.employeeId = employeeId;
        this.jobTitle = jobTitle;
    }

    public int getEmployeeId(){
        return employeeId;
    }

    @Override
    public String getLastName(){
        return lName + "(" + jobTitle + ")";
    }
}

public class lab_22_person {
    public static void main(String[] args) {
        Person person = new Person("Abhishek", "Wankhade");
        Employee employee = new Employee("Abhishek", "W", 1345, "Manager");

        System.out.println("Person Name: " + person.getFirstName() + " " + person.getLastName());
        System.out.println("Employee Name: " + employee.getFirstName() + " " + employee.getLastName());

        System.out.println("Employee ID: " + employee.getEmployeeId());
    }
}

```

23. Write a Java program to find the length of the longest consecutive elements sequence from an unsorted array of integers.
Sample array: [49, 1, 3, 200, 2, 4, 70, 5]
The longest consecutive elements sequence is [1, 2, 3, 4, 5], therefore the program will return its length 5.

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

```

public class lab_23_array {

    public static void main(String[] args) {
        int[] nums = {49, 1, 3, 200, 2, 4, 70, 5};

        Arrays.sort(nums);
        int count1=1;
        int lcount=1;

        for(int i=1;i<nums.length;i++){
            if(nums[i]!=nums[i-1]){
                if(nums[i]==nums[i-1]+1){
                    count1++;
                }
                else{
                    if(count1>lcount){
                        lcount=count1;
                    }
                    count1=1;
                }
            }
        }

        System.out.println(lcount);
    }
}

```

24. Create a class Student with attributes roll no, name, age and course. Initialize values through parameterized constructor. If age of student is not in between 15 and 21 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

```

import java.util.*;

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

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

class Student{
    int rollNo;
    String name;
    int age;
    String course;

    public Student(int rollNo, String name, int age, String course) throws AgeNotWithinRangeException, Na

```

```

meNotValidException{
    if(age<15 && age>21){
        throw new AgeNotWithinRangeException("Age is not within the valid range (15-21).");
    }

    if(!name.matches("^[a-zA-Z\\s]+$")){
        throw new NameNotValidException("Name contains numbers or special symbols.");
    }

    this.rollNo = rollNo;
    this.name = name;
    this.age = age;
    this.course = course;
}

}

public class lab_24_student {
    public static void main(String[] args) {
        try{
            Student stud1 = new Student(65,"Abhishek",21,"java");
            System.out.println("Student 1 created successfully.");
            Student stud2 = new Student(2, "Jane123", 20, "Medicine");
            System.out.println("Student 2 created successfully.");
        }
        catch(NameNotValidException e){
            System.out.println(e.getMessage());
        }
        catch(AgeNotWithinRangeException e){
            System.out.println(e.getMessage());
        }
    }
}

```

25.Gui all the best

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
class Color_Demo extends Frame
{
    Label lbl1, lbl2, lbl3, lbl4, lbl5;
    public Color_Demo()
    {
        lbl1 = new Label("All The Best");
        lbl1.setForeground(Color.red);
        add(lbl1);
        lbl2 = new Label("All The Best");
        lbl2.setForeground(Color.magenta);
        add(lbl2);
        lbl3 = new Label("All The Best");
        lbl3.setForeground(Color.blue);
    }
}

```



```

        add(lbl3);
        lbl4 = new Label("All The Best");
        lbl4.setForeground(Color.green);
        add(lbl4);
        lbl5 = new Label("All The Best");
        lbl5.setForeground(Color.cyan);
        add(lbl5);
        setVisible(true);
        setSize(400, 300);
        setLayout(new FlowLayout());
        setBackground(Color.gray);
    }
    public void paint(Graphics g)
    {
        g.setColor(Color.magenta);
        g.drawString("All The Best",100,100);
        g.setColor(Color.cyan);
        g.drawString("All The Best",150,150);
        g.setColor(Color.red);
        g.drawString("All The Best",200,200);
        g.setColor(Color.black);
        g.drawString("All The Best",250,250);
    }
    public static void main(String[] args)
    {
        new Color_Demo();
    }
}

```