National Academy of Science and Technology (NAST)

Accredited by University Grants Commission (UGC), Nepal (2022)

(Affiliated to Pokhara University)

Uttar Behadi-04, Dhangadhi, Kailali

A

Lab Report on

Object Oriented Programming using Java

Submitted To

Department of Computer Application

National Academy of Science and Technology

Submitted By:
Lila Sagar Ghimire (23530011)

Roll No: 11

Program: BCA

Semester: Third

CONTENTS

Date:2081/08/18

Lab 1

1.1 WAP to display hello World

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

Procedure to Generate Output:

1.Save the file HelloWorld.java

2. Compile the HelloWorld.java file:

Command: javac HelloWorld.java

3.Run the HelloWorld.class file

command: java HelloWorld

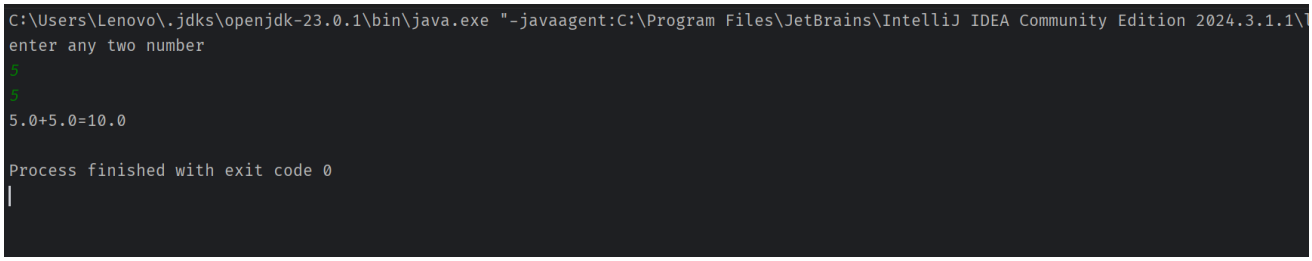
Output:

```
C:\Users\Lenovo\Downloads>javac HelloWorld.java

C:\Users\Lenovo\Downloads>java HelloWorld.java
Hello world
```

1.2 WAP to Add two number

```
import java.util.Scanner;
public class Add{
	public static void main(String... args){
		Scanner in=new Scanner(System.in);
		System.out.println("enter any two number");
		double a=in.nextDouble();
		double b=in.nextDouble();
		System.out.println(a+"+"+b +"="+(a+b));
	}
}
```

Output:



Date:2081/08/19

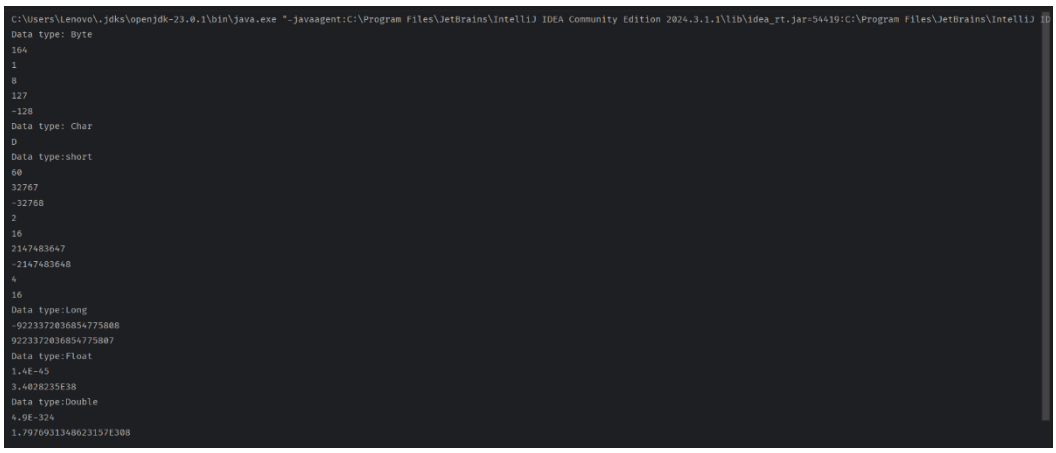
Lab 2

2.1.WAP to demonstrate Data Type

```
public class DataType {
    public static void main(String[] args) {
        //byte
        System.out.println("Data type: Byte");
        byte b='a';
        byte byt=67;
        System.out.println(b+byt);
        System.out.println(Byte.BYTES);
        System.out.println(Byte.SIZE);
        System.out.println(Byte.MAX_VALUE);
        System.out.println(Byte.MIN_VALUE);
        //char
        System.out.println("Data type: Char");
        char c='D';
        System.out.println(c);
        //short
        System.out.println("Data type:short");
        short st=60;
        System.out.println(st);
        System.out.println(Short.MAX_VALUE);
        System.out.println(Short.MIN_VALUE);
        System.out.println(Short.BYTES);
```

```
        System.out.println(Short.SIZE);
        //int
        System.out.println(Integer.MAX_VALUE);
        System.out.println(Integer.MIN_VALUE);
        System.out.println(Integer.BYTES);
        System.out.println(Short.SIZE);
        //long
          System.out.println("Data type:Long");
            System.out.println(Long.MIN_VALUE);
            System.out.println(Long.MAX_VALUE);
        //Float
          System.out.println("Data type:Float");
            System.out.println(Float.MIN_VALUE);
            System.out.println(Float.MAX_VALUE);
        //Double
          System.out.println("Data type:Double");
            System.out.println(Double.MIN_VALUE);
            System.out.println(Double.MAX_VALUE);
    }
}
```

Output:



Date:2081/09/12

Lab 3

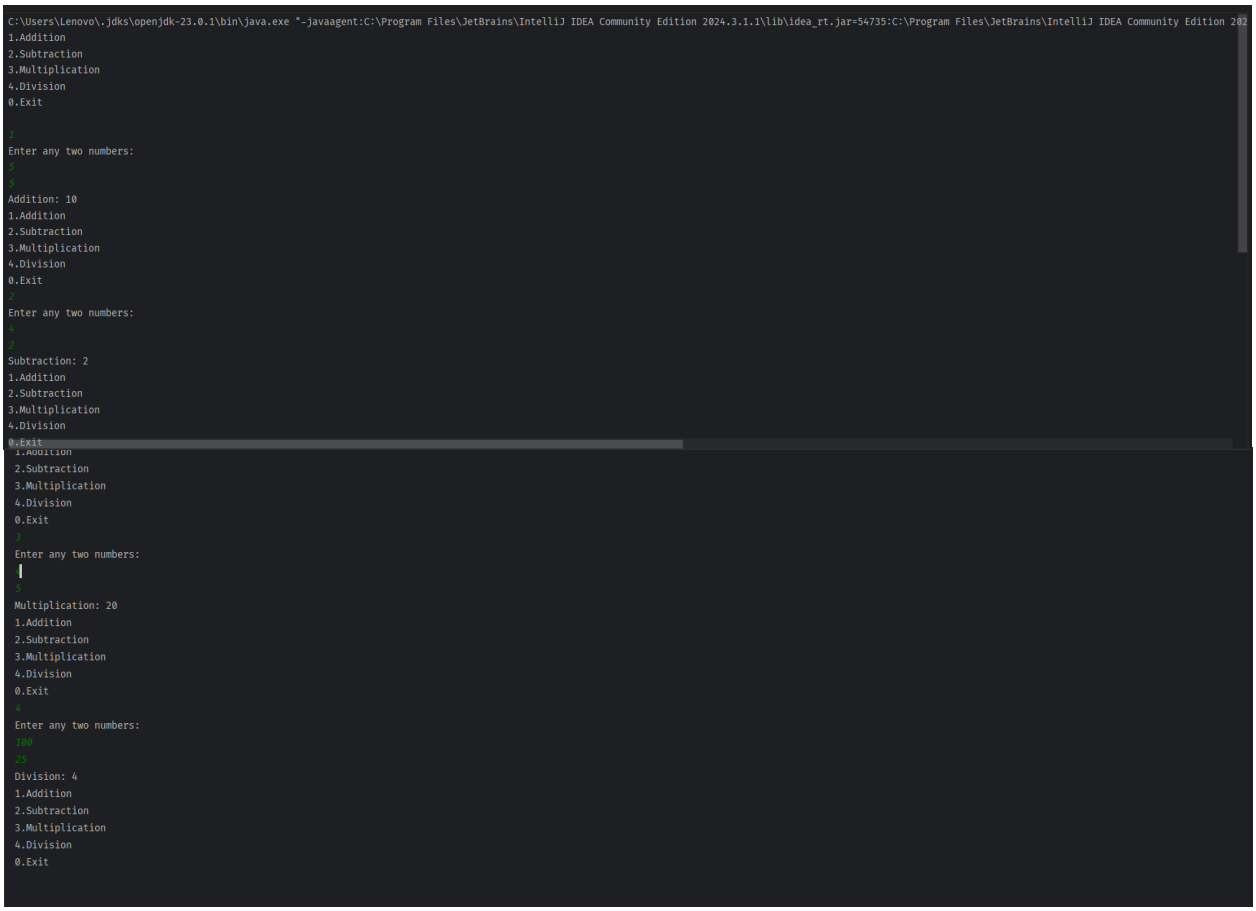
3.1.WAP using switch case where user input 1 for addition 2 for subtraction 3 for multiplication 4 for division and if user hit any other option show invalid message.

```
import java.util.Scanner;
public class Variable {
    static int firstNumber, secondNumber;
    static Scanner in;
    public static void main(String[] args) {
        in = new Scanner(System.in);
        while (true) {
            System.out.println("1.Addition");
            System.out.println("2.Subtraction");
            System.out.println("3.Multiplication");
            System.out.println("4.Division");
            System.out.println("0.exit");
            int choice = in.nextInt();
            System.out.println("Enter any two number");
            firstNumber = in.nextInt();
            secondNumber = in.nextInt();
            switch (choice) {
                case 1:
                    add();
                    break;
```

```
            case 2:
                sub();
                break;
            case 3:
                mul();
                break;
            case 4:
                div();
                break;
            default:
                System.out.println("INVALID");
        }
        if (choice == 0) {
            break;
        }
    }
}
public static void add() {
    System.out.println(“Addition”+(firstNumber + secondNumber));
}
public static void sub() {
    System.out.println(“”Subtraction”+(firstNumber - secondNumber));
}
public static void mul() {
    System.out.println(“Multiplication”+(firstNumber * secondNumber));
```

```
    }

    public static void div() {
        System.out.println("Division"+(firstNumber / secondNumber));
    }
}
```

Output:



Date:2081/09/18

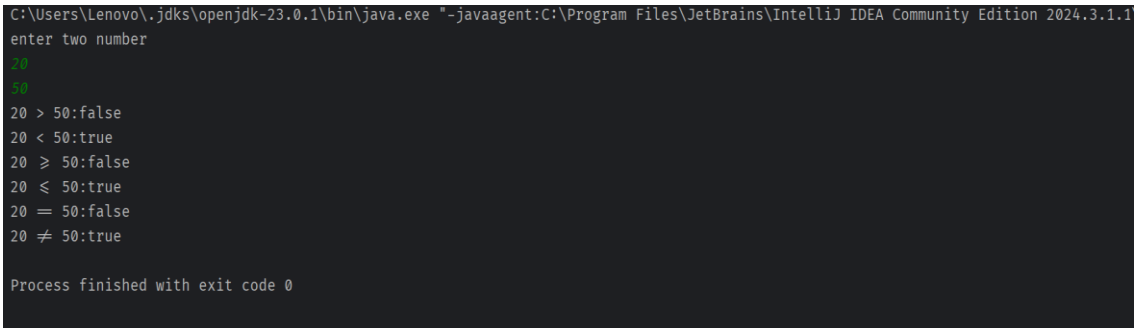
Lab 4

4.1. WAP to illustrate relational operators.

```
import java.util.Scanner;
public class RelationalOperator {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("enter two number");
        int first = in.nextInt();
        int second = in.nextInt();
        boolean r = first > second;
        System.out.println(first + " > " + second + ":" + r);
        r = first < second;
        System.out.println(first + " < " + second + ":" + r);
        r = first >= second;
        System.out.println(first + " >= " + second + ":" + r);
        r = first <= second;
        System.out.println(first + " <= " + second + ":" + r);
        r = first == second;
        System.out.println(first + " == " + second + ":" + r);
        r = first != second;
        System.out.println(first + " != " + second + ":" + r);
        in.close();
         }
```

}

Output:



4.2. WAP to illustrate conditional operator

```
import java.util.Scanner;
public class ConditionalOperator {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("enter two number");
        int first = in.nextInt();
        int second = in.nextInt();
        int r = first > second ? first : second;
        System.out.println("result " + r);
        in.close();
    }
}
```

Output:

4.3.

Program to illustrate Ternary Operator and pre and post increment/decrement operator.

```
public class TernaryDemo {
    public static void main(String[] args) {
        int a, b, c; //a=0,b=0,c=0
        a = 10; //a=10
        b = ++a; // a=11, b=11
        c = a++; // c=11, a=12
        System.out.println(a + b + c); //34 (12+11+11)
        c++; //c=12
        b = --c; // b=11, c=11
        a = --b; //a=10, b=10
        System.out.println(--a); //8 (9)
        b=a>b?a:b; //9>10 b=10
        System.out.println(b--); //print 10, b=9
        System.out.println(++c); //c=12
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\l
34
9
10
12

Process finished with exit code 0
```

Date:2081/09/21

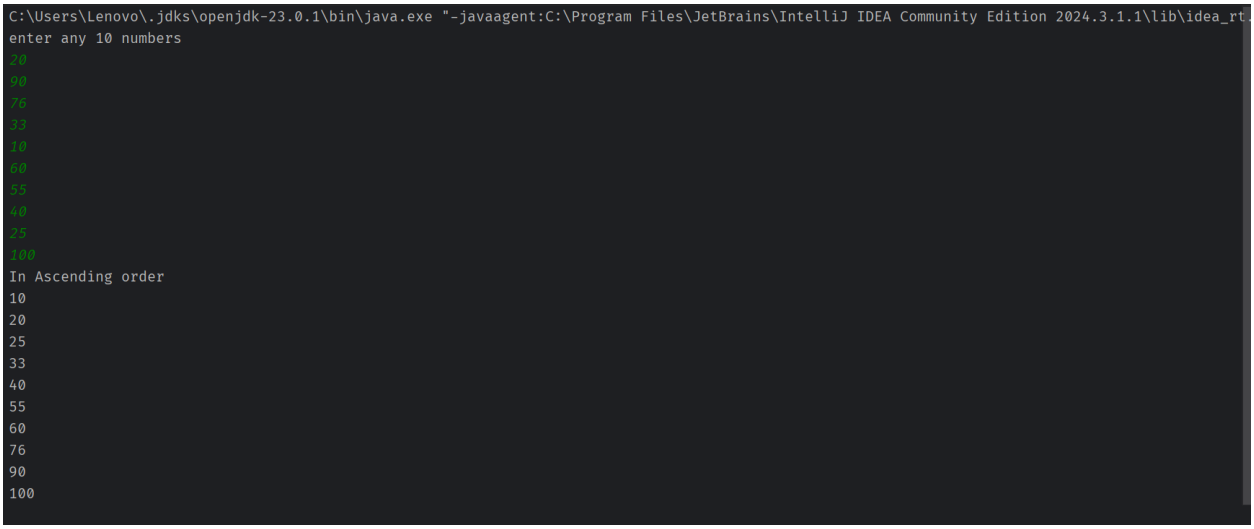
Lab 5

5.1. WAP to take 10 number input from user and sort in ascending order and display

```
import java.util.Scanner;
public class Ascending {
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        int[] numbers=new int[10];
        System.out.println("enter any 10 numbers");
        for(int i=0;i<10;i++){
            numbers[i]=in.nextInt();
        }
        int temp;
        for(int i = 0; i<10; i++){
            for(int j=i+1;j<10;j++){
                if(numbers[i]>numbers[j]){
                    temp=numbers[i];
                    numbers[i]=numbers[j];
                    numbers[j]=temp;
                }
            }
        }
        System.out.println("In Ascending order");
        for(int i=0;i<10;i++){
```

```
            System.out.println(numbers[i]);
        }
        in.close();
    }
}
```

Output:

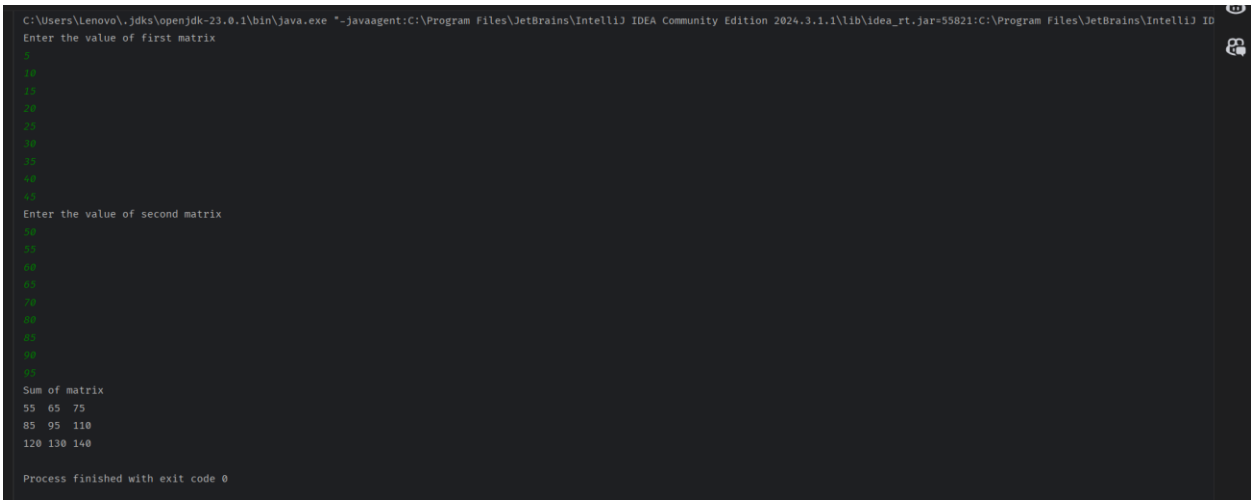


5.2. WAP to take two 3X3 matrix sum on third matrix

```
import java.util.Scanner;
public class MatrixSum {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        int[][] matrix1 = new int[3][3];
        int[][] matrix2 = new int[3][3];
        int[][] matrixSum = new int[3][3];
        System.out.println("Enter the value of first matrix");
```

```
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                matrix1[i][j] = in.nextInt();
            }
        }
        System.out.println("Enter the value of second matrix");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                matrix2[i][j] = in.nextInt();
            }
        }
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                matrixSum[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        System.out.println("Sum of matrix");
        for (int i = 0; i < 3; i++) {
            for (int j = 0; j < 3; j++) {
                System.out.print(matrixSum[i][j] + "\t");
            }
            System.out.println(" ");
        }
        in.close();
    }
}
```

Output:



5.3. WAP to take age of fifty person and display the age from older to younger and count the number of person whose age is greater than or equal to average age.

```
import java.util.Scanner;

public class CitizenOlderYounger {
    public static void main(String[] args) {
        int[] ages = new int[100];

        //create object of Scanner class to take input from user
        Scanner in = new Scanner(System.in);

        //taking 100 citizens age
```

```
for (int i = 0; i < 100; i++) {
    System.out.println("Enter age of " + (i + 1) + " citizen");
    ages[i] = in.nextInt();
}

//sorting and calculate average age;
int totalAge = 0;
for (int i = 0; i < 100; i++) {
    for (int j = i + 1; j < 100; j++) {
        if (ages[i] < ages[j]) {
            int temp = ages[i];
            ages[i] = ages[j];
            ages[j] = temp;
        }
    }
    totalAge += ages[i];
}

double averageAge = totalAge / 100;
int count = 0;
for (int i = 0; i < 100; i++) {
    System.out.println(ages[i]);
    if (ages[i] > averageAge) {
        count++;
    }
```

```
        }
       System.out.println("Average age: "+averageAge);
        System.out.println("total no of citizen whose age is greater than or equals to average age: "+count);
    }
}
```

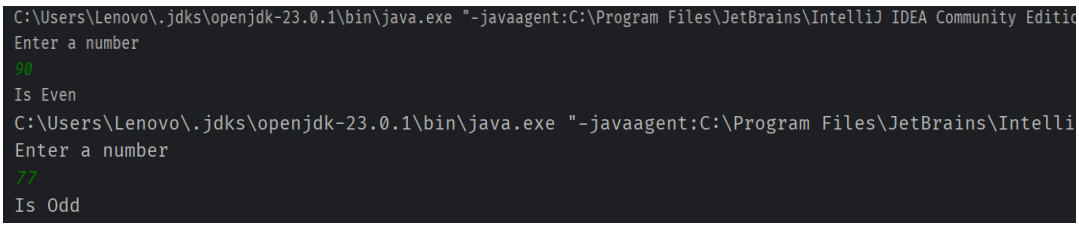
Date:2081/09/22

Lab 6

6.1.WAP to take a number input from user and determine either the input number is odd or even.

```
import java.util.Scanner;

public class OddEven {

    public static void main(String[] args) {

        Scanner in = new Scanner(System.in);

        System.out.println("Enter a number");

        int num = in.nextInt();

        if (num % 2 == 0) {

            System.out.println("Is Even");

        } else {

            System.out.println("Is Odd");

        }

        in.close();

    }

}
```

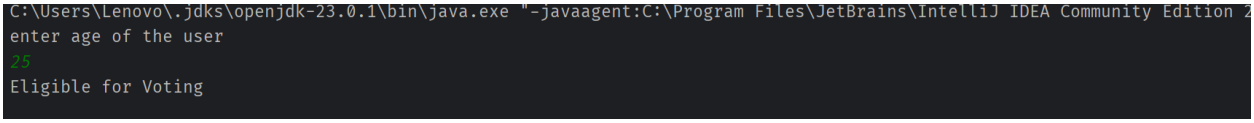
Output:



6.2. WAP to take age input from user and display eligible for voting if age is greater or equal to 18.

```
import java.util.Scanner;
public class VoterEligibility {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("enter age of the user");
        int age = in.nextInt();
        if (age >= 18) {
            System.out.println("Eligible for Voting");
        }
        if(age<18){
            System.out.println("Not Eligible for voting");
        }
        in.close();
    }
}
```

Output:



6.3. WAP using switch case statement when user input a name of day it will display the number of respective number

```
import java.util.Scanner;
public class DayToNumberOfWeek {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter the Day");
        String day = in.nextLine();
        switch (day.toLowerCase()) {
            case "sunday":
                System.out.println("1");
                break;
            case "monday":
                System.out.println(2);
                break;
            case "tuesday":
                System.out.println(3);
                break;
            case "wednesday":
                System.out.println(4);
                break;
            case "thursday":
                System.out.println(5);
                break;
            case "friday":
```

```
            System.out.println(6);
            break;
        case "saturday":
            System.out.println(7);
            break;
        default:
            System.out.println("Invalid ");
    }
    in.close();
  }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Enter the Day
Saturday
7
```

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editi
Enter the Day
Monday
2
```

6.4. WAP using switch case statement when user input a number from 1 to 7 it will display respective day name.

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

```
System.out.print("Enter the DayNumber[1 - 7]:");
int dayNumber = in.nextInt();

switch (dayNumber) {
    case 1:
        System.out.println("sunday");
        break;
    case 2:
        System.out.println("monday");
        break;
    case 3:
        System.out.println("tuesday");
        break;
    case 4:
        System.out.println("wednesday");
        break;
    case 5:
        System.out.println("thursday");
        break;
    case 6:
        System.out.println("friday");
        break;
    case 7:
        System.out.println("saturday");
        break;
    default:
```

```
            System.out.println("Invalid ");
        }


        in.close();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Enter the DayNumber[1 - 7]:
5
thursday
```

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2
Enter the DayNumber[1 - 7]:4
wednesday
```

6.5. WAP to take input mark of java program and display the respective grade based on PU grading system.

```
import java.util.Scanner;
public class MarksToGrade {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter java programming marks");
        double marks = in.nextDouble();
        if (marks >= 90) {
            System.out.println("A");
        } else if (marks >= 85 && marks < 90) {
```

```
            System.out.println("A-");
        } else if (marks >= 80 && marks < 85) {
            System.out.println("B+");
        } else if (marks >= 75 && marks < 80) {
            System.out.println("B");
        } else if (marks >= 70 && marks < 75) {
            System.out.println("B-");
        } else if (marks >= 65 && marks < 70) {
            System.out.println("C+");
        } else if (marks >= 60 && marks < 65) {
            System.out.println("C");
        } else if (marks >= 55 && marks < 60) {
            System.out.println("c-");
        } else if (marks >= 50 && marks < 55) {
            System.out.println("D+");
        } else if (marks >= 45 && marks < 50) {
            System.out.println("D");
        } else {
            System.out.println("F");
        }
        in.close();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Editio
Enter java programming marks

58
c-
```

Date:2081/09/23

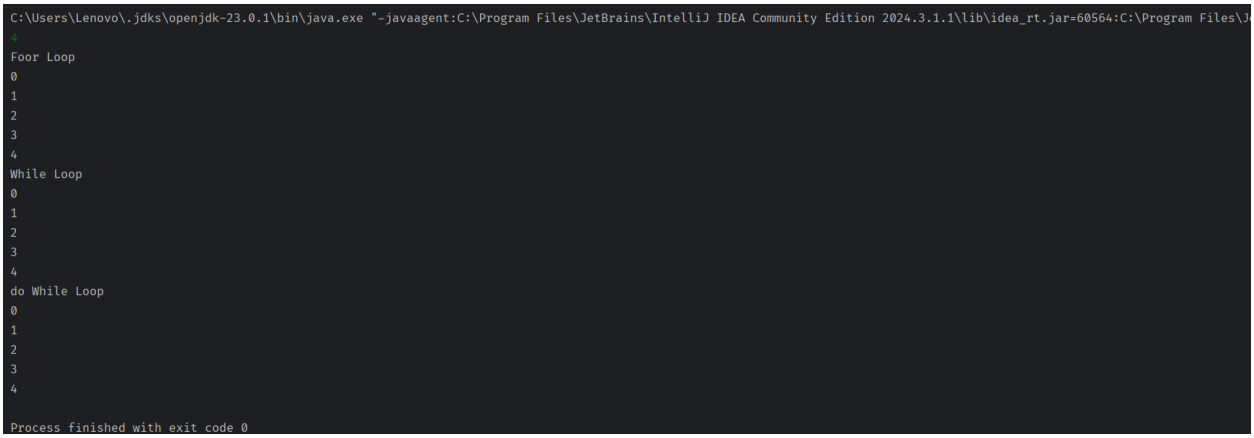
Lab 7

7.1.WAP to display number from 1 to input number using for, while and do while loop.

```
import java.util.Scanner;
public class Looping {
	public static void main(String[] args) {
		Scanner in=new Scanner(System.in);
		int n=in.nextInt();
		//for loop
		System.out.println("Foor Loop");
		for(int i=0;i<=n;i++) {
			System.out.println(i);
		}
		//while loop
		System.out.println("While Loop");
		int i=0;
		while(i<=n) {
			System.out.println(i);
			i++;
		}
		//do while loop
		System.out.println("do While Loop");
		i=0;
```

```
        do {
            System.out.println(i);
            i++;
        }while(i<=n);
        in.close();
    }
}
```

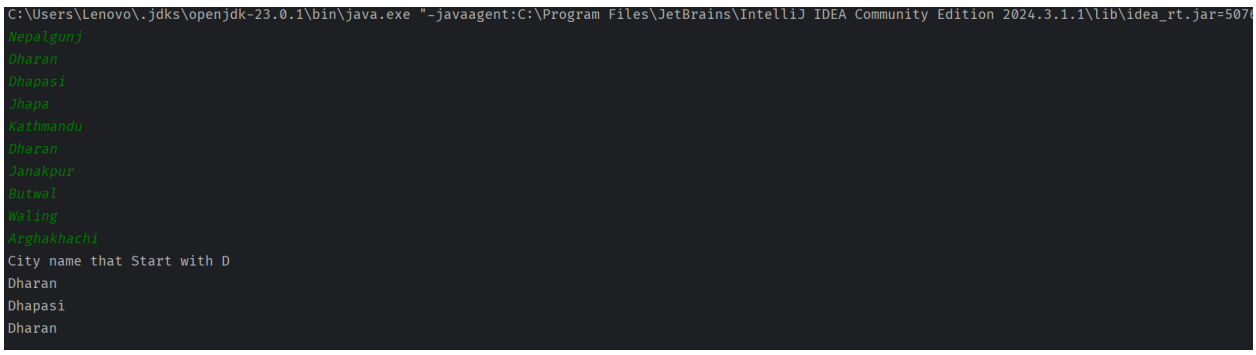
Output:



7.2. WAP to take 10 city name in any array from user and display using for each loop.

```
import java.util.Scanner;
public class CityNameDisplay {
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        String[] cityName=new String[10];
        for(int i=0;i<10;i++) {
            cityName[i]=in.nextLine();

        }
        System.out.println("City name that Start with D");
        for(String i:cityName) {
            if(i.toLowerCase().charAt(0)=='d') {
                System.out.println(i);
            }
        }
    }
}
```

Output:



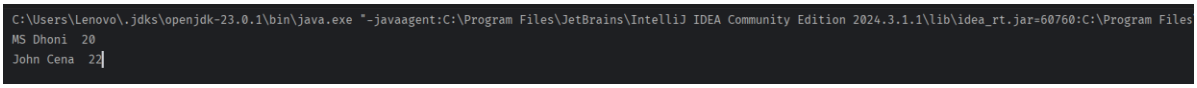
Date:2081/09/24

Lab 8

8.1. WAP to create two constructor to initialize default value of person name and age and parameterized value from another constructor

```
public class Person {
    private String name;
    private int age;
    // default constructor
    public Person() {
        name = "MS Dhoni";
        age = 20;
    }
    // parameterized constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public static void main(String[] args) {
        Person p = new Person();
        Person p1 = new Person("John Cena", 22);
        System.out.println(p.name + "  " + p.age);
        System.out.println(p1.name + "  " + p1.age);
    }
}
```

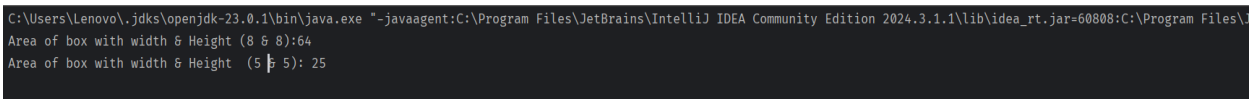
Output:



8.2. Create a class Box having width and height as a instance variable initialize default and parameterized value.

```
public class Box {
   private int width, height;
   public Box(){
      width = 5;
      height = 5;
   }
   public Box(int width, int height){
      this.width=width;
      this.height=height;
   }
   public static void main(String[] args) {
      Box b=new Box();
      Box b1=new Box(10,10);
      System.out.println("Area of box with width & Height ("+b.width+" &
"+b.height+"):"+(b.width*b.height));
      System.out.println("Area of box with width & Height  ("+b1.width+" & "+b1.height+"):
"+(b1.width*b1.height));
   }
   }
```

Output:



8.3. Create a class Box and one method to set value to width and height property and another method to calculate area.

```
public class Box {
        private int width;
        private int height;
        public void set(int width, int height) {
                this.width=width;
                this.height=height;
        }
        public void area() {
      System.out.println("Area of box with width & Height (" + width+ " & "+ height+
"):"+(width*height));
        }
   public static void main(String[] args) {
      Box b=new Box();
      b.set(10, 10);
      b.area();
   }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=60907:C:\Program Files\
Area of box with width & Height (7 & 7):49

Process finished with exit code 0
```

Date:2081/09/25

Lab 9

9.1. WAP to handle string related functionality using built-in function

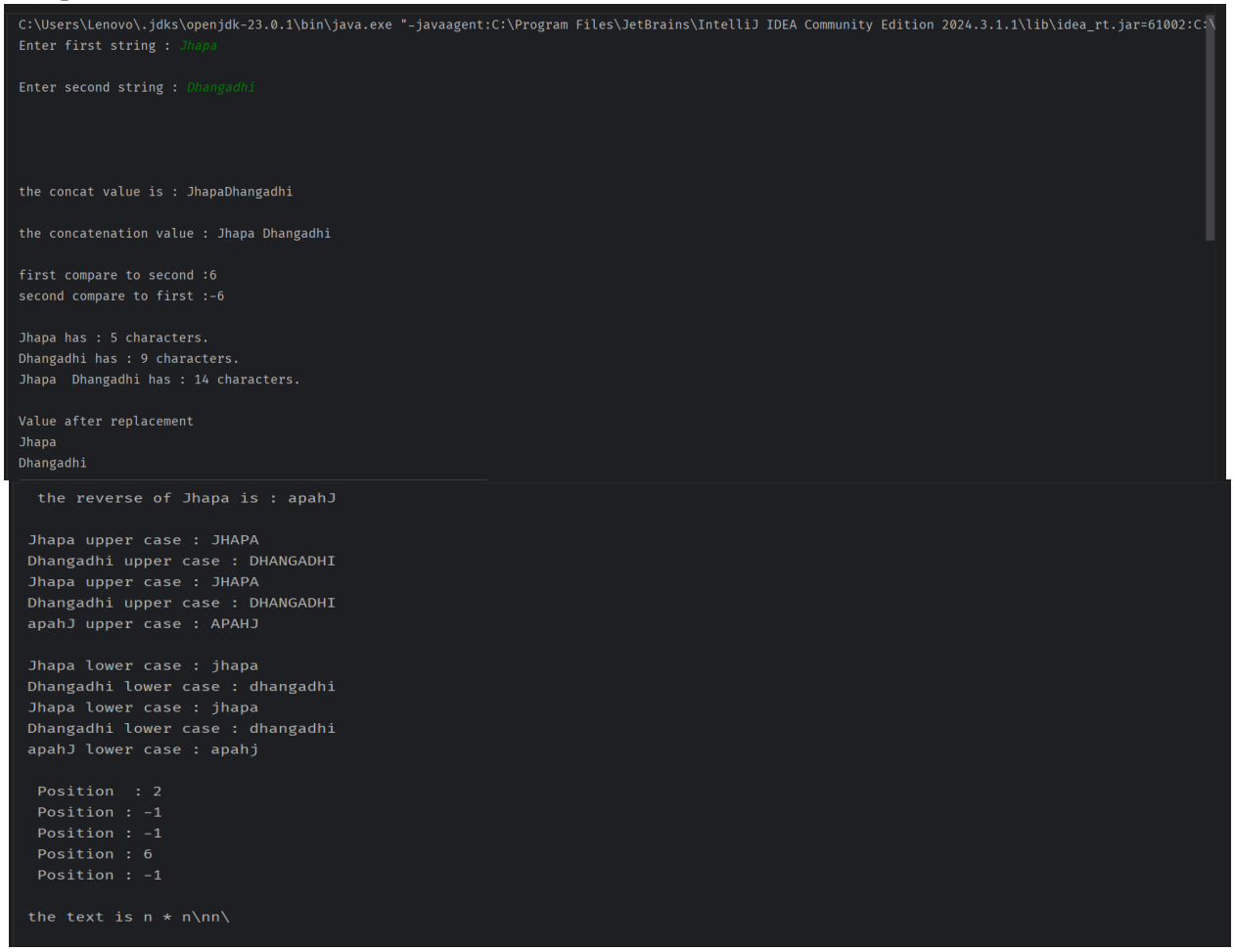
```
import java.util.Scanner;
public class StringDemo {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter first string : ");
        String first = in.nextLine();
        System.out.println();
        System.out.print("Enter second string : ");
        String second = in.nextLine();
        // equals
        System.out.println();
        if (first.equals(second)){
            System.out.println("both are same");
        }
        //equals ignore case
        System.out.println();
        if (first.equalsIgnoreCase(second)){
            System.out.println("both are equals ignoring case");
        }
        //==
        System.out.println();
```

```
if(first==second){
   System.out.println("both are ==");
}
//concat
System.out.println();
System.out.println("the concat value is : "+(first.concat(second)));
// for concatenation
System.out.println();
System.out.println("the concatenation value : "+first +" " +second);
// compare to
System.out.println();
System.out.println("first compare to second :"+ (first.compareTo(second)));
System.out.println("second compare to first :"+(second.compareTo(first)));
// to find length
System.out.println();
System.out.println(first+" has : "+first.length()+" characters. ");
System.out.println(second+" has : "+second.length()+" characters. ");
System.out.println(first+"  "+ second+" has : "+(first+second).length()+" characters. ");
// replacement
System.out.println();
System.out.println("Value after replacement");
String third = first.replace('n','d');
String fourth = second.replace('o','d');
System.out.println(third);
System.out.println(fourth);
// reverse
```

```
System.out.println();
StringBuffer r = new StringBuffer(first);
String fifth = r.reverse().toString();
System.out.println(" the reverse of "+first+" is : "+fifth);
// upper case
System.out.println();
System.out.println(first+" upper case : "+first.toUpperCase());
System.out.println(second+" upper case : "+second.toUpperCase());
System.out.println(third+" upper case : "+third.toUpperCase());
System.out.println(fourth+" upper case : "+fourth.toUpperCase());
System.out.println(fifth+" upper case : "+fifth.toUpperCase());
// lower case
System.out.println();
System.out.println(first+" lower case : "+first.toLowerCase());
System.out.println(second+" lower case : "+second.toLowerCase());
System.out.println(third+" lower case : "+third.toLowerCase());
System.out.println(fourth+" lower case : "+fourth.toLowerCase());
System.out.println(fifth+" lower case : "+fifth.toLowerCase());
// index or finding position
System.out.println();
System.out.println(" Position  : "+first.indexOf('a'));
System.out.println(" Position : "+second.indexOf('b'));
System.out.println(" Position : "+third.indexOf('c'));
System.out.println(" Position : "+fourth.indexOf('d'));
System.out.println(" Position : "+fifth.indexOf('e'));
```

```
        // escape sequencing
        System.out.println();
        String text = "the text is n * n\\nn\\ ";
        System.out.println(text);
    }
}
```

Output:



```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=61002:C:\
Enter first string : Jhapa

Enter second string : Dhangadhi



the concat value is : JhapaDhangadhi

the concatenation value : Jhapa Dhangadhi

first compare to second :6
second compare to first :-6

Jhapa has : 5 characters.
Dhangadhi has : 9 characters.
Jhapa  Dhangadhi has : 14 characters.

Value after replacement
Jhapa
Dhangadhi
  the reverse of Jhapa is : apahJ

 Jhapa upper case : JHAPA
 Dhangadhi upper case : DHANGADHI
 Jhapa upper case : JHAPA
 Dhangadhi upper case : DHANGADHI
 apahJ upper case : APAHJ

 Jhapa lower case : jhapa
 Dhangadhi lower case : dhangadhi
 Jhapa lower case : jhapa
 Dhangadhi lower case : dhangadhi
 apahJ lower case : apahj

  Position  : 2
  Position : -1
  Position : -1
  Position : 6
  Position : -1

 the text is n * n\nn\
```

Date:2081/09/26

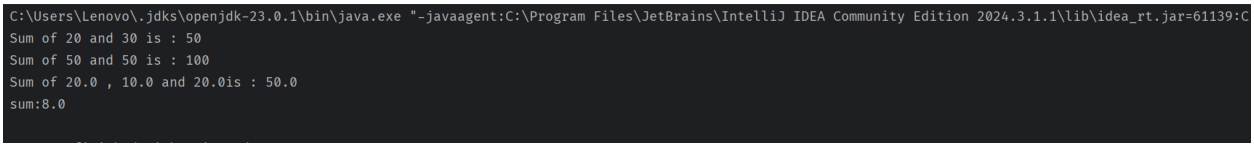
Lab 10

10.1.WAP to illustrate an example of method overloading

```
public class OverLoading {
    public void sum(){
        System.out.println("Sum of 20 and 30 is : "+(20+30));
    }
    public void sum(int a, int b){
        System.out.println("Sum of "+a+" and "+b+" is : "+(a+b));
    }
    public void sum(double a, double b, double c){
        System.out.println("Sum of "+a+" , "+b+" and "+c+ "is : "+(a+b+c));
    }
    public void sum(double[] a){
        double sum=0;
        for(double i:a ){
            sum +=i;
        }
        System.out.println("sum:"+sum);
    }
    public static void main(String[] args) {
        OverLoading obj= new OverLoading();
        double[] a= {1.2,2.3,4.5};
```

```
        obj.sum();
        obj.sum(50,50);
        obj.sum(20,10,20);
        obj.sum(a);
    }
}
```

Output:



10.2. WAP that will overload a method area to calculate area of triangle.

```
public class AreaOfTriangle {
    public void area(double base, double height) {
        System.out.println("Area of Triangle: " + 1.0 / 2 * base * height);
    }
    public double area(double height) {
        double base = 30;
        return (1.0 / 2 * base * height);
    }
    public double area(double base, double height, double form) {
        return (form * base * height);
    }
    public static void main(String[] args) {
        AreaOfTriangle t = new AreaOfTriangle();
```

```
        t.area(40, 10);
        double result = t.area(50);
        System.out.println("Width Fix area:" + result);
        double result2 = t.area(22, 25, 0.5);
        System.out.println("Another Area: " + result2);
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=61321:C:
Area of Triangle: 200.0
Width Fix area:750.0
Another Area: 275.0
```

Date:2081/09/28

Lab 11

11.1. WAP to show simple example of inheritance create a class person and a method work. Create subclass employee and a method getSalary and call necessary method

```
public class Person {
    public void work(){
        System.out.println("Person Working");
    }
}


public class Employee extends Person {
    public void getSalary() {
        System.out.println("Your salary is :80000");
    }
    public static void main(String[] args) {
        Employee emp = new Employee();
        emp.work();
        emp.getSalary();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=6255
Person Working
Your salary is :80000
```

11.2 Create a class student having two properties name and age. A subclass PayFee which have fee property both function have two functions to set and display data and create a class CollectFee from where student fee can be set and display

```
public class Student {
    private String name;
    private int age;
    public void setStudentInfo(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayStudentInfo() {
        System.out.println("Name: " + name + " Age: " + age);
    }
}
```

```
public class PayFee extends Student {
    private double fee;
    public void setFee(double fee){
        this.fee=fee;
    }
    public void displayFee(){
        System.out.println("Fee: "+fee);
    }
```

```
}
public class CollectFee {
    public static void main(String[] args) {
       //object create
        PayFee fee = new PayFee();
        //method call
        fee.setStudentInfo("Ankitt", 20);
        fee.setFee(45000);
        //method call
        fee.displayStudentInfo();
        fee.displayFee();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=62634:C
Name: Ankitt Age: 20
Fee: 45000.0
```

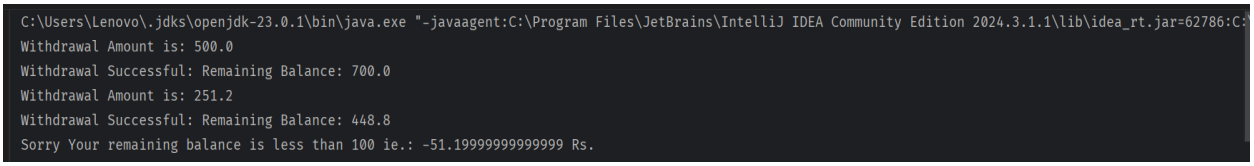
11.3 Write a Java program to create a class known as "BankAccount" with methods called deposit(double amt) and withdraw(double amt). Create a subclass called Savings Account that overrides the withdraw(double amt) method to prevent withdrawals if the account balance falls below one hundred.

```
public class BankAccount {
    double balance = 0;
    public void deposit(double amt) {
        balance = balance + amt;
    }
    public void withdraw(double amt) {
        balance = balance - amt;
    }
}
```

```
public class SavingAccount extends BankAccount{
    @Override
    public void withdraw(double amt) {
        double remaining=super.balance-amt;
        if(remaining<100){
            System.out.println("Sorry Your remaining balance is less than 100 ie.: "+remaining+" Rs.");
        }else{
            balance = balance - amt;
            System.out.println("Withdrawal Amount is: "+amt);
            System.out.println("Withdrawal Successful: Remaining Balance: "+balance);
```

```
        }
    }

    public static void main(String[] args) {
        SavingAccount sa = new SavingAccount();
        sa.deposit(300);
        sa.deposit(600);
        sa.deposit(300);
        sa.withdraw(500);
        sa.withdraw(251.20);
        sa.withdraw(500);
    }
}
```

Output:



Date:2081/10/03

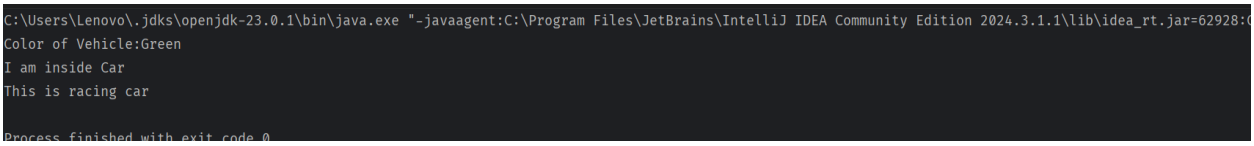
Lab 12

12.1.WAP to illlustrate Multilevel inheritance

```
public class Vehicle {
    private String color;
    void setColor(String color){
        this.color=color;
    }
    void showColor(){
        System.out.println("Color of Vehicle:"+color);
    }
}

public class Car extends Vehicle{
    public void carInfo(){
        System.out.println("I am inside Car");
    }
}

public class RacingCar extends Car{
    public void race(){
        System.out.println("This is racing car");
    }
```

```
    public static void main(String[] args) {
        RacingCar rc = new RacingCar();
        rc.setColor("Green");
        rc.showColor();
        rc.carInfo();
        rc.race();
    }
}
```

Output:



12.2. WAP to illlustrate Multilevel inheritance

```
public class GroundFloor {
    public void groundFloorInfo(){
        System.out.println("You are in Ground Floor!!!");
    }
}

public class FirstFloor extends  GroundFloor{
    public void firstFloorInfo(){
        System.out.println("You are in First Floor!!!");
    }
```

```
}
```

```
public class SecondFloor extends FirstFloor{
    public void secondFloorInfo(){
        System.out.println("You are in Second Floor!!!");
    }
    public static void main(String[] args) {
        SecondFloor secondFloor = new SecondFloor();
        secondFloor.groundFloorInfo();
        secondFloor.firstFloorInfo();
        secondFloor.secondFloorInfo();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1\lib\idea_rt.jar=62972
You are in Ground Floor!!!
You are in First Floor!!!
You are in Second Floor!!!
```

12.3. WAP to illustrate Hierarchy inheritance

```
public class Engine {
    public void on(){
        System.out.println("Engine ON");
    }
    public void off(){
        System.out.println("Engine OFF");
    }
    public void run(){
    }
}
```

```
public class DeliveryVan extends Engine{
    @Override
    public void run() {
            system.out.println("DeliveryVan has :100KM/HRS Speed");
    }
     public static void main(String[] args) {
                DeliveryVan cycle = new DeliveryVan();
              cycle.on();
              cycle.run();
              cycle.off();

    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\
Engine ON
DeliveryVan has :75/HRS Speed
Engine OFF
```

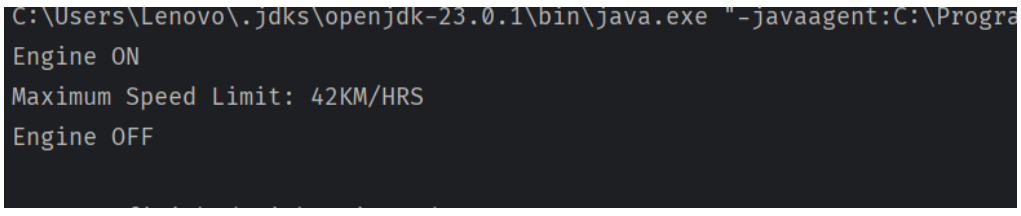
```
public class MotorCycle extends Engine{
    @Override
    public void run(){
        System.out.println("Maximum Speed Limit: 42KM/HRS");
    }
    public static void main(String[] args) {
        MotorCycle cycle = new MotorCycle();
        cycle.on();
        cycle.run();
        cycle.off();
    }
}
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition
Engine ON
Takeaway Successfully...
Aeroplane has :520KM/HRS Speed
Landed Successfully...
Engine OFF
```

```
public class Aeroplane extends Engine{
    @Override
    public void run() {
        System.out.println("Aeroplane has :600KM/HRS Speed");
    }
    public void takeAway(){
        System.out.println("Takeaway Successfully...");
    }
    public void land(){
        System.out.println("Landed Successfully...");
    }
    public static void main(String[] args) {
        Aeroplane aeroplane = new Aeroplane();
        aeroplane.on();
        aeroplane.takeAway();
        aeroplane.run();
        aeroplane.land();
        aeroplane.off();
    }
}
```

Output:



12.4. Write a Java program to create a vehicle class hierarchy. The base class should be Vehicle with subclasses Truck, Car and Motorcycle. Each subclass should have properties such as make, model, year, and fuel type. Implement methods for calculating fuel efficiency, distance traveled, and maximum speed.

```
public class Vehicle {
    private String make;
    private String model;
    private int year;
    private String fuelType;
     public void fuelEfficiency(){

    }
    public void distanceTraverse(){

    }
    public void maxSpeed(){

    }
}
```

```
public class Truck extends Vehicle{
    @Override
    public void fuelEfficiency() {
```

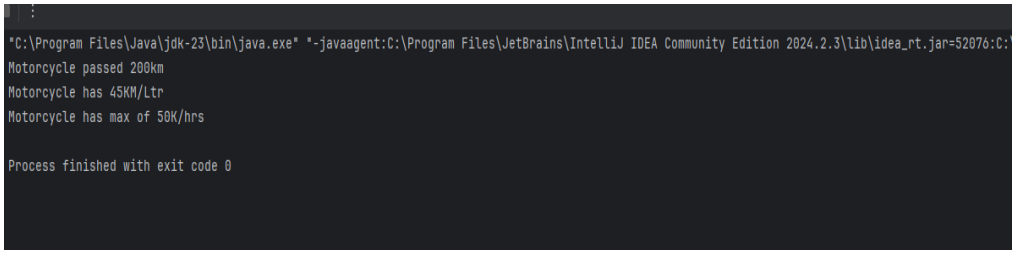
```
    System.out.println("Truck has 30KM/Ltr");
  }
  @Override
  public void distanceTraverse() {
    System.out.println("Truck passed 500km");
  }

@Override
  public void maxSpeed() {
    System.out.println("Truck has max of 70K/hrs");
  }
  public static void main(String[] args) {
    Truck truck = new Truck();
    truck.distanceTraverse();
    truck.fuelEfficiency();
    truck.maxSpeed();
   }
  }
```

Output:

```
C:\Users\Lenovo\.jdks\openjdk-23.0.1\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.3.1.1
Truck passed 550km
Truck has 45M/Ltr
Truck has max of 55K/hrs
```

```
public class Motorcycle  extends Vehicle{
    @Override
    public void fuelEfficiency() {
        System.out.println("Motorcycle has 45KM/Ltr");
    }
    @Override
    public void distanceTraverse() {
        System.out.println("Motorcycle passed 200km");
    }
    @Override
    public void maxSpeed() {
        System.out.println("Motorcycle has max of 50K/hrs");
    }
    public static void main(String[] args) {
        Motorcycle car = new Motorcycle();
        car.distanceTraverse();
        car.fuelEfficiency();
        car.maxSpeed();
    }
}
```

Output:



Date:2081/10/04

Date:2081/10/04

Date:2081/10/04

Lab 13

13.1. WAP to implement the concept of Polymorphism. Create a class Shape and a method draw and two properties name and color. Create a subclass Square and overwrite draw method. Create another subclass Triangle and overwrite draw method which will implement the concept of polymorphism. When user supply sides of Shape if he/she input three it should set and display name and color of triangle and draw a triangle if user supply four sides it should draw Square, otherwise it should call method of Shape

```
public class Shape {
    private String name;
    private String color;
    void draw(){
        System.out.println("Shape Draw");
    }
    void setInfo(String name, String color){
        this.name=name;
        this.color=color;
    }
    void displayInfo(){
        System.out.println("Name: "+name + " Color:"+color);
    }
}
```

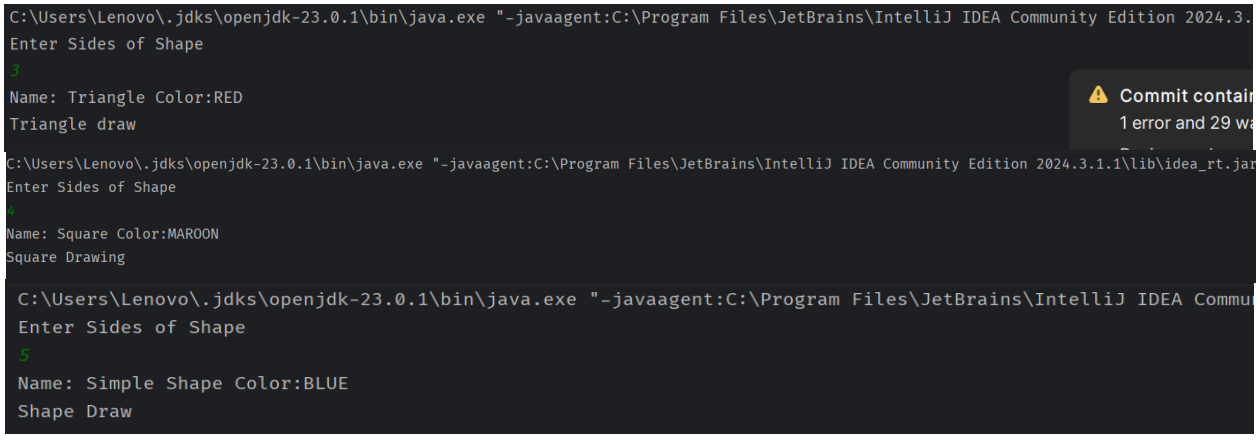
```
public class Square extends Shape{
    @Override
    void draw() {
        System.out.println("Square Drawing");
    }
}
```

```
public class Triangle extends Shape{
    @Override
    void draw() {
        System.out.println("Triangle draw");
    }
}
```

```
import java.util.Scanner;
public class ShapeDraw {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter Sides of Shape");
        int sides = in.nextInt();
        Shape shape;
        if(sides==3){
            shape=new Triangle();
```

```
            shape.setInfo("Triangle","RED");
            shape.displayInfo();
        }else if(sides==4){
            shape=new Square();
            shape.setInfo("Square","MAROON");
            shape.displayInfo();
        }else{
            shape= new Shape();
            shape.setInfo("Simple Shape","BLUE");
            shape.displayInfo();
        }
        shape.draw(); //it will show polymorphism behaviour based on users input
    }
}
```

Output:



13.2. Write down the step to create and use package with eg.

Step 1:Create a folder “My app” which is our application root folder

Step 2:Create a folder all in lower case.

Eg. nast that will be our first package.

Step 3:Go inside the first package i.e nast and create a class Student.java and write down package nast in the very first line of Student class and write down the code as below .

```
package nast;

public class Student{

        public void studentnfo(){

                System.out.println(“Inside Package nast”);

        }

}
```

Like wise you can add as required as class and interface as you want.

Step 4:Go to root folder and create a class Main.java and write down the code to import i.e use the class that we already created inside nast package . and write down the code as below.

```
//we are using Student class that is inside the nast package/folder but if we need all the class and interface inside it we can use * insted of Student(i.e import nast.*)

import nast.Student

public class Main{

        public static void main(String[] args){

                Student s=new Student();

                s.studentInfo();

        }

        }
```

Step 5(opt):Set class path of My app inside Environment Variable.

Step 6:Compile & Run the Main class

2. Compile the Main class:

Command: javac Main.java

3.Run the Main file

command: java HelloWorld