**15.08.2024**

**ASSIGNEMENT-1**

**2022503003**

**Exercise 1:** **Hello friends:**

Write the program and compile the code @command line to execute to greet your friends. Output: Hello, Alice, Bob, Charlie! Good Morning!

**CODE:**

import java.util.\*;

class hello{

    public static void main(String[] args){

        if(args.length==0){

            System.out.println("Hello!Good morning!");

        }

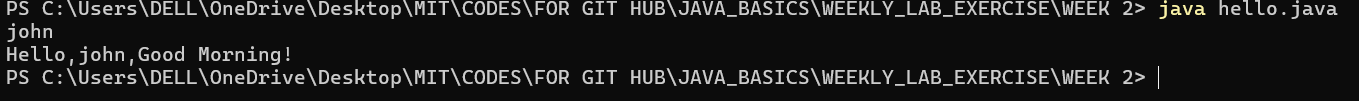
        else{

            System.out.printf("Hello,%s,Good Morning!",args[0]);

        }

    }

}



**Exercise 2: Error message**

Find the maximum Compile time and Run time error messages of simple one line output message. Example • Delete any of the semicolons. • Swap the word public, static, void, main • Omit the word public, static, void, main • Remove the array Subscript around string • Replace Sring with int or float • Replace String[] as String…

**CODE:**

import java.util.\*;

class error{

    public static void main(String[] args){

        System.out.println("Hello,world!");

    }

}



**CODE:**

import java.util.\*;

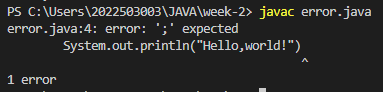
class error{

    public static void main(String[] args){

        System.out.println("Hello,world!")

    }

}



**CODE:**

import java.util.\*;

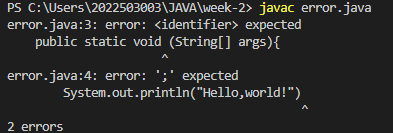
class error{

    public static void (String[] args){

        System.out.println("Hello,world!")

    }

}



**CODE:**

import java.util.\*;

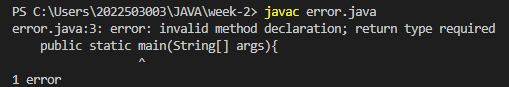
class error{

public static main(String[] args){

System.out.println("Hello,world!");

}

}



**CODE:**

import java.util.\*;

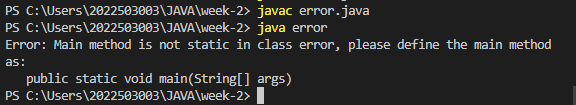
class error{

    public void main(String[] args){

        System.out.println("Hello,world!");

    }

}



**CODE:**

import java.util.\*;

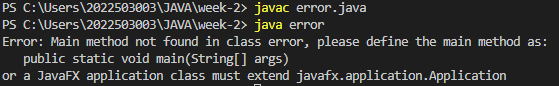
class error{

    static void main(String[] args){

        System.out.println("Hello,world!");

    }

}



**CODE:**

import java.util.\*;

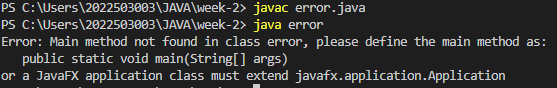
class error{

    public static void main(String args){

        System.out.println("Hello,world!");

    }

}



**CODE:**

import java.util.\*;

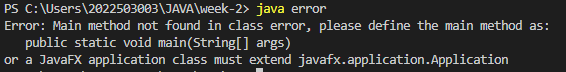
class error{

    public static void main(int[] args){

        System.out.println("Hello,world!");

    }

}



**CODE:**

import java.util.\*;

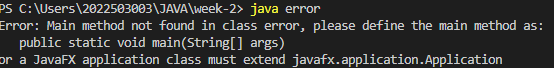
class error{

    public static void main(float[] args){

        System.out.println("Hello,world!");

    }

}



**Exercise 3: Conversation**

Write a Java program to create a Conversation between Java and Python Java: Hi, I'm Java. What's your name? Python: I'm Python. Nice to meet you! Java: Programmers use me for large-scale systems and performancecritical applications. Python: I'm best for rapid development and scripting tasks. Java: I use static typing for early error detection. Python: I use dynamic typing for more flexibility. Java: I run on the JVM, making me portable across many platforms. Python: I'm portable with the Python interpreter on any system. Java: Ideal for enterprise applications and Android apps. Python: Perfect for web development and data analysis. Java: My performance is strong with JVM optimizations. Python: I excel in ease of use and quick development cycles. Java: Use me for performance and large projects. Python: Use me for ease and speed in development. Java: This was great!. Bye for now! Python: Bye!

**CODE:**

import java.util.\*;

class hello{

    public static void main(String[] args){

        if(args.length==0){

            System.out.println("Hello!Good morning!");

        }

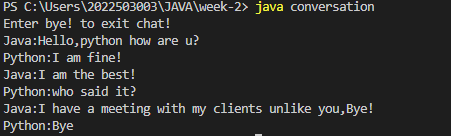
        else{

            System.out.printf("Hello,%s,Good Morning!",args[0]);

        }

    }

}



**Exercise4: Leap year**

Write a program to check if the given year is a leap year or not. Your input is an integer(year).The program should print a Boolean value: True if the year is a leap year, False if not. Constraint year≥1000 Input: 2024 Output: True Input: 2025 Output: False Also find the next leap year

import java.util.\*;

class leapyear{

    static Boolean checkleap(int a){

        if((a%4==0 && a%100!=0) || a%400==0){

                System.out.println(a+" is a leap year!!");

                return true;

        }

        else{

            System.out.println(a+" is a NOT leap year!!");

            return false;

        }

    }

    public static void main(String[] args){

        Scanner input=new Scanner(System.in);

        System.out.print("ENTER A YEAR:");

        int a=input.nextInt();

        while(a<=1000){

            System.out.print("CONSTRAINT VIOLATED!TRY AGAIN!!");

            a=input.nextInt();

        }

        checkleap(a);

        while(true){

            if(checkleap(++a)){

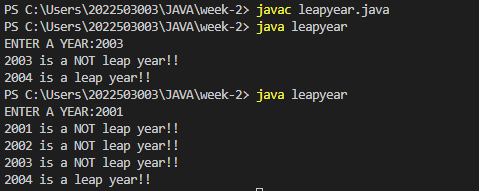
                break;

            }

        }

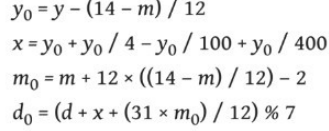
    }

}



**Exercise 5:Day of the Week**

Write a program that takes a date as input and prints the day of the week that date falls on. Read the three int input as m(month), d(day) and y(year). Use 1 of m for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday and so forth. Use the following formula for the Gregorian calendar.



**CODE:**

import java.util.\*;

class Dayoftheweek{

    public static void main(String[] args){

        Scanner input=new Scanner(System.in);

        System.out.print("MONTH:");

        int m=input.nextInt();

        System.out.print("DAY:");

        int d=input.nextInt();

        System.out.print("YEAR:");

        int y=input.nextInt();

        int y0=y-(14-m)/12;

        int x=y0+y0/4-y0/100+y0/400;

        int m0=m+12\*((14-m)/12)-2;

        int d0=(d+x+(31\*m0)/12)%7;

        System.out.println(d0);

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

        mapp.put(0,"Sunday");

        mapp.put(1,"Monday");

        mapp.put(2,"Tueday");

        mapp.put(3,"Thursday");

        mapp.put(4,"Fridday");

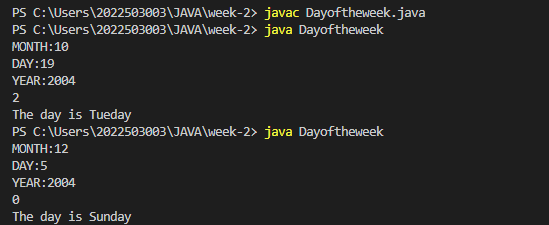
        mapp.put(5,"Saturday");

        String f=mapp.get(d0);

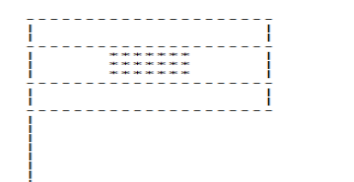
        System.out.println("The day is "+f);

    }

}



**Exercise 6:** Write a Java program to create a Indian Flag



class indianflag{

    static void o\_line(){

        for(int i=0;i<20;i++){

            System.out.print("- ");

        }

        System.out.println();

    }

    static void v\_line(){

        System.out.printf("|%38s|\n","");

    }

    static void m\_line(){

        System.out.printf("|%15s\*\*\*\*\*\*%17s|\n","","");

    }

    static void b\_line(){

        for(int i=0;i<15;i++){

            System.out.println("|");

        }

    }

    public static void main(String[] args){

        o\_line();

        v\_line();

        v\_line();

        o\_line();

        m\_line();

        m\_line();

        m\_line();

        o\_line();

        v\_line();

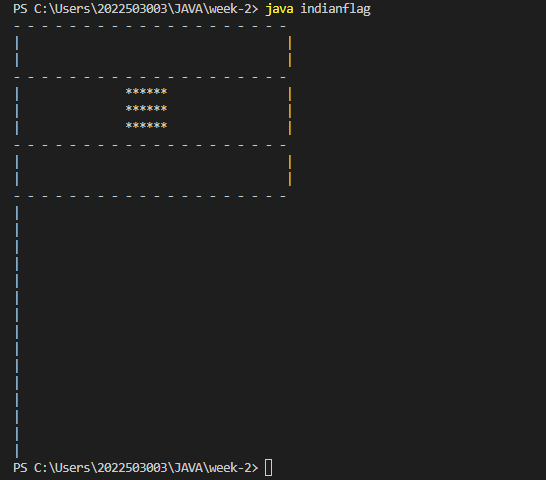
        v\_line();

        o\_line();

        b\_line();

    }

}



**Exercise 7: Model AND gate**

Write a program to model the AND gate using the linear combination of inputs formula Y=Bias +W0⋅X1+W1⋅X2 where X1 and X2 are the input values, and y is the output, determine the values for the weights W0 and W1, and the bias term that will correctly model the behavior of a logical AND gate. Use the condition that Y>0.5 results in output 1 and Y≤0.5 results in output 0.

import java.util.Scanner;

**CODE:**

public class modelAnd {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        double bias = -0.5 ;

        int w1 = 1 , w2 = 1 ;

        System.out.print("Enter x1 : ");

        int x1 = scanner.nextInt();

        System.out.print("Enter x2 : ");

        int x2 = scanner.nextInt();

        double y =bias + (w1\*x1) + (w2\*x2);

        if( y > 0.5 ){

            System.out.println(1);

        }

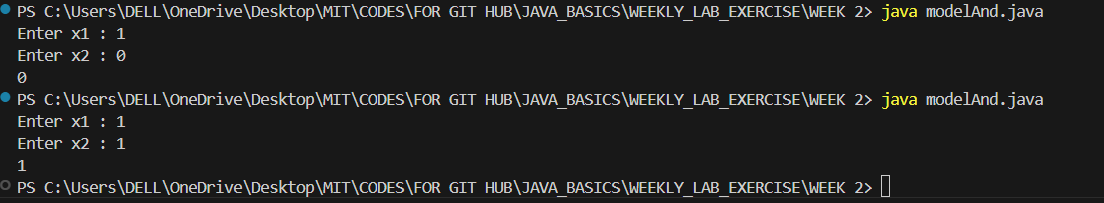
        else if( y <= 0.5){

            System.out.println(0);

        }

    }

}



**Exercise 8:** Write a program that converts a given integer into its equivalent words representation. The program should handle negative numbers and checks if the input is within the specified range of 0 to 999. Input 123 Output One Hundred and Twenty Three

**CODE:**

import java.util.\*;

class one\_s{

    String s1;

    HashMap<Integer,String> digit\_1=new HashMap<>();

    one\_s(){

        digit\_1.put(0,"");

        digit\_1.put(1,"One");

        digit\_1.put(2,"Two");

        digit\_1.put(3,"Three");

        digit\_1.put(4,"Four");

        digit\_1.put(5,"Five");

        digit\_1.put(6,"Six");

        digit\_1.put(7,"Seven");

        digit\_1.put(8,"Eight");

        digit\_1.put(9,"Nine");

    }

    void one(ArrayList<Integer> n){

        s1=digit\_1.get(n.get(n.size()-1));

        if (n.size()==1){

            System.out.println("The one digit number ----> "+s1);

        }

    }

}

class two\_s extends one\_s{

    String s2,x;

    HashMap<Integer,String> digit\_2=new HashMap<>();

    two\_s(){

        digit\_2.put(0,"");

        //for eleven to ninteen

        digit\_2.put(1,"");

        digit\_2.put(2,"Twenty");

        digit\_2.put(3,"Thirty");

        digit\_2.put(4,"Fourty");

        digit\_2.put(5,"Fifty");

        digit\_2.put(6,"Sixty");

        digit\_2.put(7,"Seventy");

        digit\_2.put(8,"Eighty");

        digit\_2.put(9,"Ninenty");

    }

    void two(ArrayList<Integer> n){

        if(n.get(n.size()-2)==1){

            int b=n.get(n.size()-1);

            HashMap<Integer,String> special\_x=new HashMap<>();

            special\_x.put(0,"Ten");

            special\_x.put(1,"Eleven");

            special\_x.put(2,"Twelve");

            special\_x.put(3,"Thirteen");

            special\_x.put(4,"Fourteen");

            special\_x.put(5,"Fifteen");

            special\_x.put(6,"Sixteen");

            special\_x.put(7,"Seventeen");

            special\_x.put(8,"Eighteen");

            special\_x.put(9,"Nineteen");

            x =special\_x.get(b);

            System.out.println("The Speical Two digit number -----> "+x);

        }

        else{

            one(n);

            s2=digit\_2.get(n.get(n.size()-2));

            if( n.size()==2){

                System.out.println("The Two digit number -----> "+s2+" "+s1);

            }

        }

    }

}

class three\_s extends two\_s{

    String s3;

    void three(ArrayList<Integer> n){

        two(n);

        s3=digit\_1.get(n.get(n.size()-3));

        if(n.get(n.size()-3)==1){

            if(n.get(n.size()-2)==1)

                System.out.println("The Three digit number ----> "+s3+" Hundred "+x);

            else

                System.out.println("The Three digit number ----> "+s3+" Hundred "+s2+" "+s1);

        }

        else if(n.size()==3){

            if(n.get(n.size()-2)==1)

                System.out.println("The Three digit number ----> "+s3+" Hundred and "+x);

            else

                System.out.println("The Three digit number ----> "+s3+" Hundred and "+s2+" "+s1);

        }

    }

}

class Helper{

    Helper(ArrayList<Integer> n){

        int s=n.size();

        System.out.println(s);

        three\_s number=new three\_s();

        if(s==0){

            System.out.println("Zero");

        }

        else if(s==1){

            number.one(n);

        }

        else if(s==2){

            number.two(n);

        }

        else if(s==3){

            number.three(n);

        }

    }

}

class numconvertor{

    public static void main(String[] args){

        Scanner input=new Scanner(System.in);

        System.out.print("Enter a 3 digit number from 0 to 999:");

        int num=input.nextInt();

        while(num>999 || num<0){

            System.out.print("Enter a VALID 3 digit number from 0 to 999:");

            num=input.nextInt();

        }

        ArrayList<Integer> n=new ArrayList<>();

        int a;

        while(num!=0){

            a=num%10;

            n.add(0,a);

            num=num/10;

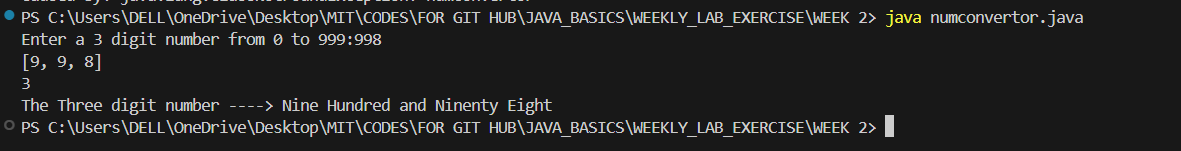
        }

        System.out.println(n);

        Helper z=new Helper(n);

    }

}



**Exercise 9: Casino Game** Write a Java program to simulate a simple Casino game where the player starts with 1000 credit points. Each roll costs 100 credits (the bet amount). For each roll, If the sum of the two dice is 7 or 11, the player wins and gains 100 credits or If the sum of the dice is 2, 3, or 12, the player loses 100 credits or For any other sum (4, 5, 6, 8, 9, or 10) there is no change in credits. The game continues until the player either goes bankrupt (reaches 0 credits) or reaches the target win amount of 2000 credits. Output: Current Credits: 1000 Rolling the dice... Dice sum: 7 You win 100 credits! New Credits: 1100 Current Credits: 1100 Rolling the dice... Dice sum: 4 Oops! No change in credits. New Credits: 1100 … Game Over! Final Credit:----[0 or 2000]

**CODE:**

import java.util.\*;

class casino{

    public static void main(String[] args){

        int credits=1000;

        Random r=new Random();

        System.out.println("Rolling the dice..........");

        int a=1+r.nextInt(6);

        int b=1+r.nextInt(6);

        System.out.println(a+" "+b+"CREDITS:"+credits);

        int sum=a+b;

        while(true){

            //System.out.println("Hello");

            if(credits==0||credits>=2000){

                if(credits==0){

                     System.out.println("You went bankrupt!");

                }

                else{

                     System.out.println("You won!");

                }

                break;

            }

            if(sum==7 || sum==11){

                credits+=100;

            }

            else if(sum==2||sum==3||sum==12){

                credits-=100;

            }

            a=1+r.nextInt(6);

            b=1+r.nextInt(6);

            sum=a+b;

            System.out.println(a+" "+b+"CREDITS:"+credits);

        }

    }

}

