OOPS Practical Question

1. Write a Java program to read 5 subject marks of a student and calculate the total and grade. The grade system is as follows.

Letter Grade	Grade Points	Marks Range
O (Outstanding)	10	91 – 100
A+ (Excellent)	9	81 – 90
A (Very Good)	8	71 – 80
B+ (Good)	7	61 – 70
B (Average)	6	50 – 60
RA	0	< 50

Link-->

https://github.com/Sachin-Ram/OOPS-JAVA/blob/main/exp2/marks.java

```
package exno2;

import java.util.Scanner;

public class Exno2 {

public static void main(String args[])

  {


    int marks[] = new int[5];

    int i;

    float total=0, avg;

    try(Scanner scanner = new Scanner(System.in))

    {
```

```
        for(i=0; i<5; i++) {
            System.out.print("Enter Marks of Subject"+(i+1)+":");
            marks[i] = scanner.nextInt();
            total = total + marks[i];
        }
    }
        avg = total/5;
        System.out.print("The student Grade is: ");
        if(avg>=90)
        {
            System.out.print("A");
        }
        else if(avg>=80 && avg<90)
        {
          System.out.print("B");
        }
        else if(avg>=70 && avg<80)
        {
            System.out.print("C");
        }
        else
        {
            System.out.print("D");
        }
```

}

}

2.

a. Define a class named COMPLEX for representing complex numbers that contains necessary data members and member functions. A complex number has the general form a + ib, where a is the real part and b is the imaginary part (i stands for imaginary). Include methods for all the four basic arithmetic operators.

b. Write a Java program that determines the number of days in a month.

Link-->illa

Algorithm:

Begin

Define a class operations with instance variables real and imag

Input the two complex numbers c1=(a+ib) and c2=(c+id)

Define the method add(c1,c2) as (a+ib)+(c+id) and stores result in c3

Define the method sub(c1,c2) as (a+ib)-(c+id) and stores result in c3

Define the method mul(c1,c2) as (a+ib)*(c+id) and store the result in c3 as (ac-bd)+i(bc+ad)

Define the method div(c1,c2) as (a+ib)/(c+id) and stores the quotient c3 as

{(ac+bd)/(c2+d2)}+i{(bc-ad)/(c2+d2)}

Define the method display() which outputs each result

End

Program:

```
import java.io.*;

import java.util.*;

public class Complex
```

```
{
    public static void main(String args[])
    {
        int ch=0;
        float num1,num2,answer;
        Complex_Op cal = new Complex_Op() ;
        Scanner input =  new Scanner(System.in);
        System.out.print("Enter the first Number\n");
        System.out.print("Real Part:");
        num1 = input.nextInt();
        System.out.print("Imaginary Part:");
        num2 = input.nextInt();
        Complex_Op Object1  = new Complex_Op(num1,num2);

        System.out.print("Enter the Second Number\n");
        System.out.print("Real Part:");
        num1 = input.nextInt();
        System.out.print("Imaginary Part:");
        num2 = input.nextInt();
        Complex_Op Object2  = new Complex_Op(num1,num2);
        do
        {
            System.out.println("1.Add");
            System.out.println("2.Subtract");
```

```
                System.out.println("3.Multiplication");
                System.out.println("4.Division");
                System.out.println("5.Exit");
                System.out.print("Choose ur choice:");
                ch = input.nextInt();
                switch(ch)
                {
                        case 1:
                                cal.AddNumbers(Object1,Object2);
                                break;
                        case 2:
                                cal.SubtractNumbers(Object1,Object2);
                                break;
                        case 3:
                                cal.MultiplyNumbers(Object1,Object2);
                                break;
                        case 4:
                                cal.DivideNumbers(Object1,Object2);
                                break;
                        case 5:
                                break;
                }
        }while(ch!=5);
}
```

```
}
class Complex_Op
{
	private float real,imag;
	Complex_Op()
	{
		real=0;
		imag=0;
	}
	Complex_Op(float Comp1,float Comp2)
	{
		real=Comp1;
		imag=Comp2;
	}
	void  AddNumbers(Complex_Op C1,Complex_Op C2)
	{
		float real,imag;
		this.real = (C1.real + C2.real);
		this.imag = (C1.imag + C2.imag);
		System.out.println("Answer is:("+this.real+") + ("+this.imag+")i" );
	}
	void SubtractNumbers(Complex_Op C1,Complex_Op C2)
	{
		float real,imag;
```

```
        this.real = (C1.real - C2.real);
        this.imag = (C1.imag - C2.imag);
        System.out.println("Answer is:("+this.real+") + ("+this.imag+")i" );
    }
    void MultiplyNumbers(Complex_Op C1,Complex_Op C2)
    {
        float real,imag;
        this.real = (C1.real*C2.real - C1.imag*C2.imag);
        this.imag = (C1.real*C2.imag + C2.real*C1.imag);
        System.out.println("Answer is:("+this.real+") + ("+this.imag+")i" );
    }
    void DivideNumbers(Complex_Op C1,Complex_Op C2)
    {
        float real,imag,deno;
        deno = (C2.real*C2.real + C2.imag*C2.imag);
        this.real = (C1.real*C2.real + C1.imag*C2.imag)/deno;
        this.imag = (C2.real*C1.imag - C1.real*C2.imag)/deno;
        System.out.println("Answer is:("+this.real+") + ("+this.imag+")i" );
    }
}
```

Output:

```
Enter the first Number
Real Part:1
Imaginary Part:2
```

Enter the Second Number

Real Part:3

Imaginary Part:4

1.Add

2.Subtract

3.Multiplication

4.Division

5.Exit

Choose ur choice:1

Answer is:$\left(4.0\right)$ + $\left(6.0\right)$i

1.Add

2.Subtract

3.Multiplication

4.Division

5.Exit

Choose ur choice:2

Answer is:$\left(-2.0\right)$ + $\left(-2.0\right)$i

1.Add

2.Subtract

3.Multiplication

4.Division

5.Exit

Choose ur choice:3

Answer is:$\left(-5.0\right)$ + $\left(10.0\right)$i1.Add

2.Subtract

3.Multiplication

4.Division

5.Exit

Choose ur choice:4

Answer is:(0.44) + (0.08)i

1.Add

2.Subtract

3.Multiplication

4.Division

5.Exit

Choose ur choice:5

3. Write a Java program to create a Package "YEAR_I" which has a class YearIMarks (members – sub1mark, sub2mark). Create another package "YEAR_II" which has a class YearIIMarks (members – sub3mark, sub4mark). Create n objects of Student class (having rollNumber, name, YearIMarks and YearIIMarks). Calculate the Grade ('Pass' > =50 else 'Fail') for each subject and display the result of the student in proper format.

Link-->

https://github.com/RKHariharan/Java-OOPS/tree/main/Exp5/Program1

Can't include package Answers. 

4. Create a package named 'com'. Define subpackages;

'transact': with class 'Transaction' with static methods credit() and debit()

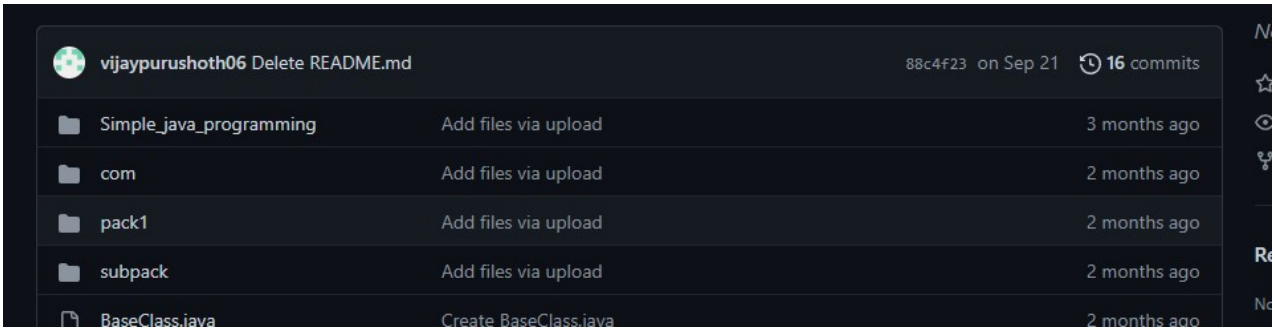
'loan': with class 'LoanAccount' with method doTransaction() which calls Transaction class methods.

Create one 'LoanAccount' object in main to perform operations on it by accepting command line arguments.

Link-->

https://github.com/vijaypurushoth06/Java-prog.

In this use com package to read this program. :)



5. **Define an interface "QueueOperations" which declares methods for a static queue. Define a class "MyQueue" which contains an array and front and rear as data members and implements the above interface. Initialize the queue using a constructor. Write the code to perform operations on a queue object.**

Link-->

https://github.com/CKBhalaji/EX-4/blob/main/Queue.java

```
import java.util.*;

interface queueoperations

{

   public void enqueue(int b);

   public void dequeue();

   public void display();

   public void peek();

}

class Myqueue implements queueoperations
```

```
{
    int i;
    int queue[];
    int arr_size;
    int rear;
    int front;
    int data;
    Myqueue(int size)
    {
        arr_size=size;
        queue=new int[arr_size];
        rear=-1;
        front=0;
    }
    public void enqueue(int b)
    {
        data=b;
        if(rear==arr_size-1)
        {
            System.out.println("overflow");
        }
        else
        {
        rear=rear+1;
```

```
    queue[rear]=data;
    }
}
public void dequeue()
{
    if(front==arr_size)
    {
        System.out.println("underflow");
    }
    else
    {
    System.out.println(queue[front]);
    front++;
    }
}
public void display()
{
    for(i=front;i<=rear;i++)
    {
        System.out.print("||"+queue[i]+"||-");
    }
}
public void peek()
{
```

```
        System.out.println("||"+queue[front]+"||");
    }
}
public class queue3 {
    public static void main(String args[])
    {
        int ch,n,a;
        Scanner s=new Scanner(System.in);
        System.out.println("Enter your array size");
        n=s.nextInt();
        Myqueue obj=new Myqueue(n);
        do
        {
            System.out.println("\n1.Insert\n2.Delete\n3.Display\n4.Display First value\nEnter your
choice");
            ch=s.nextInt();

            switch(ch)
            {
                case 1:
                {
                    System.out.println("Enter your data");
                    a=s.nextInt();
                    obj.enqueue(a);
```

```
            obj.display();
            break;
        }
        case 2:
        {
            obj.dequeue();
            obj.display();
            break;
        }
        case 3:
        {
            obj.display();
            break;
        }
        case 4:
        {
            obj.peek();
            break;
        }
        default:
            System.exit(0);
      }
    }while(ch!=5);
  }
```

```
}
```

6. Write a java class called 'student' with name, and rollno. Write a class 'Result' to get Marks of 3 subjects and another class "Sports' to get the points obtained in sports. Calculate the total Marks and displays the result (pass or fail) with points obtained in sports for three students using inheritance and constructor.

Link-->

https://github.com/RKHariharan/Java-OOPS/blob/main/Exp3/Exp3.java

```
package exp3;

import java.util.Scanner;

public class Exp3 {

    public static void main(String[] args) {

        student obj=new student("Hari","Abc","CSE",2022,51000.00);

        System.out.println(obj.getname());

        System.out.println(obj.getaddress());

        System.out.println(obj.getprogram());

        System.out.println(obj.getyear());

        System.out.println((obj.getfee()));

        Scanner obj1=new Scanner(System.in);

        System.out.println("Enter the address to be changed:");

        String ad1=obj1.next();

        obj.setaddress(ad1);

        System.out.println("Enter program name to be changed");

        String program1=obj1.next();

        obj.setprogram(program1);
```

```
        System.out.println("Enter a year to be changed:");
        int year1=obj1.nextInt();
        obj.setyear(year1);
        System.out.println("Enter fees to be changed:");
        double fees1=obj1.nextDouble();
        obj.getfee(fees1);
        System.out.println(obj.toString());
        staff obj2=new staff("Heisenberg","Alberqerque","Breaking bad",30000.00);
        System.out.println(obj2.getname());
        System.out.println(obj2.getaddress());
        System.out.println(obj2.getschool());
        System.out.println(obj2.pay());
        Scanner obj3=new Scanner(System.in);
        System.out.println("Enter a address to be changed:");
        String ad2=obj3.next();
        obj2.setaddress(ad2);
        System.out.println("Enter a school to be changed:");
        String school1=obj3.next();
        obj2.setschool(school1);
        System.out.println("Enter pay to be changed:");
        double pay1=obj3.nextDouble();
        obj2.setpay(pay1);
        System.out.println(obj2.toString());
    }
```

```
}
class person
{
   String name;
   String address;
   person(String n,String a)
   {
      name=n;
      address=a;
   }
   String getname()
   {
      return name;
   }
   String getaddress()
   {
      return address;
   }
   void setaddress(String a)
   {
      address=a;
   }
   @Override
```

```
    public String toString()
    {
       return "Name:"+name+"\n"+"Address:"+address;
    }
}
class student extends person
{
  String program;
  int year;
  double fee;
  student(String n,String a,String p,int y,double f)
  {
    super(n,a);
    program=p;
    year=y;
    fee=f;
  }
  String getprogram()
  {
     return program;
  }
  void setprogram(String p)
  {
     program=p;
```

```
    }
    int getyear()
    {
        return year;
    }
    void setyear(int y)
    {
        year=y;
    }
    double getfee()
    {
        return fee;
    }
    void getfee(double f)
    {
        fee=f;
    }
    @Override
    public String toString()
    {
        return
"Name:"+name+"\n"+"Address:"+address+"\n"+"Program:"+program+"\n"+"Year:"+year+"\n"+"Fe
e:"+fee;
    }
```

```
}
class staff extends person
{
   String school;
   double pay;
   staff(String n,String a,String s,double p)
   {
      super(n,a);
      school=s;
      pay=p;
   }
   String getschool()
   {
      return school;
   }
   void setschool(String s)
   {
      school=s;
   }
   double pay()
   {
     return pay;
   }
   void setpay(double p)
```

```
    {
        pay=p;
    }
    @Override
    public String toString()
    {
        return "Name:"+name+"\n"+"Address"+address+"\n"+"School:"+school+"\n"+"Pay:"+pay;
    }
}
```

7. Define an abstract class "car" with members reg_no, model, reg_date. Define two subclasses of this class – "transportVehicles " (validity_no, start_date, period) and "privateVehicle " (owner_name, owner_address). Define appropriate constructors. Create n objects which could be of either transportVehicles or privateVehicle class by asking the user's choice. Display details of all "privateVehicle" objects and all "transportVehicles" objects.

Link-->

https://github.com/Prem2372002/java/blob/main/expt4/Transport.java

```
package EXP4;
import java.util.Scanner;
public class transport
{
    public static void main(String[] args) {
    int n;
    System.out.println("ENTER NO OF OBJECTS-->\n");
    Scanner o=new Scanner(System.in);
```

```
n=o.nextInt();
car ob[]=new car[n];
for(int i=1;i<=n;i++)
{
  System.out.println(" 1.TRANSPORT VEHICLES ");
  System.out.println(" 2.PRIVATE VEHICLES ");
  System.out.println(" ENTER YOUR CHOICE ");
  int ch=o.nextInt();
if(ch==1)
{
System.out.println(" ENTER VALIDITY NUMBER--> ");
int v=o.nextInt();
System.out.println(" ENTER START DATE-->");
String s=o.next();
System.out.println(" ENTER PERIOD--> ");
String p=o.next();
ob[i]=new transportvehicles(v,s,p);
System.out.println(ob[i]);
}
if (ch==2)
{
   System.out.println("ENTER OWNER NAME-->");
   String oo=o.next();
   System.out.println("ENTER OWNER ADDRESS-->");
```

```
        String ad=o.next();
        ob[i]=new privatevehicles(oo,ad);
        System.out.println(ob[i]);
    }
    }
    }
}
abstract class car
{
String reg_no;
String model;
String reg_date;
}
class transportvehicles extends car
{
int validity_no;
String start_date;
String period;
transportvehicles(int n,String m,String d)
{
  validity_no=n;
  start_date=m;
  period=d;
  reg_no="9131";
```

```
 model="9ACD890";
 reg_date="12/09/2021";
}
public String toString()
{
   return "VALIDITY NUMBER-->\n"+validity_no+"MODEL-->\n"+start_date+"PERIOD--
>\n"+period+"REGISTRATION NUMBER-->\n"+reg_no+"MODEL--
>\n"+model+"REGISTRATION DATE-->\n"+reg_date;
}
}
class privatevehicles extends car
{
String owner_name;
String owner_address;
privatevehicles(String na,String ad)
{
 owner_name=na;
 owner_address=ad;
}
public String toString()
{
  return "OWNER NAME-->\n"+owner_name+"OWNER ADDRESS-->\n"+owner_address;
}
}
```

8. Create an interface "CreditCardInterface" with methods to viewCreditAmount, viewPin, changePin and payBalance. Create a class Customer (name, card number, pin, creditAmount – initialized to 0). Implement methods of the interface "CreditCardInterface" in Customer class. Create an array of customer objects and perform the following actions.

Pay Balance

Change Pin

Link-->

https://github.com/Prem2372002/java/blob/main/expt4/Card.java

```
package javaapplication4;

import java.util.*;

public class card {

   public static void main(String[] args) {

      Scanner ob=new Scanner(System.in);

      System.out.println("enter number of customer:");

      int n=ob.nextInt();

      customer obj[] = new customer[n];

      for(int i=0;i<n;i++)

      {

         System.out.println("enter customer name:");

         String m=ob.next();

         System.out.println("enter card no:");

         int ca=ob.nextInt();

         System.out.println("enter pin number:");

         int pi=ob.nextInt();
```

```
            System.out.println("enter amount:");
            int amt=ob.nextInt();
            obj[i]=new customer(m,ca,pi,amt);
            System.out.println("- - - - - - - -");
            obj[i].viewcreditamount();
            obj[i].viewpin();
            System.out.println("-> -> -> -> -> -> -> ->");
            obj[i].changepin();
            System.out.println("-> -> -> -> -> -> -> ->");
            System.out.println("ENTER AMOUNT:");
            int pa=ob.nextInt();
            obj[i].paybalance(pa);
            System.out.println(obj[i]);
            System.out.println();


        }
    }


}
interface credit_card_interface
{
    void viewcreditamount();
    void viewpin();
    void changepin();
```

```
    void paybalance(int pay);
}
class customer implements credit_card_interface
{
    String name;
    int cardno;
    int pin;
    int creditamount=0;
    customer(String m,int ca,int pi,int credit)
    {
        name=m;
        cardno=ca;
        pin=pi;
        creditamount=credit;
    }
    @Override
    public void viewcreditamount()
    {
        System.out.println("CREDIT AMOUNT = "+creditamount);
    }
    @Override
    public void viewpin()
    {
        System.out.println("PIN = "+pin);
```

```
}
@Override
public void changepin()
{
   System.out.println("CURRENT PIN NUMBER = "+pin);
   Scanner o=new Scanner(System.in);
   System.out.println("ENTER NEW PIN NUMBER-->");
   int n=o.nextInt();
   pin=n;
}
@Override
public void paybalance(int pay)
{
   if(creditamount!=0)
   {
      creditamount=creditamount-pay;
      System.out.println("BALANCE--> "+creditamount);
   }
   else
   {
      System.out.println("ALL CREDITS CLEARED");
   }
}
@Override
```

```
    public String toString()
    {
        return "name = "+name+" card no = "+cardno+" pin = "+pin+" balance = "+creditamount;
    }
}
```

9. **Write a Java program to perform the following task.**

Take an integer array of size 20, initialize values randomly between 10 and 90, simultaneously sum all values and calculate average. Now separate values below average and above average in ArrayLists. Finally print both lists in 2 separate rows.

Link-->illa.-Theriyala.

Task 1: Take an integer array of size 20. Initialize values randomly between 10 and 90. Simultaneously sum all values and calculate average. Now separate values below average and above average in ArrayLists. Finally print both lists in 2 separate rows?

```
int i,j, sum=0;
double avg;
int arr[]=new int[20];
for(i=0;i<arr.length;i++){
  arr[i]=(int)(Math.random()*80+10);
  sum=sum+arr[i];
}
avg=sum/(double)arr.length;
?ln("Average:"+avg);
ArrayList<Integer> bAvg=new
            ArrayList<Integer>();
ArrayList<Integer> aAvg=new
            ArrayList<Integer>();
for (i=0;i<arr.length;i++)
  if (arr[i]<avg)
    bAvg.add(arr[i]);
  else
    aAvg.add(arr[i]);
for (i=0;i<bAvg.size();i++)
  ?(bAvg.get(i)+" ");
?ln();
for (i=0;i<aAvg.size();i++)
  ?(aAvg.get(i)+" ");
?ln();
```

10. **Write a java program that reads a string from inputs containing first name, last name and computes an e-mail address with first 3 letters of the first name, first 4 letters of last name, '.' separator and domain. Display the outputs by invoking objects.**

Link-->

https://github.com/Sachin-Ram/OOPS-JAVA/blob/main/exp7/emailgenerator

```
package email;

import java.util.Scanner;

class convert{

    String f_name;

    String s_name;

    convert(String f_name,String s_name){

        this.f_name= f_name;

        this.s_name=s_name;



    }

    void display(){

        String s1;

        String s2;

        try{

        s1=f_name.substring(0,3);

        s2=s_name.substring(0,4);

        System.out.println("generated email "+s1+"."+s2+"@gmail.com");

        }

        catch(Exception e){

            System.out.println(e);

        }



    }

}
```

```
public class Email {
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.println("enter the first name");
        String f_name=sc.next();
        System.out.println("enter the last name");
        String s_name=sc.next();
        convert c=new convert(f_name,s_name);
        c.display();
        sc.close();
    }
}
```

11. Create a java abstract class to implement stack concept. Check for the overflow and empty conditions.

Link-->

https://www.sanfoundry.com/java-program-implement-stack

```
import java.util.*;
/* Class arrayStack */
class arrayStack
{
  protected int arr[];
  protected int top, size, len;
  /* Constructor for arrayStack */
  public arrayStack(int n)
```

```
    {
        size = n;
        len = 0;
        arr = new int[size];
        top = -1;
    }
    /* Function to check if stack is empty */
    public boolean isEmpty()
    {
        return top == -1;
    }
    /* Function to check if stack is full */
    public boolean isFull()
    {
        return top == size -1 ;
    }
    /* Function to get the size of the stack */
    public int getSize()
    {
        return len ;
    }
    /* Function to check the top element of the stack */
    public int peek()
    {
        if( isEmpty() )
            throw new NoSuchElementException("Underflow Exception");
        return arr[top];
    }
```

```
/* Function to add an element to the stack */
public void push(int i)
{
   if(top + 1 >= size)
      throw new IndexOutOfBoundsException("Overflow Exception");
   if(top + 1 < size )
      arr[++top] = i;
   len++ ;
}
/* Function to delete an element from the stack */
public int pop()
{
   if( isEmpty() )
      throw new NoSuchElementException("Underflow Exception");
   len-- ;
   return arr[top--];
}
/* Function to display the status of the stack */
public void display()
{
   System.out.print("\nStack = ");
   if (len == 0)
   {
      System.out.print("Empty\n");
      return ;
   }
   for (int i = top; i >= 0; i--)
      System.out.print(arr[i]+" ");
```

```
        System.out.println();


    }
}
/* Class StackImplement */
public class StackImplement
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.println("Stack Test\n");
        System.out.println("Enter Size of Integer Stack ");
        int n = scan.nextInt();
        /* Creating object of class arrayStack */
        arrayStack stk = new arrayStack(n);
        /* Perform Stack Operations */
        char ch;
        do{
            System.out.println("\nStack Operations");
            System.out.println("1. push");
            System.out.println("2. pop");
            System.out.println("3. peek");
            System.out.println("4. check empty");
            System.out.println("5. check full");
            System.out.println("6. size");
            int choice = scan.nextInt();
            switch (choice)
            {
```

```
        case 1 :
          System.out.println("Enter integer element to push");
          try
          {
            stk.push( scan.nextInt() );
          }
          catch (Exception e)
          {
            System.out.println("Error : " + e.getMessage());
          }
          break;
        case 2 :
          try
          {
            System.out.println("Popped Element = " + stk.pop());
          }
          catch (Exception e)
          {
            System.out.println("Error : " + e.getMessage());
          }
          break;
        case 3 :
          try
          {
            System.out.println("Peek Element = " + stk.peek());
          }
          catch (Exception e)
          {
```

```
            System.out.println("Error : " + e.getMessage());
        }
        break;
      case 4 :
        System.out.println("Empty status = " + stk.isEmpty());
        break;
      case 5 :
        System.out.println("Full status = " + stk.isFull());
        break;
      case 6 :
        System.out.println("Size = " + stk.getSize());
        break;
      default :
        System.out.println("Wrong Entry \n ");
        break;
      }
      /* display stack */
      stk.display();
      System.out.println("\nDo you want to continue (Type y or n) \n");
      ch = scan.next().charAt(0);
    } while (ch == 'Y'|| ch == 'y');
  }
}
```

12. Write a java program for exception handling:

a. To create a user defined exception whenever user input the word “hello”.

b. To add two integers and raise exception when any other character except number (0 – 9) is given as input.

Link-->

```
package javaapplication1; import java.util.Scanner;

class MyException extends Exception

{

   public MyException(String s2)

   {

      super(s2);

   }

}

public class Main{

   public static void main(String[] args)

   {

      int a,x1,x2,x3;

      String s,s1="hello",b,c;        boolean x=true;

      Scanner sc=new Scanner(System.in);        while(x)

      {

         System.out.println("Select the operation:\n1.Input a word\n2.Add two numbers\n3.Exit");

a=sc.nextInt();           switch(a)

         {              case 1:

               System.out.println("Enter the word:");                  s=sc.next();
```

```
try {
   if(s.equals(s1))
   {
      throw new MyException("EXCEPTIONCREATED");
   }                    else
   {
      System.out.println("Entered word:"+s);
   }
} catch (MyException e) {
   System.out.println(e.getMessage());
}                 break;              case 2:
System.out.println("Enter the integer1:");                b=sc.next();
System.out.println("Enter the integer 2:");                c=sc.next();
try
{
  x1=Integer.parseInt(b);
  x2=Integer.parseInt(c);
  x3=x1+x2;
  System.out.println("Sum:"+x3);
}
catch(NumberFormatException e)
{                    try
   {
      throw new MyException("WRONGCHARACTERS");
```

```
                }
                catch(MyException ex)
                {
                    System.out.println(ex.getMessage());
                    System.out.println("Exception caught!!");
                }               }               break;              case 3:
            x=false;              default:                 break;
        }
    }
  }
}
```

13. Create a class Doctor with attributes id, name, age and department. Initialize values through parameterized constructor. If age of Doctor is not in between 25 and 65 then generate user-defined exception "AgeNotWithinRangeException". If name contains numbers or special symbols raise exception "NameNotValidException". Define the two exception classes.

Link-->

```
package javaapplication1;

import java.util.Scanner;

public class Main{

   public static void main(String[] args)
{       Scanner o=new Scanner(System.in);
System.out.print("enter id:");
      int i=o.nextInt();
      System.out.print("enter department:");
```

```
        String d=o.next();
    boolean flag=true;
        System.out.print("enter age:");
        int a=o.nextInt();
        while(flag)


    {           t
    ry
            {
              if(a>25 &&a<65)
    flag=false;                 else
                    throw new AgeNotWithinRangeException("ineligible age");
            }
            catch(AgeNotWithinRangeException e)
            {
              System.out.println(e.getMessage());
            }
    if(flag)
            {
              System.out.print("enter age:");
              a=o.nextInt();
            }
        }
        System.out.print("enter name:");
        String n=o.next();
        while(!flag)
        {
          char c[]=n.toCharArray();
```

```
        try
{
        for(char s:c)
        {
          if(!Character.isLetter(s))
          {
            throw new NameNotValidException("invalid name");
          }
   }
flag=true;
        }
        catch(NameNotValidException e)
        {
          System.out.println(e.getMessage());
        }
if(!flag)
        {
          System.out.print("enter name:");
          n=o.next();
        }
      }
      Doctor obj=new Doctor(i,n,a,d);
      System.out.println(obj);
   }

} class
Doctor {
  int id;
```

```
    String name;
    int age;
    String department;
    Doctor(int i,String n,int a,String d)
    {       id=i;
name=n;
age=a;
department=d;
    }
    @Override
    public String toString()
    {
        return "id="+id+" name= "+name+" age= "+age+" department= "+department;
    }
}
class AgeNotWithinRangeException extends
Exception {
    AgeNotWithinRangeException(String msg)
    {
        super(msg);
    }
}
class NameNotValidException extends Exception
{
    NameNotValidException(String msg)
    {
        super(msg);
    }
```

```
}
```

14. A program accepts two integers as command line arguments. It displays all prime numbers between these two. Validate the input for the following criteria: Both should be positive integers. The second should be larger than the first. Create user defined exceptions for both.

Link-->illa

```
import java.io.*;
class validate extends Exception
{
    validate(String msg)
    {
        super(msg);
    }
}
public class PrimeNo {
    public static void main(String args[])throws IOException
    {
     int a,b;
     a=Integer.parseInt(args[0]);
     b=Integer.parseInt(args[1]);
     System.out.println("Prime No. Between"+a+"&"+b);
     try
     {
         if(a>=0&&b>=0)
         {
```

```
                if(b>a)
                {
                    for(int i=a;i<b;i++)
                    {
                        if(i==2||i==3||i==5||i==7)
                        {
                            System.out.println("\n"+i);
                        }
                        else
                        {
                            if(i%2!=0&&i%3!=0&&i%5!=0&&i%7!=0)
                            {
                                System.out.println("\n"+i);
                            }
                        }
                    }
                }
            else
            {
                throw new validate("Second Number Must Be Greaterv then First");
            }
        }
else
{
```

```
        throw new validate("Bothe Number Should Be Greater Then 0");
    }
}
catch(validate e)
{
    System.out.println(e);
}
}
}
```

15. Write a Java program 'WordCount' that counts the words in one or more files. Start a new thread for each file. For example, if you call

i. "java WordCount report.txt address.txt Homework.java "

ii. then the program might print

1. address.txt: 1052
2. Homework.java: 445
3. report.txt: 2099

Link-->

https://github.com/CKBhalaji/Ex-8/blob/main/Ex-8/WordCount.java

```
import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import java.util.logging.Level;
```

```
import java.util.logging.Logger;

class countword extends Thread

{

   //count  c;

   String str;

   int read,count = 0;

   countword(String c)

   {

      str=c;

   }

   @Override

   public void run()

   {

      FileReader file = null;

      try {

         file = new FileReader(str);

      }

      catch (FileNotFoundException ex) {

         Logger.getLogger(countword.class.getName()).log(Level.SEVERE, null, ex);

      }

      BufferedReader br = new BufferedReader(file);

      String line;

      try {

         while((read=file.read())!=-1)
```

```
            {
                if(read==' ')
                {
                    count++;
                }
            }
        } catch (IOException ex) {
            Logger.getLogger(countword.class.getName()).log(Level.SEVERE, null, ex);
        }
        count++;
        System.out.println(str+"-->:"+count);
    }
}
public class WordCount {
    public static void main(String args[]) {
        int n;
        n = args.length;
        countword t[]=new countword[n];
        for(int i=0;i<n;i++)
        {
            t[i]=new countword(args[i]);
            t[i].start();
        }
    }
```

```
    }
```

16. Write a Java program 'LineCounts.java' that will count the number of lines in each files that is specified on the command line. Note that multiple files can be specified, as in

a. "java LineCounts file1.txt file2.txt file3.txt".

Link-->

https://github.com/Prem2372002/java/blob/main/expt9/linecount.java

```
package exp_9;

import java.io.*;

public class linecount {

  public static void main(String[] args) {

    if (args.length == 0) {

      System.out.println("Usage:  java LineCounts <file-names>");

      return;

    }

     for (String arg : args) {

       System.out.print(arg + ":  ");

       countLines(arg);

     }

  }

  private static void countLines(String fileName) {

    BufferedReader in;

    int lineCount;

    try {

      in = new BufferedReader( new FileReader(fileName) );
```

```
    }
    catch (FileNotFoundException e) {
      System.out.println("Error.  Can't open file.");
      return;
    }
    lineCount = 0;
    try {
      String line = in.readLine();
      while (line != null) {
        lineCount++;
        line = in.readLine();
      }
    }
    catch (IOException e) {
      System.out.println("Error.   Problem with reading from file.");
      return;
    }

    System.out.println(lineCount + " lines");

  }
}
```

17. Develop a Java program to display the waiting list status given the PNR number of the train. Use JDBC connectivity to store the waiting list status.

a. Write a Java program to demonstrate that as a high-priority thread executes, it will delay the execution of all lower-priority threads.

b. Write a Java program to read from an input file and convert the words to lower case and write it in another file.

Link-->

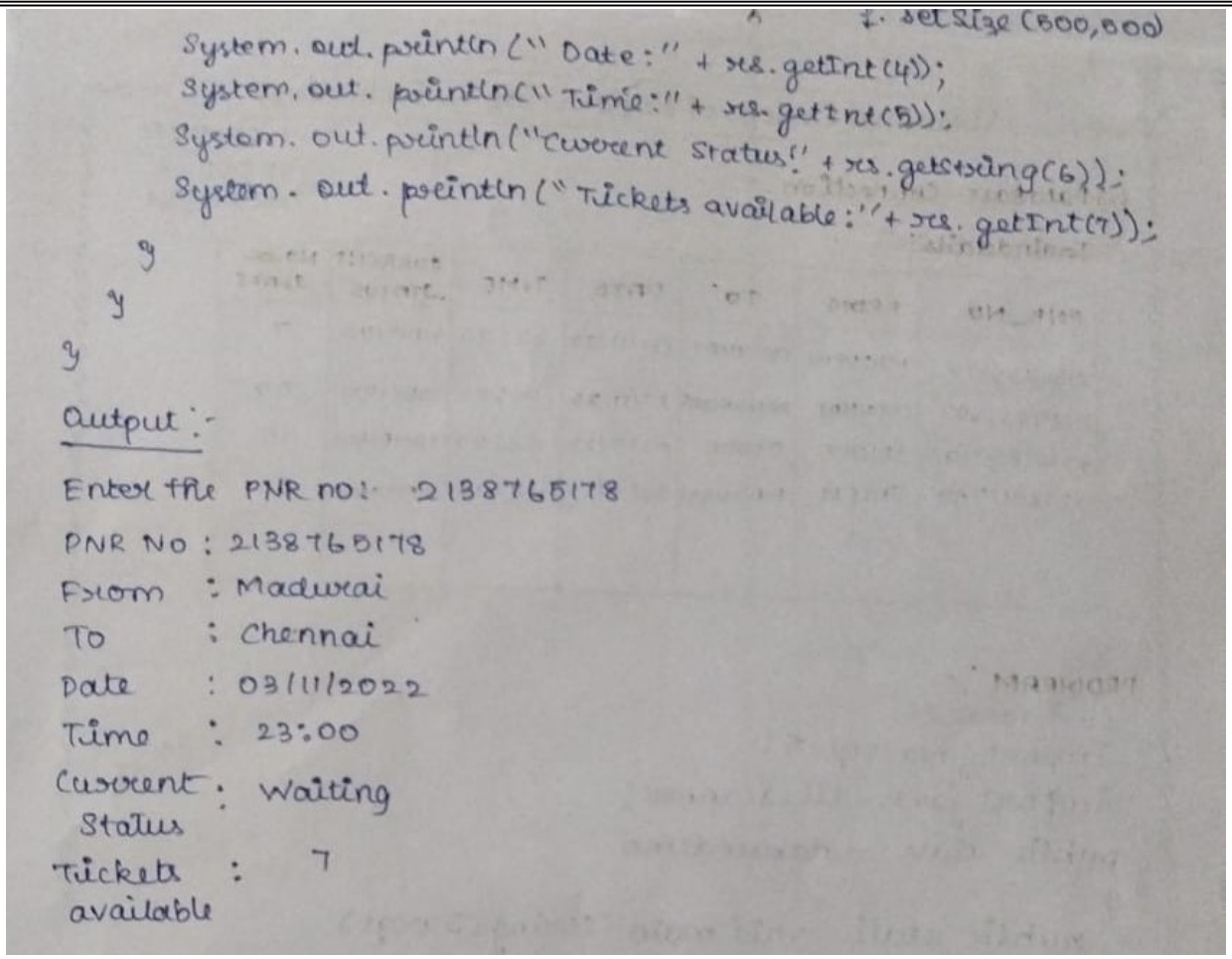
A. https://www.geeksforgeeks.org/java-thread-priority-multithreading/

B. https://github.com/CKBhalaji/Ex-9-OOPS/blob/main/EX-9/FileStoreCopy.java

PNR_NO	FROM	TO	DATE	TIME	CURRENT_STATUS	NO_OF_SEATS
2138765172	MADURAI	CHENNAI	03/11/22	23:00	WAITING	7
2189321602	CHENNAI	BANGALORE	27/11/22	01:00	CONFIRM	50
9267859108	TRICHY	ERODE	02/12/22	08:00	CONFIRM	15
2378911759	SALEM	MADURAI	30/11/22	14:00	WAITING	32

PROGRAM :-

```
package OOPS;
import java.sql.*;
import java.util.Scanner;
public class databasedemo
{
  public static void main (String[]args)
  {
    Connection con = DriverManager.getConnection
                ("jdbc:derby://localhost:1527/db;
                                    create=true");
    Statement st = con.createStatement();
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter the PNR no:");
    int pnr = sc.nextInt();
    ResultSet rs = st.executeQuery("Select * from traindetails
            where PNR_NO="+pnr+"");
    while(rs.next())
    {
      System.out.println("PNR NO:"+rs.getInt(1));
      System.out.println("From:"+rs.getString(2));
      System.out.println("To:"+rs.getString(3));
```



```
f.setSize(500,500)
System.out.println(" Date:" + rs.getInt(4));
System.out.println(" Time:" + rs.getInt(5));
System.out.println("Current Status:" + rs.getString(6));
System.out.println(" Tickets available:" + rs.getInt(7));
    }
  }
}
```

Output :-

```
Enter the PNR no: 2138765178
PNR No : 2138765178
From   : Madurai
To     : Chennai
Date   : 03/11/2022
Time   : 23:00
Current Status : Waiting
Tickets available : 7
```

A.// Java Program to Illustrate Priorities in Multithreading

// via help of getPriority() and setPriority() method

// Importing required classes

```
import java.lang.*;
```

// Main class

```
class ThreadDemo extends Thread {

    // Method 1

    // run() method for the thread that is called

    // as soon as start() is invoked for thread in main()

    public void run()

    {

        // Print statement

        System.out.println("Inside run method");

    }
```

```
// Main driver method
public static void main(String[] args)
{
        // Creating random threads
        // with the help of above class
        ThreadDemo t1 = new ThreadDemo();
        ThreadDemo t2 = new ThreadDemo();
        ThreadDemo t3 = new ThreadDemo();
        // Thread 1
        // Display the priority of above thread
        // using getPriority() method
        System.out.println("t1 thread priority : "
                                        + t1.getPriority());

        // Thread 1
        // Display the priority of above thread
        System.out.println("t2 thread priority : "
                                        + t2.getPriority());
        // Thread 3
        System.out.println("t3 thread priority : "
                                        + t3.getPriority());

        // Setting priorities of above threads by
        // passing integer arguments
        t1.setPriority(2);
```

```
t2.setPriority(5);

t3.setPriority(8);

// t3.setPriority(21); will throw

// IllegalArgumentException

// 2

System.out.println("t1 thread priority : "

                                + t1.getPriority());

// 5

System.out.println("t2 thread priority : "

                                + t2.getPriority());

// 8

System.out.println("t3 thread priority : "

                                + t3.getPriority());

// Main thread

// Displays the name of

// currently executing Thread

System.out.println(

        "Currently Executing Thread : "

        + Thread.currentThread().getName());

System.out.println(

        "Main thread priority : "

        + Thread.currentThread().getPriority());

// Main thread priority is set to 10

Thread.currentThread().setPriority(10);

System.out.println(

        "Main thread priority : "
```

```
                    + Thread.currentThread().getPriority());
    }
}
```

B.

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.util.Scanner;
class STUDENT
{
  int no;
  Scanner s=new Scanner(System.in);
  STUDENT()
  {
     System.out.println("Enter how many students");
     no=s.nextInt();
  }
  String Stu_Name[]=new String[10];
  String Stu_Address[]=new String[10];
  byte name[];
  byte add[];
  void InsertStuDetail()throws Exception
  {
        for(int i=0;i<no;i++)
        {
           System.out.println("Enter stu_name and stu_address");
           Stu_Name[i]=s.nextLine();
```

```
            Stu_Address[i]=s.nextLine();
        }
        FileOutputStream fout=new FileOutputStream("IN.txt");
        for(int i=0;i<no;i++)
        {
            name=Stu_Name[i].getBytes();
            add=Stu_Address[i].getBytes();
            fout.write(name);
            fout.write(add);
        }
        System.out.println("successfully written");
        fout.close();
    }
    void CopyLower()throws Exception
    {
       FileInputStream fin=new FileInputStream("IN.txt");
       FileOutputStream fout=new FileOutputStream("OUT.txt");
       int read;
       String c="";
       while((read=fin.read())!=-1)
       {
          c+=((char)read);
       }
       c=c.toLowerCase();
       byte b[]=c.getBytes();
       fout.write(b);
```

```
        fin.close();
        fin.close();
    }
}
public class FileStoreCopy
{
    public static void main(String[] args)throws Exception
    {
        STUDENT st=new STUDENT();
        st.InsertStuDetail();
        st.CopyLower();
    }
}
```

18. Write java programs that include generic method to satisfy the following property.

a. **To counts the number of odd integers in an integer list**

b. **To exchange the positions of two different elements in an array.**

c. **To find the maximal element in the range [begin, end] of a list.**

Link-->

https://github.com/Prem2372002/java/blob/main/ExpNo12/Program1.java

```
package ExpNo12;
public class Program1 {
    public static void main(String[] args) {
        Integer n[]={1,2,3,4,5,6,7,8,9};
        test1<Integer> obj=new test1<>();
        System.out.println("BEFORE SWAP");
```

```
        for (Integer num1 : n) {
            System.out.println(num1);
        }
        obj.swap(n,2,4);
        System.out.println("AFTER SWAP");
        for (Integer num1 : n) {
            System.out.println(num1);
        }

        test1<Integer> obj1=new test1<>();
        obj1.c(n);

        test1<Integer> obj2=new test1<>();
        obj2.max(n);

    }
}
class test1<T> {
    public <T> void swap(T[] a,int i,int j) {
            T temp = a[i];
            a[i] = a[j];
            a[j] = temp;

    }
```

```
    public  <T> void c(T[] value)
    {   int count = 0;
       for(int i=0;i<value.length;i++)
       {
          int v = (Integer)value[i];
          if(v%2==0){
             count++;
          }


       }


      System.out.println("count : "+count);
    }
    public <T extends Comparable<T>>void max(T[] a) {
    T max = a[0];
    for (T i : a) {
       if (i.compareTo(max) > 0) {
          max = i;
       }
    }
    System.out.println("THE MAXIMUM IN ARRAY IS : "+max);
}
}
```

19. Create a new Java GUI application to convert miles to kilometers when pressing the "Convert!" button. Note that you need to implement the ActionListener interface and override the actionPerformed() method. Note that 1 mile is equal to 1.609 kilometers.

Link-->

https://github.com/Prem2372002/java/blob/main/Expt10/converter.java

```
package EXP10;

import java.awt.Color;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JTextField;

public class conversion implements ActionListener

  {

   JLabel l1,l3;

   JTextField tf1,tf3;

   JButton b;

   conversion(){

      JFrame f= new JFrame();

      l1=new JLabel("MILE");

      l1.setBounds(50,50,150,20);

      l3=new JLabel("KILOMETER");

      l3.setBounds(50,150,150,20);
```

```
    tf1=new JTextField();
    tf1.setBounds(200,50,150,20);
    tf3=new JTextField();
    tf3.setBounds(200,150,150,20);
    tf3.setEditable(false);
    b=new JButton("CONVERT");
    b.setBounds(100,250,200,50);
    b.addActionListener(this);
    f.add(l1);f.add(l3);
    f.add(tf1);f.add(tf3);f.add(b);
    f.setSize(500,500);
    f.getContentPane().setBackground(Color.pink);
    f.setLayout(null);
    f.setVisible(true);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
}
@Override
public void actionPerformed(ActionEvent e) {
    String s1=tf1.getText();
    int a=Integer.parseInt(s1);
    double k=1.607;
    double c=0;
    if(e.getSource()==b)
    {
```

```
            c=a*k;
        }
        String result=String.valueOf(c);
        tf3.setText(result);
    }
public static void main(String[] args) {
        new conversion();
}}
```

20.Develop a course registration form with Name, Address, phone number,Gender(Male or Female),department(user have to select from CSE, ECE,EEE, Mech, Civil) and course (user have to select from (C,C++,JAVA,PYTHON). When the user submits the form, a dialog box should appear with a message "Username , you have successfully enrolled inCourse Name"

Link-->

https://github.com/RevathyJS/JavaPrograms/blob/master/swingdemo/registration.java

```
package swingdemo;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import javax.swing.JRadioButton;
```

```
import javax.swing.JTextArea;
import javax.swing.JTextField;
public class registration implements ActionListener{
    JLabel l1,l2,l3,l4,l5,l6;
    JTextField tf1,tf3;
    JTextArea tf2;
    JRadioButton r1,r2;
    JButton b1;
    ButtonGroup bg;
    JComboBox cb,cb1;
    registration(){
        JFrame f= new JFrame();
        l1=new JLabel("Name");
        l1.setBounds(50,50,150,20);
        l2=new JLabel("Address");
        l2.setBounds(50,100,150,20);
        l3=new JLabel("Phone No");
        l3.setBounds(50,200,150,20);
        tf1=new JTextField();
        tf1.setBounds(200,50,150,20);
        tf2=new JTextArea();
        tf2.setBounds(200,100,150,80);
        tf3=new JTextField();
        tf3.setBounds(200,200,150,20);
```

```
l4=new JLabel("Gender");
l4.setBounds(50,250,150,50);
r1=new JRadioButton("Male");
r2=new JRadioButton("Female");
bg=new ButtonGroup();
bg.add(r1);
bg.add(r2);
b1=new JButton("Submit");
r1.setBounds(200,250,100,50);
r2.setBounds(300,250,100,50);
String dept[]={"CSE","ECE","EEE","IT","MECH"};
cb=new JComboBox(dept);
String languages[]={"C","C++","C#","Java","PHP"};
cb1=new JComboBox(languages);
cb.setBounds(200,300,70,50);
cb1.setBounds(300,300,70,50);
b1.setBounds(200,400,100,30);
b1.addActionListener(this);
f.add(l1); f.add(l2); f.add(l3);f.add(l4);
f.add(tf1);f.add(tf2);f.add(tf3);f.add(r1);f.add(r2);  f.add(cb);f.add(cb1);f.add(b1);
f.setSize(500,500);
f.setLayout(null);
f.setVisible(true);
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        }
        public void actionPerformed(ActionEvent e) {


            if(e.getSource()==b1){
                JOptionPane.showMessageDialog(null,"User Name:"+tf1.getText()+"\nSelected
Course:"+cb1.getItemAt(cb1.getSelectedIndex()));
            }
        }
      public static void main(String[] args) {
        new registration();
      }
}
```

Prepared By.:

G.Eswar

N.Gowtham Kumar