LAB 4.

1. Create a package CIE which has two classes Student & Internals. The class Student has members like usn, name, sem. The class internals derived from student has an array that stores the internal marks scored in five courses of current semester of the student. Create another package SEE ~~marks scored in fi~~ which has the class External which is a derived class of student. This class has an array that stores the see marks scored in five courses of the current sem of the students. Import the two package in a file that declares the final marks of n students in all five course.

1. Create a folder CIE and save the programs Student.java & Internal.java

2 Create a folder SEE & save the program External.java within it

3. Save the Main program outside two folders

4. Compile Main.java & execute the Main class.

=>

```
package SEE;

impost CIE. Internals;

import java.util.Scanner;

public class Externals extends Internals{
        protected int marks[];
        protected int finalMarks[];

        public Externals(){
            marks = new int[5];
            finalMarks = new int[5];
        }

        public void inputSEEmarks(){
            Scanner s = new Scanner(System.in);
            SOP("Enter SEE marks for"+name);
            for(int i=0; i<5; i++){
                SOP("Subject"+(i+1));
                marks[i]= s.nextInt();
            }
        }

        public void calculatefinal(){
            for(int i=0; i<5; i++)
                finalMark[i]= marks[i]%2
                + super.marks[i];
        }
```

```
public void displayFinal(){
        displayStudentDetails();
        for(int i=0; i<5; i++)
            SOP("Subject"+(i+1)+":"+
            finalMarks[i]);
    }
}
```

```
package CSE

import java.util.Scanner;

public class Internals extends Student{
        protected int marks[] = new int[5

        public Internals(){


        }

        public void inputCIE(){
            Scanner s=new Scanner(System.

            System.out.print("Enter Inter

            for(int i=0; i<5; i++){
                SOP("Subject"+(i+1));
                marks[i]=scanner.next
                ();
            }
        }
}
```

```
package CIE;

import java.util.Scanner;

public class Student{
    protected String usn = new String();
    protected String name = new String();
    protected int sem;

    public void inputStuddetails(){
        Scanner s = new Scanner(System.in);
        SOP("Enter USN:");
        usn = s.next();
        SOP("Enter Name:");
        name = s.next();
        SOP("Enter Semester:");
        sem = s.nextInt();
    }

    public void displayStud(){
        SOP("USN" + usn);
        SOP("Name:" + name);
        SOP("sem:" + sem);
    }
}

import SEE.Externals;

public class Main{
    public static void main(String args[]){
        int numOfStudents = 2;
        Externals finalMarks[] = new
            Externals[numOfStudents];
```

```
import SEE.Externals;

public class Main

        for (int i=0; i<numOfStudents; i++){
            finalMarks[i]=new Externals();
            finalMarks[i].inputStudentDetails();
            System.out.println("Enter CIE marks");
            finalMarks[i].inputCIEmarks();
            System.out.println("Enter SEE marks");
            finalMarks[i].inputSEEmarks();
        }

        System.out.println("Displaying data:\n");

        for (int i=0; i<numOfStudents; i++){
            finalMarks[i].calculateFinalMarks();
            finalMarks[i].displayFinalMarks();
        }
    }
};
```

2. Write a program that demonstrates handling of exception in inheritance tree. Create a base class called 'father' and derived class called 'Son' which extend the base class. In father class, implement a constructor which takes the age and throws the exception WrongAge() when the input age<0. In Son class, implement a constructor that cases both father and son's age and throws an exception if son's age is >=father's age.

=>

```
import java.util.Scanner;

public class ExceptionInheritance{
    static class father{
        int age;

        father(int age) throws WrongAge{
            if(age<0){
                throw new WrongAge
                ("father's age cannot be
                 negative");
            }
            this.age = age;
        }
    }

    static class Son{
        int age;
        father father;

        Son(int fatherAge, int SonAge)
        throws WrongAge{
            father = new father(father
```

```
if (son.Age >= father.age){
        throw new WrongAge("Son's age
        cannot be equal or greater than
        father's age");

        this.age = son Age
    }
}

Static class WrongAge extends Exception{
        WrongAge(String message){
            Super(message);
    }
}

public static void main (string[] Args){
    Scanner s = new Scanner (System.in);
    System.out.println("Enter father's age");
    int fatherAge = s.nextInt();
    System.out.println("Enter son's age");
    int sonAge = s.nextInt();

    try{
        Son son=new Son (fatherAge, sonAge);
        System.out.println("Son's Age:"+son.
    } catch (WrongAge e){
        System.out.println.getmessage();
    }
}
}
```

Output:

Enter father's age:
29

Enter Son's age:
32

Son's age cannot be equal or greater than father's age.

3. Write a program which creates two threads, one thread displaying "BMS College of Engineering" once every ten seconds and another displaying "CSE" once every two seconds.

=>

```
class DisplayThread extends Thread{
        private String message;
        private int interval;

        public DisplayThread(String message,
          int interval){
                this.message = message;
                this.interval = interval;
        }

        public void run(){
                while(True){
                        System.out.println(message);

                        try{
                                Thread.sleep(interval);
                        }catch(InterruptedException
                        {       e.printStactTrace();
                        }
                }
        }
}
```

```
public class ThreadDemo {
    public static void main(String[] args) {
        DisplayThread t1 = new DisplayThread
        ("BMS College of Engg", 10000);
        t1.start();

        DisplayThread t2 = new DisplayThread
        ("CSE", 2000);
        t2.start();
    }
}
```

Output :

```
BMS College of Engineering
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
BMS College of Engineering
CSE
CSE
CSE
CSE
CSE
CSE
```