LAB PROGRAMS CODE AND OUTPUT:

6.Create a package CIE which has two classes- Student and Internals. The class Personal has members like usn, name, sem. The class Internals has an array that stores the internal marks scored in five courses of the current semester of the student. Create another package SEE 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 semester of the student. Import the two packages in a file that declares the final marks of n students in all five courses.

Student.java
```java
package CIE;
import java.util.Scanner;
public class Student
{
public String name;
public String usn;
public int sem;
public void display()
{
Scanner s=new Scanner(System.in);
System.out.print("Name:-");
name=s.next();
System.out.print("USN:-");
usn=s.next();
System.out.print("Semester:-");
sem=s.nextInt();
}
}
```

Internals.java
```java
package CIE;
import java.util.Scanner;
public class Internals extends Student
{
 public double ciem[];

 public void display()
 {
 ciem=new double[5];
 Scanner t = new Scanner(System.in);
 System.out.println("CIE Marks for 5 subjects(out of 50):");
 for(int i=0;i<5;i++)
 ciem[i]=t.nextDouble();
 }
}
```

Externals.java
```java
package SEE;
```

```java
import java.util.*;
import CIE.*;
public class Externals extends Student
{
 public double seem[];

 public void display()
 {
 seem=new double[5];
Scanner s=new Scanner(System.in);
 System.out.println("SEE Marks for 5 subjects(out of 100):");
 for(int i=0;i<5;i++)
 seem[i]=s.nextDouble();
 }
}
```

Main.java

```java
import CIE.*;
import SEE.*;
import java.util.Scanner;
public class Main
{
public static void main(String args[])
{
int n;
Scanner s=new Scanner(System.in);
System.out.print("Enter the number of students:-");
n=s.nextInt();
Student st[]=new Student[n];
Internals in[]=new Internals[n];
Externals e[]=new Externals[n];
for(int i=0;i<n;i++)
{
st[i]=new Student();
in[i]=new Internals();
e[i]=new Externals();
st[i].display();
in[i].display();
e[i].display();
System.out.println("Total marks of student "+st[i].name+" in 5 subjects are:");
for(int j=0;j<5;j++)
{
System.out.println(in[i].ciem[j]+(e[i].seem[j]/2));
}
}
}
}
```

OUTPUT:

```
PS C:\java\week6\lab6> javac Externals.java
PS C:\java\week6\lab6> javac Main.java
PS C:\java\week6\lab6> java Main
Enter the number of students:-1
Name:-Prathiksha
USN:-1bm19cs118
Semester:-3
CIE Marks for 5 subjects(out of 50):
45
38
32
47
30
SEE Marks for 5 subjects(out of 100):
70
65
90
85
40
Total marks of student Prathiksha in 5 subjects are:
80.0
70.5
77.0
89.5
50.0
PS C:\java\week6\lab6>                    []
```

7.Write a program to demonstrate generics with multiple object parameters.

```java
// A simple generic class with two type
// parameters: T and V.
class TwoGen<T, V> {
 T ob1;
 V ob2;
 // Pass the constructor a reference to
 // an object of type T and an object of type V.
 TwoGen(T o1, V o2) {
 ob1 = o1;
 ob2 = o2;
 }
 // Show types of T and V.
 void showTypes() {
 System.out.println("Type of T is " +
 ob1.getClass().getName());
 // Obtain and show values.
 System.out.println("value: " +ob1);

 System.out.println("Type of V is " +
 ob2.getClass().getName());
 System.out.println("value: " +ob2);
 }

}
// Demonstrate TwoGen.
class generics {
 public static void main(String args[]) {
 TwoGen<Integer, String> tgObj =
```

```
  new TwoGen<Integer, String>(88, "Generics");
  // Show the types.
  tgObj.showTypes();
   }
}
```

Output:

```
   Type of T is java.lang.Integer
   value: 88
   Type of V is java.lang.String
   value: Generics
   PS C:\java\week6\lab6> []
```

8.Write a program that demonstrates handling of exceptions in inheritance tree. Create a base class called "Father" and derived class called "Son" which extends 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.

```java
import java.util.*;
class WrongAge extends Exception{
    private String detail;
    WrongAge(String s)
    {
        detail = s;
    }
  public String toString()
    {
        return("Invalid age exception:" +detail);
    }
}
class father
{
int age;
father(int x) throws WrongAge
{
age=x;
if(age<0)

    throw new WrongAge("Age cant be negative");

}
}
class son extends father{
    int age1;
    son(int fage,int sage)throws WrongAge{
        super(fage);
        age1=sage;
```

```java
        if(age1>=age)
            throw new WrongAge("Son's age cant be greater than father's age");
    }

}

class lab8
{
    public static void main(String args[])
    {
        Scanner s =new Scanner(System.in);
        System.out.print("ENTER FATHER'S AGE: ");
        int m=s.nextInt();
        System.out.print("ENTER SON'S AGE: ");
        int n=s.nextInt();

        try{
         son ob = new son(m,n);

                System.out.println("Father's Age: "+ob.age);
                System.out.println("Son's Age: "+ob.age1);

        }
        catch(WrongAge e)
        {
            System.out.println(e);
        }
    }
}
```

OUTPUT:

```
PS C:\java\lab> javac lab8.java
PS C:\java\lab> java lab8
ENTER FATHER'S AGE: -10
ENTER SON'S AGE: 30
Invalid age exception:Age cant be negative
PS C:\java\lab> java lab8
ENTER FATHER'S AGE: 40
ENTER SON'S AGE: 41
Invalid age exception:Son's age cant be greater than father's age
PS C:\java\lab> java lab8
ENTER FATHER'S AGE: 40
ENTER SON'S AGE: 10
Father's Age: 40
Son's Age: 10
PS C:\java\lab> 
```

9.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.

```java
class NewThread implements Runnable
{
    private String name;
    private int interval;
    private Thread t;

    NewThread(String threadname, int interval)
    {
        this.name = threadname;
        this.interval = interval;
        t = new Thread(this, name);
        t.start();
    }

    public void run()
    {
        try {
            for(int i=5;i>0;i--) {
                System.out.println("Thread--" + this.name);
                Thread.sleep(this.interval);
            }
        }
        catch (InterruptedException e) {
            System.out.println(name + "Interrupted");
        }
    }
}

class lab9
{
    public static void main(String args[])
    {
        new NewThread("BMS College of Engineering", 10000);
        new NewThread("CSE", 2000);
    }
}
```

OUTPUT:

```
Thread--BMS College of Engineering
Thread--CSE
Thread--CSE
Thread--CSE
Thread--CSE
Thread--CSE
Thread--BMS College of Engineering
Thread--BMS College of Engineering
Thread--BMS College of Engineering
Thread--BMS College of Engineering
PS C:\java\lab>
```