lab-6 Packages:

```
import java.util.*;

import java.io.*;

import java.lang.*;

import cie.*;

import see.*;



public class student_end

{

        public static void main(String[] args)

        {

                int n;

                Scanner sc=new Scanner(System.in);

                int final_mark;

                System.out.println("Enter the Number of students ");

                n=sc.nextInt();

                internals[] in=new internals[n];

                external[] ex=new external[n];

                internals ob1=new internals();

                external ob2=new external();

                ob2.mar=new int[n];


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

                {

                        System.out.println("Enter the details of the student " + (i+1));
```

```
                in[i]=new internals();
                in[i].read();
                ex[i]=new external();
                ex[i].read();
            }
        System.out.println();
            for(int i=0;i<n;i++)
            {
                System.out.println("*Details Of The Student* " + (i+1));
                System.out.println("USN of the student is " + in[i].usn);
               System.out.println("Name of the stuednt is " + in[i].name);
               System.out.println("Semester of the student is " + in[i].sem);
                for(int j=0;j<5;j++)
                {
                    final_mark=in[i].a[j]+((ex[i].b[j])/2);
                    System.out.println("Final Mark of the student " + (i+1) + " " + " in course " +
(j+1) + " " + final_mark);

                }
                System.out.println();
            }

    }

}
```

lab-7

1)Exception:

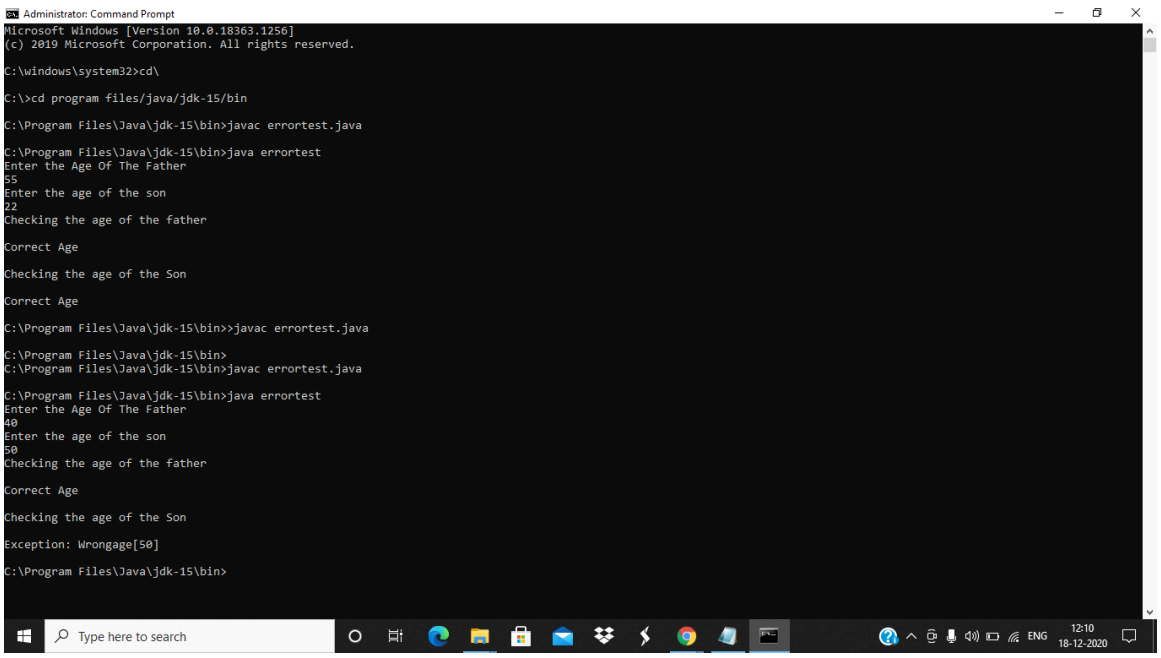
```
import java.util.*;

import java.io.*;

import java.lang.*;


class Wrongage extends Exception

{

	public int a;

	Wrongage(int x)

	{

		a=x;

	}


	public String toString()

	{

		return "Wrongage[" + a + "]";

	}

}

class father

{

	public int age;

	father(int a)

	{
```

```
		age=a;
	}
	public void check() throws Wrongage
	{
		System.out.println("Checking the age of the father ");
		System.out.println();
		if(age<0)
			throw new Wrongage(age);
		System.out.println("Correct Age");
	}
}

class son extends father
{
	public int son_age;
	son(int fa_age,int i)
	{
		super(fa_age);
		son_age=i;
	}
	public void check() throws Wrongage
	{
		super.check();
		System.out.println();
		System.out.println("Checking the age of the Son ");
		System.out.println();
```

```
		if(son_age<0||son_age>age)
			throw new Wrongage(son_age);
		System.out.println("Correct Age");
	}
}
public class errortest
{
	public static void main(String[] args)
	{
		int so_age,father_age;
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the Age Of The Father ");
		father_age=sc.nextInt();
		System.out.println("Enter the age of the son ");
		so_age=sc.nextInt();
		son s=new son(father_age,so_age);
		try
		{
			s.check();
		}catch(Wrongage w)
		{
			System.out.println("Exception: " + w);
		}

	}
```

}

Output:



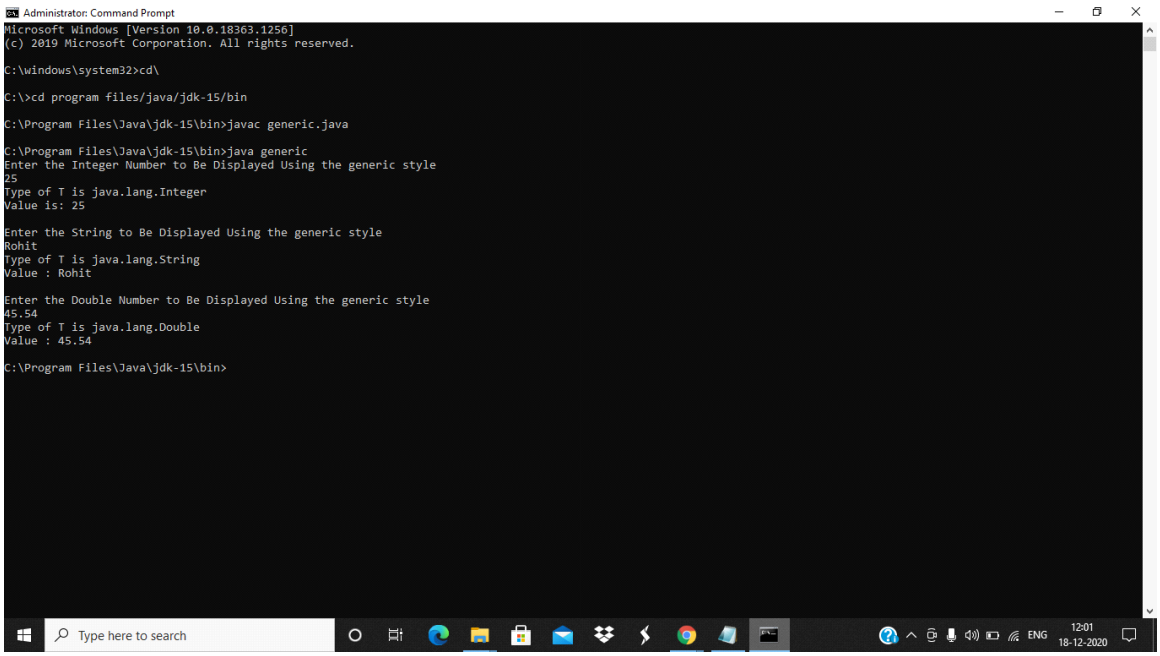
2)Generics:

```
import java.io.*;

import java.lang.*;

import java.util.*;


class gen<T>

{

        T ob;

        gen(T o)

        {
```

```
		ob=o;
	}
	T getob()
	{
		return ob;
	}
	void showtype()
	{
		System.out.println("Type of T is " + ob.getClass().getName());
	}
}

class generic
{
	public static void main(String[] args)
	{
		String n;
		Scanner sc=new Scanner(System.in);
		System.out.println("Enter the Integer Number to Be Displayed Using the generic style");
		n=sc.next();
		gen<Integer> ob1=new gen<Integer>(Integer.parseInt(n));
		ob1.showtype();
		int val=ob1.getob();
		System.out.println("Value is: " + val);

		System.out.println();
```

```
                System.out.println("Enter the String to Be Displayed Using the generic style");
                n=sc.next();
                gen<String> ob2=new gen<String>(n);
                ob2.showtype();
                String x=ob2.getob();
                System.out.println("Value : " + x);

                System.out.println();



                System.out.println("Enter the Double Number to Be Displayed Using the generic style");
                n=sc.next();
                gen<Double> ob3=new gen<Double>(Double.parseDouble(n));
                ob3.showtype();
                double ans=ob3.getob();
                System.out.println("Value : " + ans);

        }
}
```

Output:



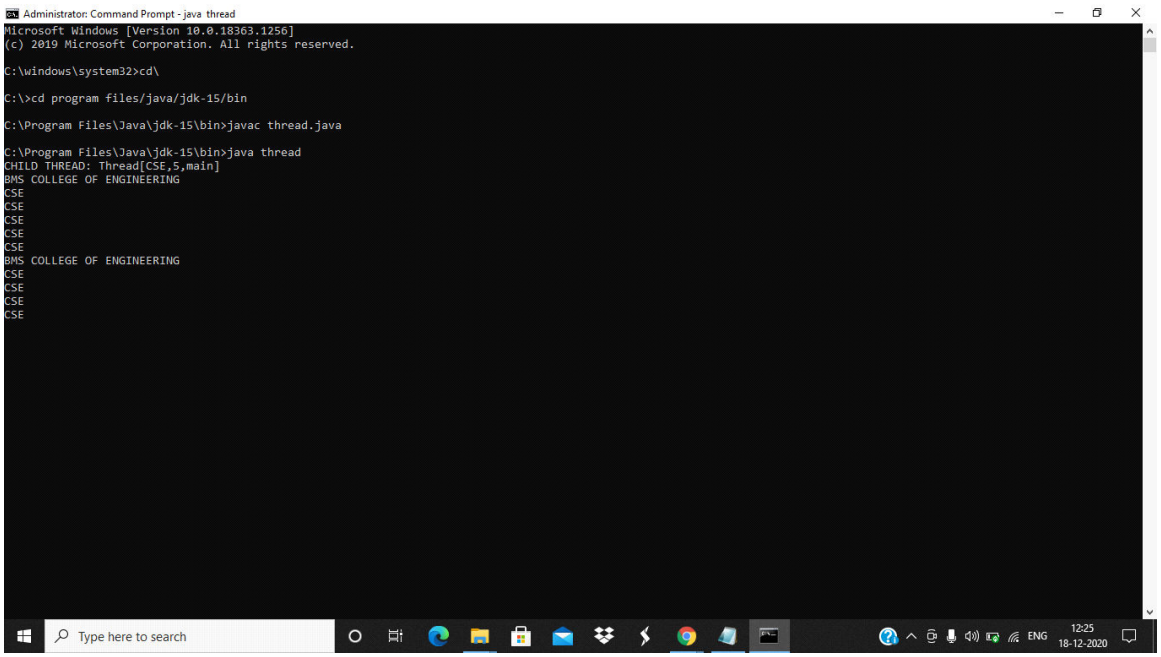
Lab-8 Threads:

```
import java.util.*;
import java.io.*;
import java.lang.*;
class newthread implements Runnable
{
        Thread t;
        newthread()
        {
                t=new Thread(this,"CSE");
                System.out.println("CHILD THREAD: "    + t);
        }
        public void run()
```

```
	{
		try{
		for(;;)
		{
			System.out.println("CSE");
			Thread.sleep(2000);
		}
		}catch(InterruptedException e){
			System.out.println("CSE Thread interrruoted ");
		}
		System.out.println("Exiting The CSE Thread");
	}
}

class thread
{
	public static void main(String[] args)
	{
		newthread nt=new newthread();
		nt.t.start();
		try{
			for(;;)
			{
			    System.out.println("BMS COLLEGE OF ENGINEERING");
			    Thread.sleep(10000);
			}
		}catch(InterruptedException e){
			System.out.println("Main Thread Interrupted: ");
```

```
            }
            System.out.println("Exiting out of the main thread ");
        }
}
```

Output:



Lab-9

1)Car mechanic:

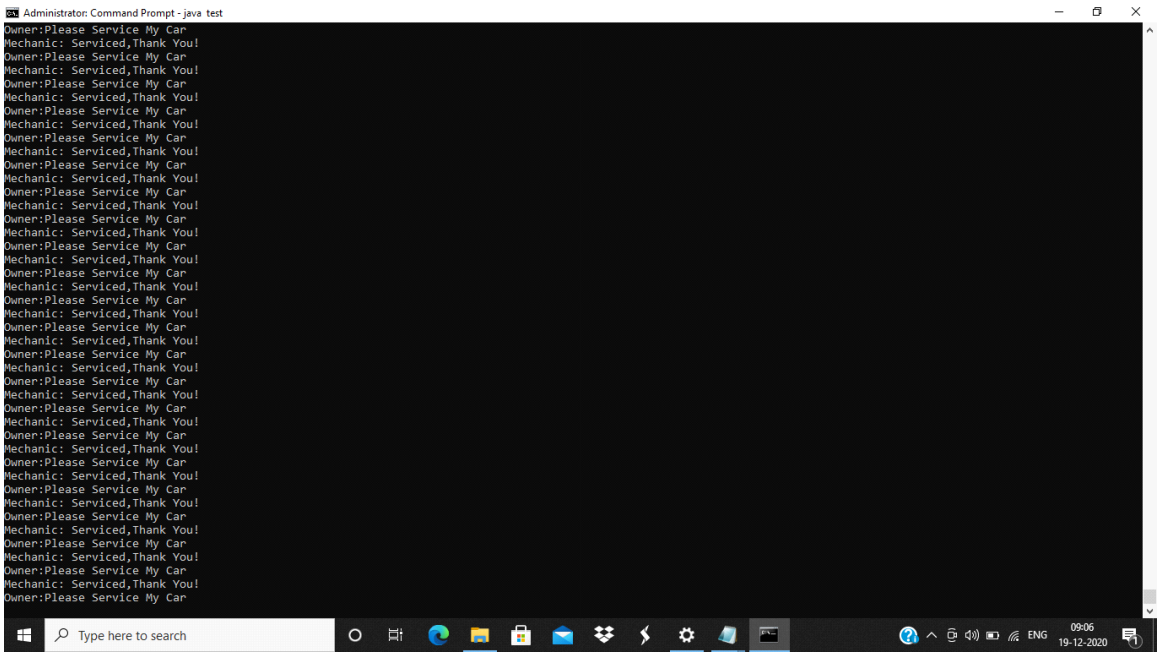
```
import java.util.*;
import java.io.*;
import java.lang.*;
class car_queue
{
        boolean valueset=false;
```

```
synchronized String get()
{
	try
	{
		while(!valueset)
		{
			wait();
		}

	}catch(InterruptedException e){
		System.out.println("Exception Caught");
	}
	System.out.println("Mechanic: Serviced,Thank You!");
	valueset=false;
	notify();
	return "yes";
}
synchronized void put(String msg)
{
	try
	{
		while(valueset)
		{
			wait();
		}
	}catch(InterruptedException e){
		System.out.println("Exception Caught");
	}
```

```
                System.out.println(msg);
                valueset=true;
                notify();
        }
}
class car_owner implements Runnable
{
        Thread t;
        car_queue cq;
        car_owner(car_queue cq)
        {
                this.cq=cq;
                t=new Thread(this,"OWNER");
        }
        public void run()
        {
                while(true)
                {
                        cq.put("Owner:Please Service My Car");
                }
        }
}
class car_mechanic implements Runnable
{
        Thread t;
        car_queue cq;
        car_mechanic(car_queue cq)
        {
```

```
		this.cq=cq;
		t=new Thread(this,"Mechanic");
	}
	public void run()
	{
		while(true)
		{
			cq.get();
		}
	}
}
class test
{
	public static void main(String[] args)
	{
		car_queue cq=new car_queue();
		car_owner co=new car_owner(cq);
		car_mechanic cm=new car_mechanic(cq);
		co.t.start();
		cm.t.start();
		try{
			co.t.join();
			cm.t.join();
		}catch(InterruptedException e){
			System.out.println("Exception Caught");
		}
	}
}
```

Output:



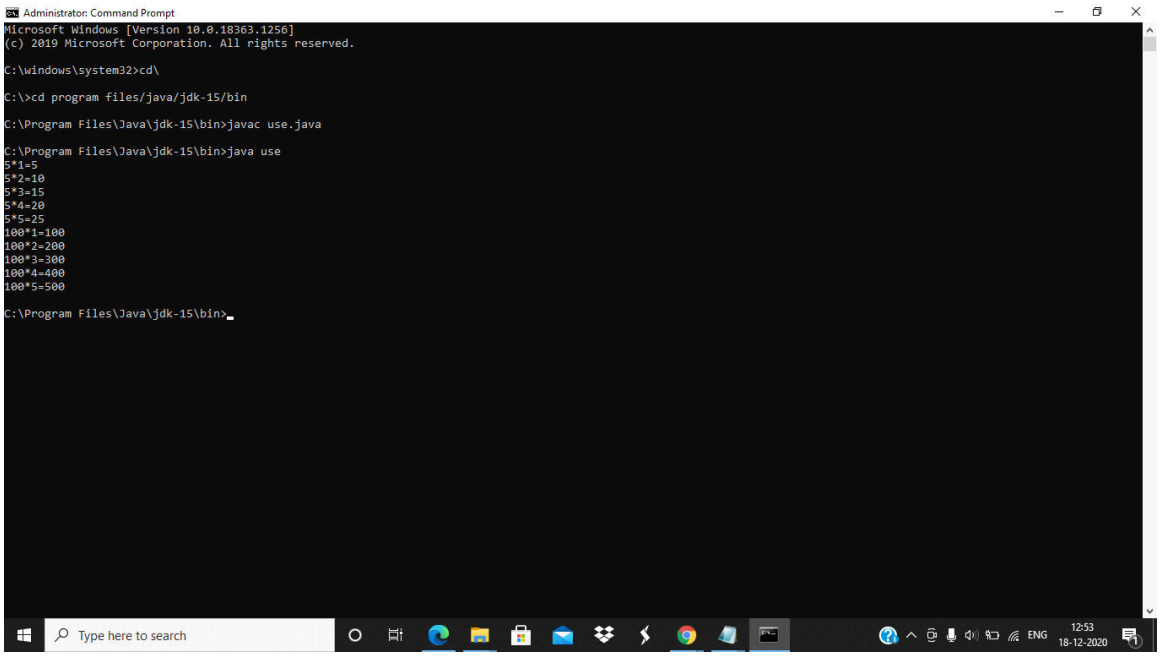
2) multiplaction table:

```
class Table
{
	void printTable(int n)
	{
		synchronized(this)
		{
			for(int i=1;i<=5;i++)
			{
				System.out.println(+n+"*"+i+"="+(n*i));
				try
				{
					Thread.sleep(400);
				}
```

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

class Mythread1 extends Thread
{
        Table t;
        Mythread1(Table t)
        {
                this.t=t;
        }
        public void run()
        {
                t.printTable(5);
        }
}


class Mythread2 extends Thread
{
        Table t;
        Mythread2(Table t)
        {
```

```
                this.t=t;
        }
        public void run()
        {
                t.printTable(100);
        }
}

class use
{
        public static void main(String args[])
        {
                Table obj = new Table();
                Mythread1 th1 = new Mythread1(obj);
                Mythread2 th2 = new Mythread2(obj);
                th1.start();
                th2.start();
        }
}
```

Output:



3)Synchronized:

```
import java.util.*;

import java.io.*;

import java.lang.*;

class callme

{

        synchronized void call(String msg)

        {

                System.out.print("[ " + msg);

                try

                {

                        Thread.sleep(2000);

                }catch(InterruptedException e)

                {

                        System.out.println("Exception Caught");

                }
```

```
                System.out.println("]");
        }
}
class caller implements Runnable
{
        String msg;
        callme target;
        Thread t;
        caller(callme targ,String s)
        {
                target=targ;
                msg=s;
                t=new Thread(this,"SYNC");
        }
        public void run()
        {
                target.call(msg);
        }
}
class sync
{
        public static void main(String[]args)
        {
                callme tt=new callme();
                caller ob1=new caller(tt,"Rohit");
                caller ob2=new caller(tt,"Anil");
                caller ob3=new caller(tt,"Chadichal");
                ob1.t.start();
```

```
                ob2.t.start();
                ob3.t.start();
                try
                {
                        ob1.t.join();
                        ob2.t.join();
                        ob3.t.join();

                }catch(InterruptedException e){
                        System.out.println("Exception Caught");
                }
        }
}
```

Output: