EXERCISE

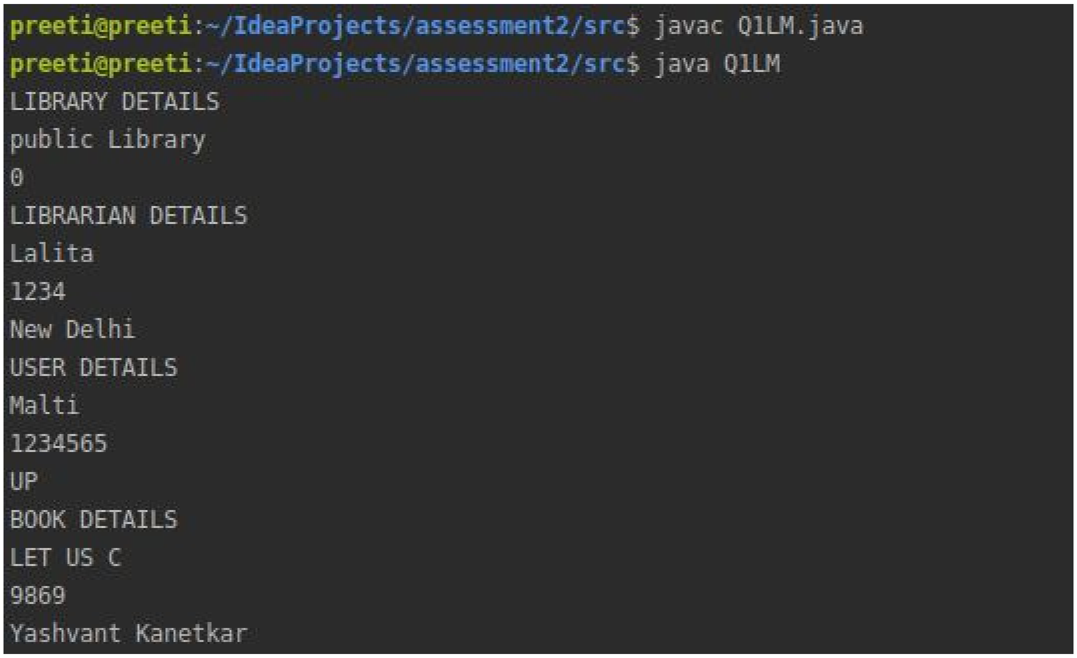
1. Create Java classes having suitable attributes for Library management system.Use OOPs concepts in your design.Also try to use interfaces and abstract classes.

CODE:

```
interface lib
{
   public void getDetail();
   public void setDetail();
}
abstract class Abstract
{
   abstract void setDetails();
   abstract void getDetails();
}

class Library extends Abstract
{
   public String Lib_Name;
   public int Lib_id;

   @Override
   void setDetails()
   {
      Lib_Name="public Library";
      Lib_id=00000;
   }

   @Override
   void getDetails()
   {
      System.out.println(Lib_Name);
      System.out.println(Lib_id);
   }
}
class Librarian implements lib
{
   public String Librarian_Name;
   public int Librarian_id;
   public String Librarian_address;

   @Override
   public void setDetail() {
      Librarian_Name="Lalita";
```

```
        Librarian_id=1234;
        Librarian_address="New Delhi";

    }
    @Override
    public void getDetail() {
        System.out.println(Librarian_Name);
        System.out.println(Librarian_id);
        System.out.println(Librarian_address);

    }
}
class user extends Abstract
{
    public String user_name;
    public int user_id;
    public String user_address;
    @Override
    public void setDetails() {
        user_name="Malti";
        user_id=1234565;
        user_address="UP";

    }
    @Override
    public void getDetails() {
        System.out.println(user_name);
        System.out.println(user_id);
        System.out.println(user_address);

    }
}
class Book implements lib
{
    public String Book_Name;
    public String Book_Author;
    public int Book_Id;
    @Override
    public void setDetail() {
        Book_Name="LET US C";
        Book_Author="Yashvant Kanetkar";
        Book_Id=9869;

    }
    @Override
    public void getDetail() {
        System.out.println(Book_Name);
        System.out.println(Book_Id);
        System.out.println(Book_Author);

    }
}
class Q1LM
{
    public static void main(String arg[])
    {
        Library l=new Library();
        Librarian ll=new Librarian();
        user u=new  user();
        Book b=new Book();
        System.out.println("LIBRARY DETAILS");
        l.setDetails();
        l.getDetails();
        System.out.println("LIBRARIAN DETAILS");
        ll.setDetail();
        ll.getDetail();
        System.out.println("USER DETAILS");
        u.setDetails();
        u.getDetails();
        System.out.println("BOOK DETAILS");
        b.setDetail();
        b.getDetail();
    }
}
```

OUTPUT:



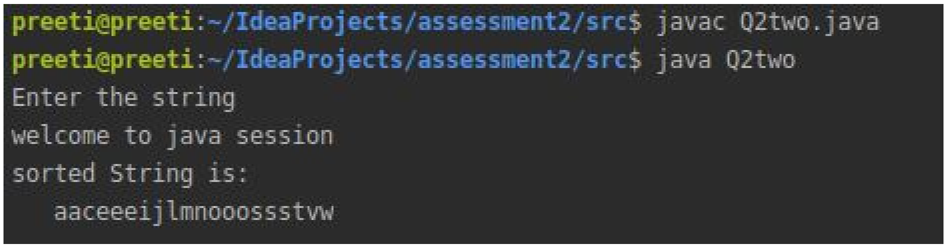
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q1LM.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q1LM
LIBRARY DETAILS
public Library
0
LIBRARIAN DETAILS
Lalita
1234
New Delhi
USER DETAILS
Malti
1234565
UP
BOOK DETAILS
LET US C
9869
Yashvant Kanetkar
```

2. WAP to sorting string without using string Methods?.

CODE:

```
import java.util.Scanner;
public class Q2two
{
        public static void main(String[] args) {
            Scanner sc = new Scanner(System.in);
            System.out.println("Enter the string");
            String s1 = sc.nextLine();
            char ch1[] = new char[s1.length()];
            ch1 = s1.toCharArray();
            //System.out.println(ch1);
            for (int i = 0; i < ch1.length; i++) {
                for (int j = 0; j < ch1.length; j++) {
                    if (ch1[i] < ch1[j]) {
                        char c = ch1[i];
                        ch1[i] = ch1[j];
                        ch1[j] = c;
                    }
                }
            }
            System.out.println("sorted String is:");
            for (int i = 0; i < ch1.length; i++) {
                System.out.print(ch1[i]);
            }
            System.out.println("\n");
        }
    }
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q2two.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q2two
Enter the string
welcome to java session
sorted String is:
   aaceeeijlmnooosssstvw
```

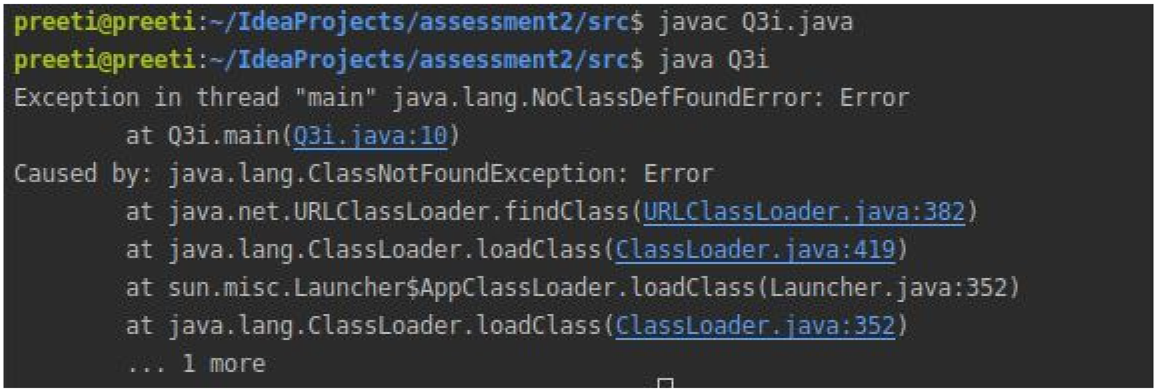
3. WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

NoClassDefFoundError

CODE:

```
class Error{
        void msg()
        {
            System.out.println("hello! everyone");
        }
    }
    class Q3i{
        public static void main(String[] args)
        {
            Error ob = new Error();
            ob.msg();
        }
    }
```

OUTPUT:



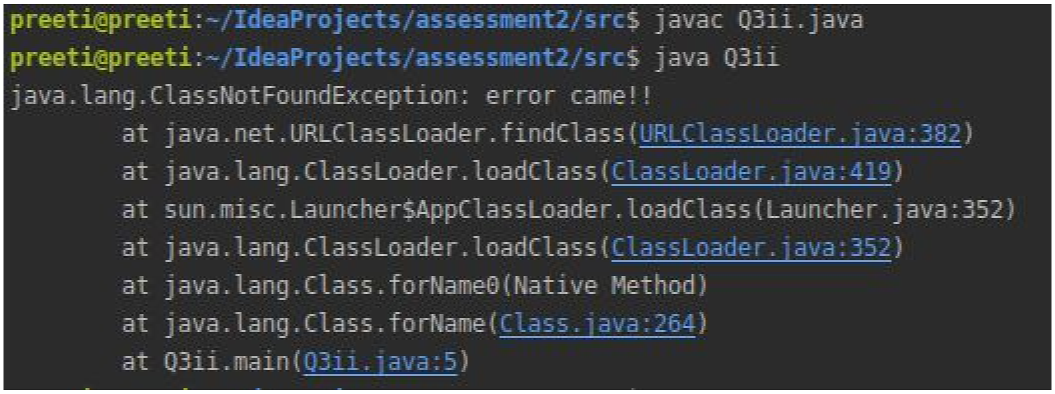
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q3i.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q3i
Exception in thread "main" java.lang.NoClassDefFoundError: Error
        at Q3i.main(Q3i.java:10)
Caused by: java.lang.ClassNotFoundException: Error
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
        ... 1 more
```

ClassNotFoundException

CODE:

```
public class Q3ii {
        public static void main(String args[]) {
            try
            {
                Class.forName("error came!!");
            }
            catch (ClassNotFoundException ex)
            {
                ex.printStackTrace();
            }
        }
}
```

OUTPUT:



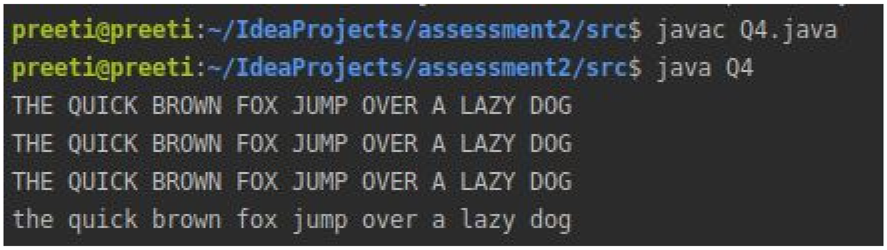
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q3ii.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q3ii
java.lang.ClassNotFoundException: error came!!
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:264)
        at Q3ii.main(Q3ii.java:5)
```

4. WAP to create a singleton class.

CODE:

```
public class Q4 {
        static Q4 s=null;
        String str;
        private Q4()
        {
            str="EVERY CLOUD HAS A SILVER LINING";
        }
        public static Q4 getInstance()
        {
            if(s==null)
            {
                s=new Q4();
            }
            return s;
        }
        public static void main(String arg[])
        {
            Q4 x=Q4.getInstance();
            Q4 y=Q4.getInstance();
            Q4 z=Q4.getInstance();
            x.str=(x.str).toLowerCase();
            System.out.println(x.str);
            System.out.println(y.str);

            y.str=(x.str).toUpperCase();

            System.out.println(x.str);
            System.out.println(y.str);
        }
    }
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q4.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q4
THE QUICK BROWN FOX JUMP OVER A LAZY DOG
THE QUICK BROWN FOX JUMP OVER A LAZY DOG
THE QUICK BROWN FOX JUMP OVER A LAZY DOG
the quick brown fox jump over a lazy dog
```

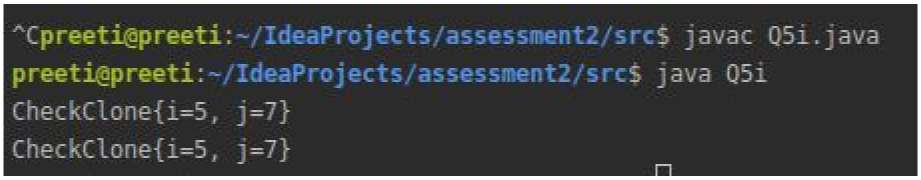
5. WAP to show object cloning in java using cloneable and copy constructor both.

Cloneable:

CODE:

```
class CheckClone implements Cloneable{
    int i;
    int j;

    @Override
    public String toString() {
        return "CheckClone{" + "i=" + i + ", j=" + j + '}';
    }
    @Override
    public Object clone() throws CloneNotSupportedException{
        return super.clone();
    }
}
public class Q5i {
    public static void main(String[] args) throws CloneNotSupportedException{
        CheckClone clone= new CheckClone();
        clone.i=5;
        clone.j=7;

        CheckClone clone1= (CheckClone) clone.clone();

        System.out.println(clone);
        System.out.println(clone1);

    }
}
```

OUTPUT:



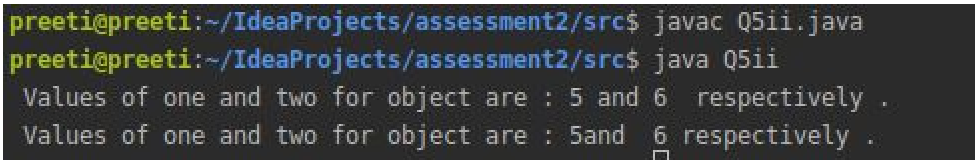
```
^Cpreeti@preeti:~/IdeaProjects/assessment2/src$ javac Q5i.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q5i
CheckClone{i=5, j=7}
CheckClone{i=5, j=7}
```

CONSTRUCTOR:

CODE:

```
public class Q5ii {
    int one, two,one1,two1;
    Q5ii()
    {
        one =5;
        two =6;
    }
    Q5ii(Q5ii sc )
    {  one1 =sc.one;
        two1 =sc.two;
    }
    public static void main(String[] args) {
        Q5ii obj = new Q5ii();
        Q5ii obj2 =new Q5ii(obj);
        System.out.println(" Values of one and two for object are : "+(obj.one) +" and "+(obj.two)+"
respectively .");
        System.out.println(" Values of one and two for object are : "+(obj2.one1) +"and  "+(obj2.two1)+"
respectively .");
    }
}
```

OUTPUT:



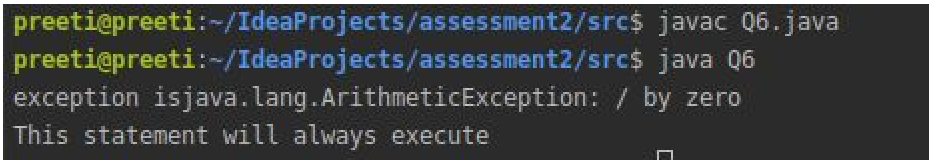
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q5ii.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q5ii
 Values of one and two for object are : 5 and 6  respectively .
 Values of one and two for object are : 5and  6 respectively .
```

6. WAP showing try, multi-catch and finally blocks.

CODE:

```
public class Q6 {
 public static void main(String[] args)
       {
           try
           {
               int a=5,b=0;
               int c=a/b;
           }
           catch(NullPointerException np)
           {
               System.out.println("exception is"+np);
           }
           catch(IllegalArgumentException il)
           {
               System.out.println("exception is"+il);
           }
           catch(ArithmeticException a)
           {
               System.out.println("exception is"+a);
           }
           catch(Exception e)
           {
               System.out.println("exception is there"+e);
           }
           finally {
               System.out.println("This statement will always execute");
           }
       }
   }
```

OUTPUT:



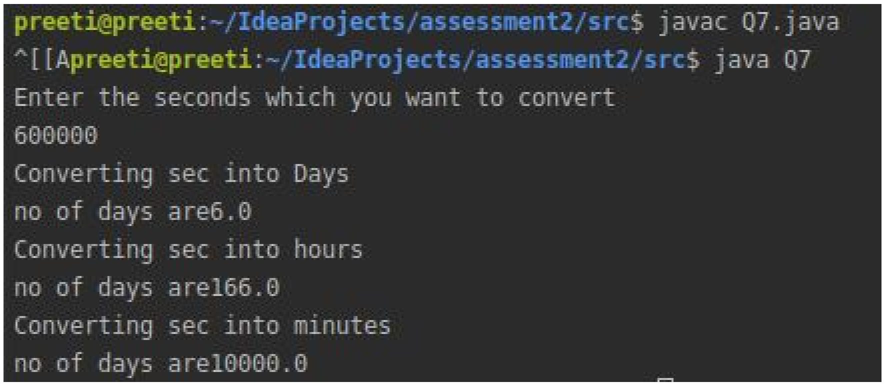
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q6.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q6
exception isjava.lang.ArithmeticException: / by zero
This statement will always execute
```

7. WAP to convert seconds into days, hours, minutes and seconds.

CODE:

```
import java.util.Scanner;
public class Q7 {
        public static void main(String[] args)
        {
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter the seconds which you want to convert");
            int sec=sc.nextInt();
            float days,hours,minutes;
            System.out.println("Converting sec into Days");
            days=sec/(24*60*60);
            System.out.println("no of days are"+days);
            System.out.println("Converting sec into hours");
            hours=sec/(60*60);
            System.out.println("no of days are"+hours);
            System.out.println("Converting sec into minutes");
            minutes=sec/60;
            System.out.println("no of days are"+minutes);
        }
}
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q7.java
^[[Apreeti@preeti:~/IdeaProjects/assessment2/src$ java Q7
Enter the seconds which you want to convert
600000
Converting sec into Days
no of days are6.0
Converting sec into hours
no of days are166.0
Converting sec into minutes
no of days are10000.0
```

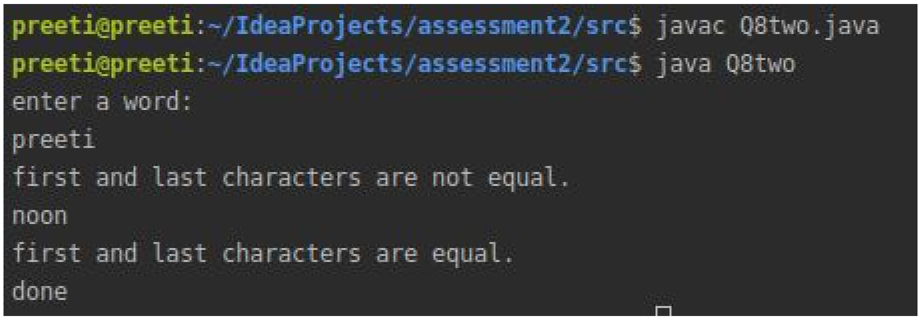
8. WAP to read words from the keyboard until the word ‘done’ is entered. For each word except ‘done’, report whether its first character is equal to its last character. For the required loop, use a

a)while statement

CODE:

```
import java.util.Scanner;

public class Q8two {

    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter a word: ");
        String s1=sc.next();
        while(!s1.equals("done"))
        {
            if(s1.charAt(0)==s1.charAt(s1.length()-1))
            {
                System.out.println("first and last characters are equal.");
            }
            else
            {
                System.out.println("first and last characters are not equal.");
            }
            s1=sc.next();
        }
    }
  }
```

OUTPUT:



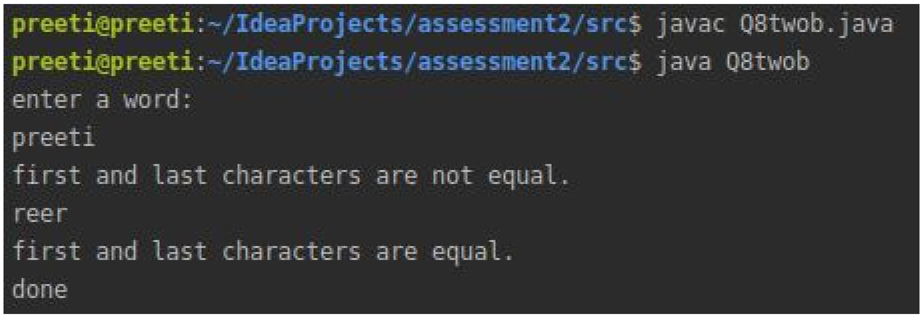
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q8two.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q8two
enter a word:
preeti
first and last characters are not equal.
noon
first and last characters are equal.
done
```

b)do-while statement

CODE:

```
import java.util.Scanner;

public class Q8twob {

    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("enter a word: ");
        String s1=sc.next();
        do{
            if(s1.charAt(0)==s1.charAt(s1.length()-1))
            {
                System.out.println("first and last characters are equal.");
            }
            else
            {
                System.out.println("first and last characters are not equal.");
            }
            s1=sc.next();
        }while(!s1.equals("done"));
    }
}
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q8twob.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q8twob
enter a word:
preeti
first and last characters are not equal.
reer
first and last characters are equal.
done
```

9. Design classes having attributes for furniture where there are wooden chairs and tables, metal chairs and tables. There are stress and fire tests for each products.

CODE:

```
import java.util.Scanner;

interface Furniture {
   public void stressTest();
   public void fireTest();
}

abstract class Chair implements Furniture {
   public abstract String chairType();
}


abstract class Table implements Furniture {
   public abstract String tableType();
}

class MetalChair extends Chair {
   @Override
   public void stressTest() {

   }

   @Override
   public void fireTest() {

   }

   @Override
   public String chairType() {
      String s = "This is a metal Chair";
      return s;
   }
}

class MetalTable extends Table {
   @Override
   public void stressTest() {

      System.out.println("Passed Stress Test");

   }
```

```
    @Override
    public void fireTest() {

        System.out.println("Passed Fire Test");
    }

    @Override
    public String tableType() {

        String s = "This is a metal Table";
        return s;
    }
}


class WoodenTable extends Table {
    @Override
    public void stressTest() {

        System.out.println("Failed Stress Test");
    }

    @Override
    public void fireTest() {

        System.out.println("Failed Fire Test");
    }

    @Override
    public String tableType() {

        String s = "This is a wooden Table";
        return s;
    }
}

class WoodenChair extends Chair {
    @Override
    public void stressTest() {
    }

    @Override
    public void fireTest() {
    }
    @Override
    public String chairType() {
        String s = "This is a wooden chair";
        return s;
    }
}
public class Q9{
    public static void main(String[] args){
        Table table = null;
        Scanner input =  new Scanner(System.in);
        String str = input.next();
        if(str.equals("wooden")){
            table = new WoodenTable();

        }   else if (str.equals("metal")){
            table = new MetalTable();
        }

        System.out.println(table.tableType());
        table.stressTest();
        table.fireTest();

    }
}
```

10. Design classes having attributes and method(only skeleton) for a coffee shop. There are three different actors in our scenario and i have listed the different actions they do also below

*** Customer**

- Pays the cash to the cashier and places his order, get a token number back

- Waits for the intimation that order for his token is ready

- Upon intimation/notification he collects the coffee and enjoys his drink

(Assumption: Customer waits till the coffee is done, he wont timeout and cancel the order. Customer always likes the drink served. Exceptions like he not liking his coffee, he getting wrong coffee are not considered to keep the design simple.)

*** Cashier**

- Takes an order and payment from the customer

- Upon payment, creates an order and places it into the order queue

- Intimates the customer that he has to wait for his token and gives him his token

(Assumption: Token returned to the customer is the order id. Order queue is unlimited. With a simple modification, we can design for a limited queue size)

*** Barista**

- Gets the next order from the queue

- Prepares the coffee

- Places the coffee in the completed order queue

- Places a notification that order for token is ready

CODE:

Cashier.java

```
public class Cashier
{
    String cashierName;
    int CashId;

    public String getCashierName() {
        return cashierName;
    }

    public void setcashierName(String cashierName) {
        this.cashierName = cashierName;
    }
    public int getCashId() {
        return CashId;
    }

    public void setCashierId(int cashierId) {
        CashId = cashId;
    }

    public void takeOrder(String cname)
    {
      //cashier will take the order
    }
    public void giveTokenNo(String cname)
    {
        //this will be called when customer places an order so cashier will provide him with a tokenno
    }
    public void receivePayment(int ctnum)
    {
        //
    }
    public void addItToPendingQueue(int ctnum)
    {
    }

}
```

Barista.java

```
public class Barista
{
    public void getOrderFromPendingQueue()
    {
//get order from the pending queue
    }
    public void prepareOrder()
    {
//prepare customer order
    }
    public void CompletedOrderQueue()
    {
//insert order into complete order queue
    }
}
```

Customer.java

```
public class Customer
{
    private String cname;
    private double cphone;
    private int ctnum;
    Customer(String cname,double cphone)
    {
        this.cname=cname;
        this.phone=phone;
    }
    public String getName() {
        return cname;
    }

    public void setName(String cname) {
        this.cname = cname;
    }

    public double getPhone() {
        return cphone;
    }

    public void setPhone(double cphone) {
        this.cphone = cphone;
    }

    public int getTokeno() {
        return ctnum;
    }

    public void setTokeno(int ctnum) {
        this.ctnum = ctnum;
    }
    public void Order()
    {

        //we will place order on the basis of name and phone number
    }
    public void OrderStatus(int tokeno)
    {

        // we will checkorderstatus on the basis of tokenno given to the customer
    }
    public void Ordercollect(int tokeno)
    {
        // customer can collect the coffee after entering tokenno
    }
    public void Charges()
    {
        //payment done through this function
    }



}
```

11. Convert the following code so that it uses nested while statements instead of for statements:

```
int s = 0;

int t = 1;

for (int i = 0; i < 10; i++)

{

s = s + i;

for (int j = i; j > 0; j−−)

{

t = t * (j - i);

}

s = s * t;

System.out.println("T is " + t);

}

System.out.println("S is " + s);
```

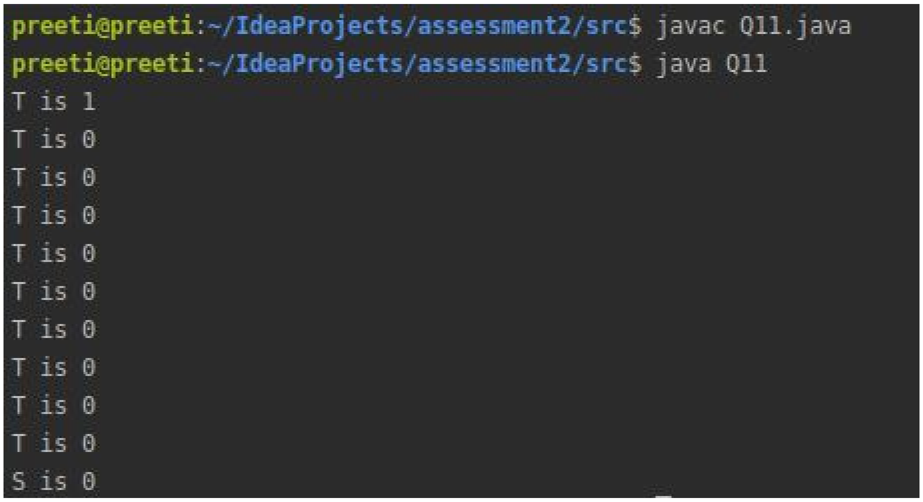
CODE:

NESTED FOR LOOP:

```
public class Q11 {
    public static void main(String[] args) {
        int s = 0;
        int t = 1;
        int i=0;
        while(i<10) {
            s = s + i;
            int j=i;
            while (j>0)
            {
                t = t * (j - i);
                j--;
            }
            s = s * t;
            System.out.println("T is " + t);
            i++;
        }
        System.out.println("S is " + s);
    }
```

```
  }
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q11.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q11
T is 1
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
T is 0
S is 0
```

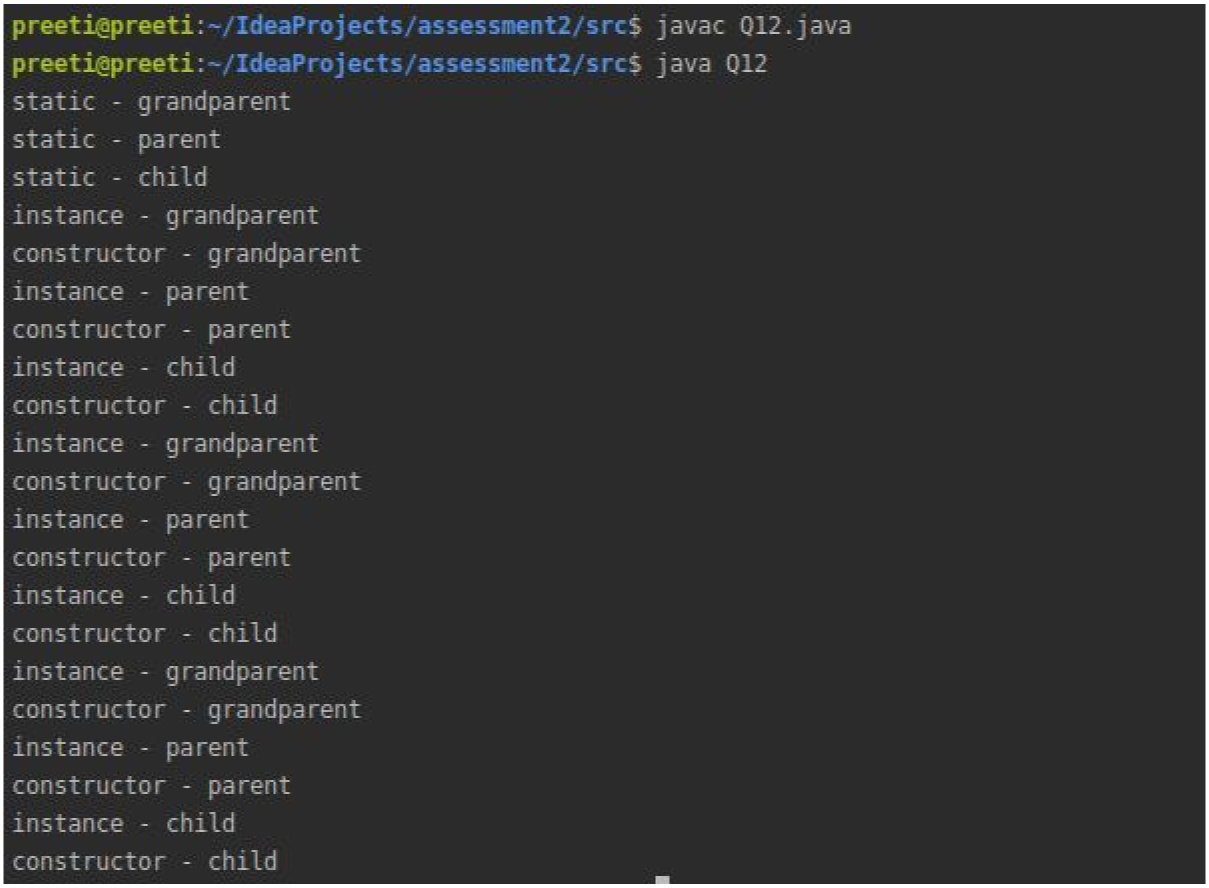
12.What will be the output on new Child(); ?

```
class Parent extends Grandparent {

    {
        System.out.println("instance - parent");
    }
    public Parent() {
        System.out.println("constructor - parent");
    }
    static {
        System.out.println("static - parent");
    }
}
class Grandparent {

    static {
        System.out.println("static - grandparent");
    }
    {
        System.out.println("instance - grandparent");
    }
    public Grandparent() {
        System.out.println("constructor - grandparent");
    }
}
class Child extends Parent {
    public Child() {
        System.out.println("constructor - child");
    }
    static {
        System.out.println("static - child");
    }
    {
        System.out.println("instance - child");
    }
}
```

CODE:

```
class Parent extends Grandparent {
    {
        System.out.println("instance - parent");
    }
    public Parent() {
        System.out.println("constructor - parent");
    }
    static {
        System.out.println("static - parent");
    }
}
class Grandparent {
    static {
        System.out.println("static - grandparent");
    }
    {
        System.out.println("instance - grandparent");
    }
    public Grandparent() {
        System.out.println("constructor - grandparent");
    }
}
class Child extends Parent {
    public Child() {
        System.out.println("constructor - child");
    }
    static {
        System.out.println("static - child");
    }
    {
        System.out.println("instance - child");
    }
}
class Q12
{
    public static void main(String[] args) {
        Parent p=new Child();
        Grandparent g=new Child();
        Child c=new Child();
    }
}
```

OUTPUT:



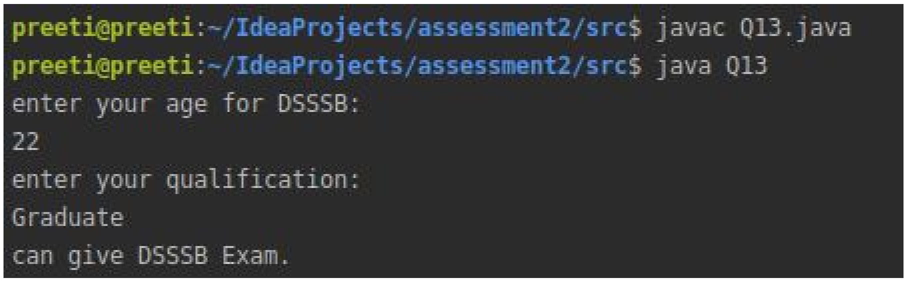
```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q12.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q12
static - grandparent
static - parent
static - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child
instance - grandparent
constructor - grandparent
instance - parent
constructor - parent
instance - child
constructor - child
```

13. Create a custom exception that does not have any stack trace.

CODE:

```
import java.util.Scanner;
public class Q13 {
    public static void main(String[] args) throws NotEligiblee
    {
        System.out.println("enter your age for DSSSB:");
        Scanner sc=new Scanner(System.in);
        int age=sc.nextInt();
        if(age<18)
        {
            throw new NotEligiblee("not eligible");
        }
        else
        {
            System.out.println("enter your qualification:");
            String s=sc.next();
            if(!s.equalsIgnoreCase("Graduate"))
            {
                throw new NotEligiblee("not eligible in qualification");
            }
            else
                System.out.println("can give DSSSB Exam.");
        }
    }
}
class NotEligiblee extends Exception{
    NotEligiblee(String s)
    {
        super(s);
    }
}
```

OUTPUT:



```
preeti@preeti:~/IdeaProjects/assessment2/src$ javac Q13.java
preeti@preeti:~/IdeaProjects/assessment2/src$ java Q13
enter your age for DSSSB:
22
enter your qualification:
Graduate
can give DSSSB Exam.
```