JAVA2 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:

Abstract Class

```
package com.Q1LibraryManagement;



public abstract class Abstract

{

   abstract void setDetails();

   abstract void getDetails();

}
```

Interface

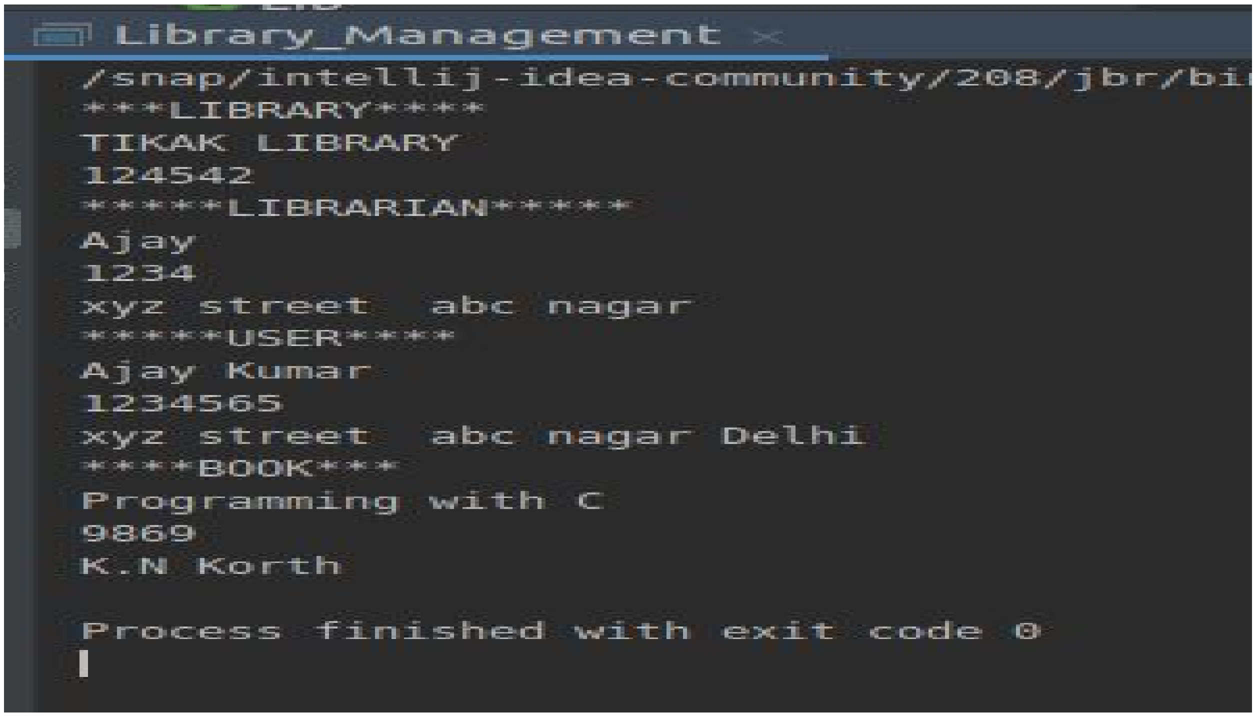
```
package com.Q1LibraryManagement;

   public interface Lib

   {

       public void getDetail();

       public void setDetail();

   }
```

Library Management

```
package com.Q1LibraryManagement;

class Library extends Abstract

{

   public String Library_Name;
```

```
    public int Library_id;


    @Override
    void setDetails()
    {
        Library_Name="TIKAK LIBRARY";
        Library_id=124542;
    }


    @Override
    void getDetails()
    {
        System.out.println(Library_Name);
        System.out.println(Library_id);
    }
}
class Librarian implements Lib
{
    public String Librarian_Name;
    public int Librarian_id;
    public String Librarian_address;


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

        Librarian_id=1234;

        Librarian_address="xyz street  abc nagar";


    }

    @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="Ajay Kumar";

        user_id=1234565;

        user_address="xyz street  abc nagar Delhi";
    }
    @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="Programming with C";
        Book_Author="K.N Korth";
        Book_Id=9869;

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

        System.out.println(Book_Author);


    }

}

public class Library_Management

{

    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****");

        l.setDetails();

        l.getDetails();

        System.out.println("*****LIBRARIAN*****");

        ll.setDetail();

        ll.getDetail();

        System.out.println("*****USER****");

        u.setDetails();

        u.getDetails();

        System.out.println("****BOOK***");

        b.setDetail();
        b.getDetail();
    }
}
```

Output



```
Library_Management
/snap/intellij-idea-community/208/jbr/bi
***LIBRARY****
TIKAK LIBRARY
124542
*****LIBRARIAN*****
Ajay
1234
xyz street  abc nagar
*****USER****
Ajay Kumar
1234565
xyz street  abc nagar Delhi
****BOOK***
Programming with C
9869
K.N Korth

Process finished with exit code 0
```

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

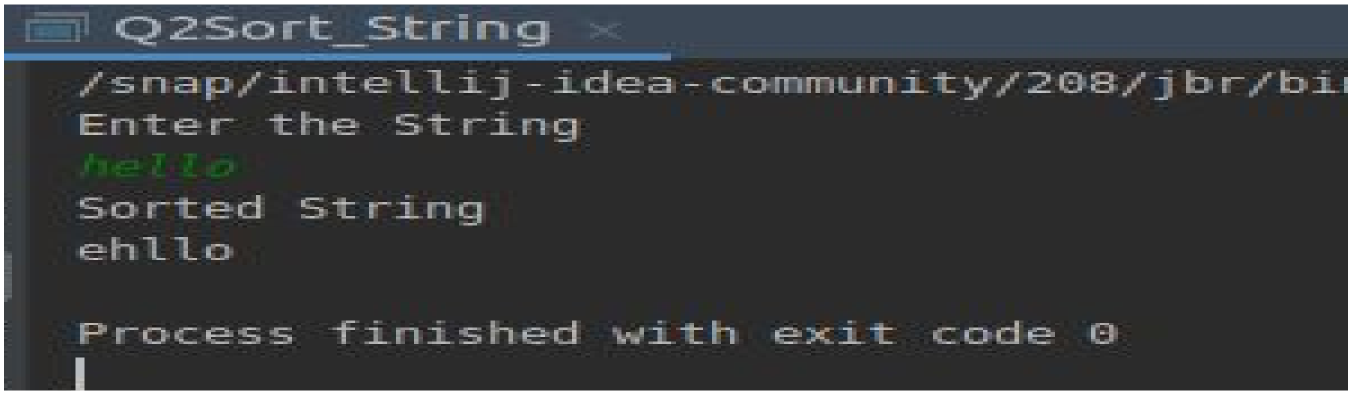
Code:

```
import java.util.*;

public class Q2Sort_String {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the String");

        String str = scan.next();

        int j=0;
```

```
char temp;

String ch = "";


char[] chars = str.toCharArray();


for (int i = 0; i <chars.length; i++) {


    for ( j = 0; j < chars.length; j++) {


        if(chars[j]>chars[i]){

            temp=chars[i];

            chars[i]=chars[j];

            chars[j]=temp;

        }


    }


}


for(int k=0;k<chars.length;k++){

    ch = ch + chars[k];

}

System.out.println("Sorted String");

System.out.println(ch);
```

```
  }

}
```

Output



```
Q2Sort_String
/snap/intellij-idea-community/208/jbr/bi
Enter the String
hello
Sorted String
ehllo

Process finished with exit code 0
```

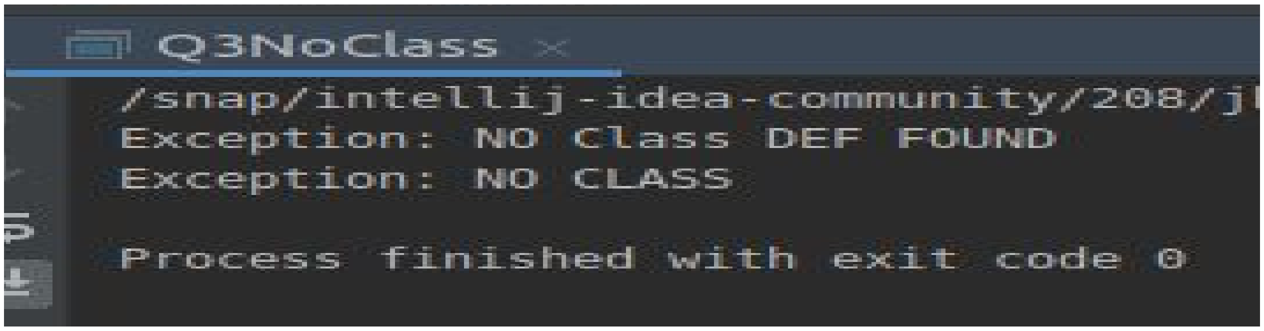
3. WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

Code:

```
public class Q3NoClass

{

  public static void main(String args[])

  {



      try

      {

         throw new NoClassDefFoundError("NO Class DEF FOUND");

      }

      catch (NoClassDefFoundError n)

      {

         System.out.println("Exception: "+n.getMessage());

      }
```

```
        try

        {

            throw new ClassNotFoundException("NO CLASS");

        }

        catch (ClassNotFoundException c)

        {

            System.out.println("Exception: "+c.getMessage());

        }


    }

}
```

<u>Output</u>



```
Q3NoClass
/snap/intellij-idea-community/208/j
Exception: NO Class DEF FOUND
Exception: NO CLASS

Process finished with exit code 0
```

4. WAP to create singleton class.

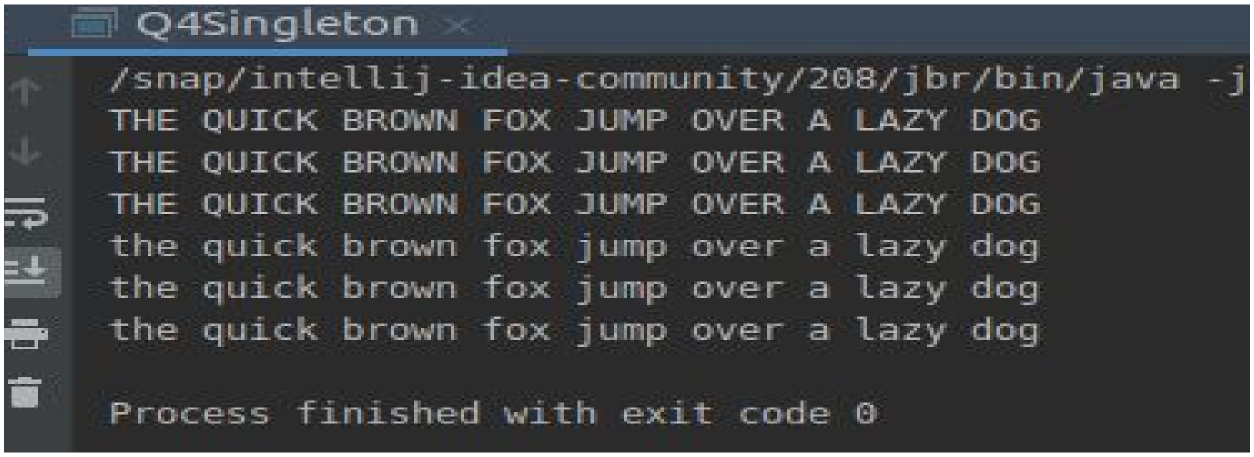
Code:

```
public class Q4Singleton

{

    static Q4Singleton s=null;

    String str;
```

```
private Q4Singleton()

{

    str="the quick brown fox jump over a lazy dog";

}

public static Q4Singleton getInstance()

{

    if(s==null)

    {

        s=new Q4Singleton();

    }

    return s;

}

public static void main(String arg[])

{

    Q4Singleton x= Q4Singleton.getInstance();

    Q4Singleton y= Q4Singleton.getInstance();

    Q4Singleton z= Q4Singleton.getInstance();

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

    System.out.println(x.str);

    System.out.println(y.str);

    System.out.println(z.str);



    y.str=(x.str).toLowerCase();
```

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

        System.out.println(y.str);

        System.out.println(z.str);

    }

}
```

<u>Output</u>



Q4Singleton

```
/snap/intellij-idea-community/208/jbr/bin/java -j
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
the quick brown fox jump over a lazy dog
the quick brown fox jump over a lazy dog

Process finished with exit code 0
```

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

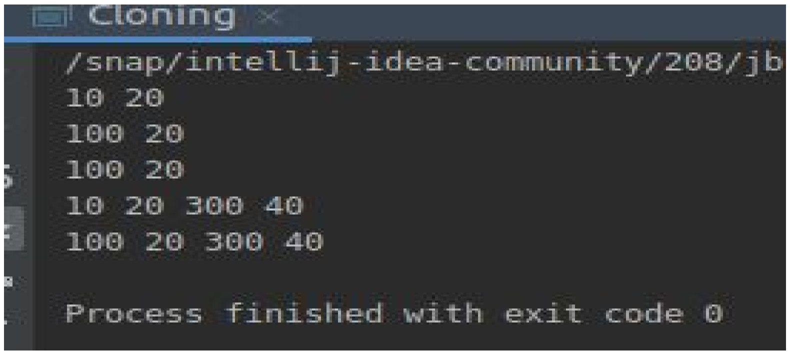
Code:

```
class Copy

{

    int x, y;

    Copy()

    {

        x = 10;

        y = 20;

    }

}
```

```
class Clone

{

   int x, y;

}

class Clone2 implements Cloneable

{

   int a;

   int b;

   Clone c = new Clone();

   public Object clone() throws

           CloneNotSupportedException

   {

       return super.clone();

   }


}

public class Q5Cloning

{

   public static void main(String[] args) throws

           CloneNotSupportedException

   {

       Copy c = new Copy();


       System.out.println(c.x + " " + c.y);
```

```
        Copy c1 = c;

        c1.x = 100;


        System.out.println(c.x+" "+c.y);

        System.out.println(c1.x+" "+c1.y);


        Clone2 t1 = new Clone2();

        t1.a = 10;

        t1.b = 20;

        t1.c.x = 30;

        t1.c.y = 40;


        Clone2 t2 = (Clone2)t1.clone();

        t2.a = 100;

        t2.c.x = 300;

        System.out.println(t1.a + " " + t1.b + " " +t1.c.x + " " + t1.c.y);

        System.out.println(t2.a + " " + t2.b + " " + t2.c.x + " " + t2.c.y);


    }

  }
```

Output


```
/snap/intellij-idea-community/208/jb
10 20
100 20
100 20
10 20 300 40
100 20 300 40

Process finished with exit code 0
```

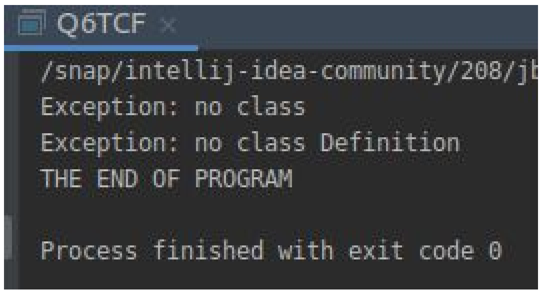
6. WAP showing try, multi-catch and finally blocks.

Code:

```
public class Q6TCF {

  public static void main (String args[])

  {

      try

      {

          throw new ClassNotFoundException("no class");

      }

      catch (ClassNotFoundException c)

      {

          System.out.println("Exception: "+c.getMessage());

      }

      try
```

```
        {
            throw new NoClassDefFoundError("no class Definition");
        }
        catch (NoClassDefFoundError n)
        {
            System.out.println("Exception: "+n.getMessage());
        }
        finally {
            System.out.println("THE END OF PROGRAM");
        }
    }
}
```

Output



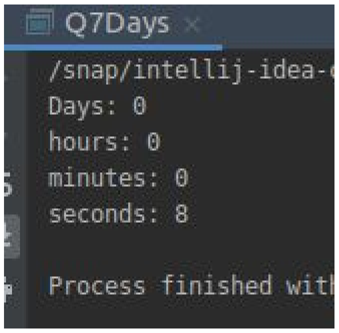
```
/snap/intellij-idea-community/208/jb
Exception: no class
Exception: no class Definition
THE END OF PROGRAM

Process finished with exit code 0
```

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

Code:

```
public class Q7Days {


        public static void main(String args[]) {
            int k = 8;
            int h = 0, m = 0, s = 0, d = 0;

            d = k / 86400;

            System.out.println("Days: " + d);

            d= k % 86400;

            h= d / 3600;

            System.out.println("hours: " + h);

            h=d % 3600;

            m=h/60;

            System.out.println("minutes: " +m);

            m=h % 60;

            s=m;

            System.out.println("seconds: "+s);

        }

    }
```

Output



```
Q7Days
/snap/intellij-idea-
Days: 0
hours: 0
minutes: 0
seconds: 8

Process finished wit
```

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

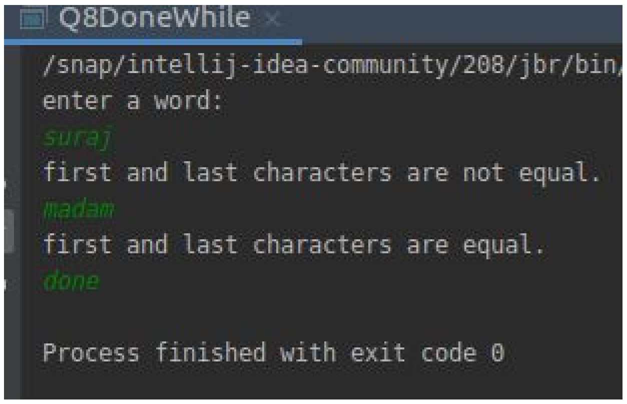
a)while statement

b)do-while statement

Code with While Statement:

```
import java.util.*;
public class Q8DoneWhile {

   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();
       }
   }
   }
```

<u>Output</u>

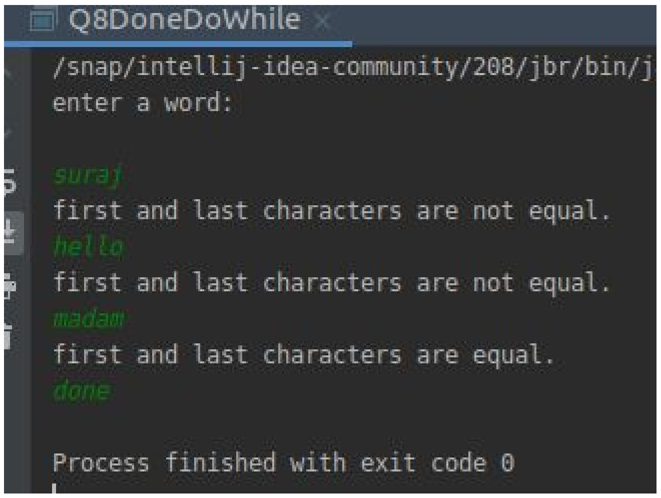


```
/snap/intellij-idea-community/208/jbr/bin,
enter a word:
suraj
first and last characters are not equal.
madam
first and last characters are equal.
done

Process finished with exit code 0
```

<u>Code with DoWhile Statement</u>

```
import java.util.Scanner;
public class Q8DoneDoWhile{
    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



Q8DoneDoWhile

```
/snap/intellij-idea-community/208/jbr/bin/j
enter a word:

suraj
first and last characters are not equal.
hello
first and last characters are not equal.
madam
first and last characters are equal.
done

Process finished with exit code 0
```

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:

<u>Chair Class</u>

```
package com.Q9Furniture;

public class Chair {

      public String Type;

      public String StressProof_;

      public String FireProof;

}
```

<u>Chairs Class</u>

```
package com.Q9Furniture;

class MetalChairs extends Chair

{

   {

      Type = "metal chair";

      StressProof_= "true";

      FireProof = "true";

   }

   void getDetails()

   {

      System.out.println(Type);

      System.out.println("Stress Proof: "+StressProof_);
```

```
        System.out.println("Fire Proof: "+FireProof);

    }

}

class WoodenChairs extends Chair

{

    {

        Type = "Wooden chair";

        StressProof_= "true";

        FireProof = "Noooo";

    }

    void getDetails()

    {

        System.out.println(Type);

        System.out.println("Stress Proof: "+StressProof_);

        System.out.println("Fire Proof: "+FireProof);

    }

}

class Chairs

{

    public static void main(String args[])

    {

        System.out.println("****METAL CHAIR DETAIL*********");

        MetalChairs m=new MetalChairs();

        m.getDetails();
        System.out.println("****WOODEN CHAIR DETAIL*********");

        WoodenChairs w=new WoodenChairs();

        w.getDetails();

    }

}
```

Table Class

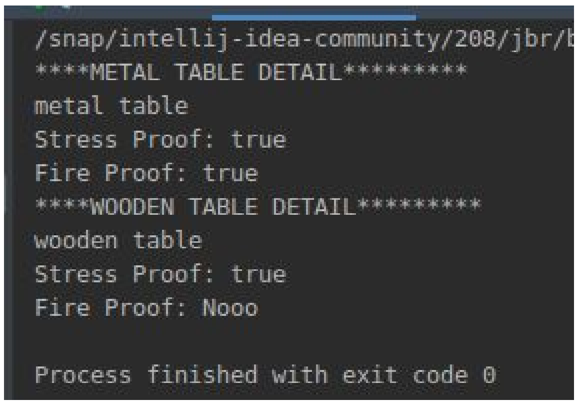
```
package com.Q9Furniture;

public class Table {

   public String Type;

   public String StressProof_;

   public String FireProof;

}
```

Tables Class

```
package com.Q9Furniture;

class MetalTables extends Table
{
    {
        Type = "metal table";
        StressProof_= "true";
        FireProof = "true";
    }
    void getDetails()
    {
        System.out.println(Type);
        System.out.println("Stress Proof: "+StressProof_);
        System.out.println("Fire Proof: "+FireProof);
    }
}
```

```
class WoodenTables extends Table {
    {
        Type = "wooden table";
        StressProof_ = "true";
        FireProof = "Nooo";
    }
    void getDetails()
    {
        System.out.println(Type);
        System.out.println("Stress Proof: "+StressProof_);
        System.out.println("Fire Proof: "+FireProof);
    }
}
class Tables
{
    public static void main(String args[])
    {
        System.out.println("****METAL TABLE DETAIL*********");
        MetalTables m=new MetalTables();
        m.getDetails();
        System.out.println("****WOODEN TABLE DETAIL*********");
        WoodenTables w=new WoodenTables();
        w.getDetails();
    }
}
```

Output



```
/snap/intellij-idea-community/208/jbr/b
****METAL TABLE DETAIL*********
metal table
Stress Proof: true
Fire Proof: true
****WOODEN TABLE DETAIL*********
wooden table
Stress Proof: true
Fire Proof: Nooo

Process finished with exit code 0
```

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:

<u>Barista Class</u>

```
package Q10Coffee;

   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
        }
    }
```

Cashier Class

```
package Q10Coffee;

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 = cashierId;
        }
        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)
        {
        }
    }
```

Customer Class

```
package Q10Coffee;

public class Customer
{
    private String cname;
    private double cphone;
    private int ctnum;
    Customer(String cname,double cphone)
    {
        this.cname=cname;
        this.cphone= cphone;
    }
    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:

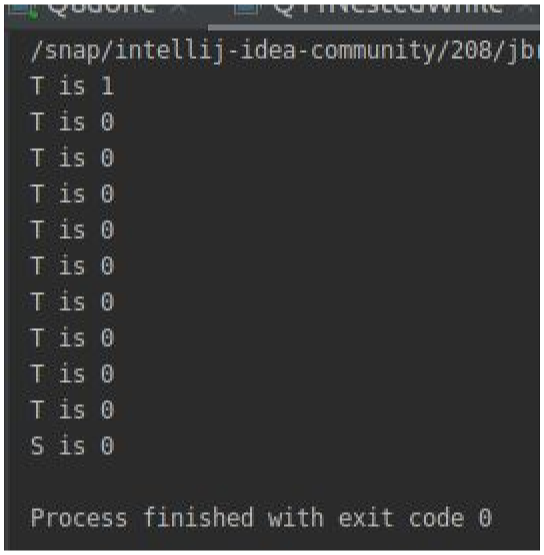
```
    int s = 0;

    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:

```
public class Q11NestedWhile
{
    public static void main(String args[])
    {
        int i,j;
        int s = 0;
        int t = 1;
        i = 0;
        while(i<10)
        {
            s = s + i;
            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);
    }
}
```

<u>**Output**</u>



```
/snap/intellij-idea-community/208/jb
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

Process finished with exit code 0
```

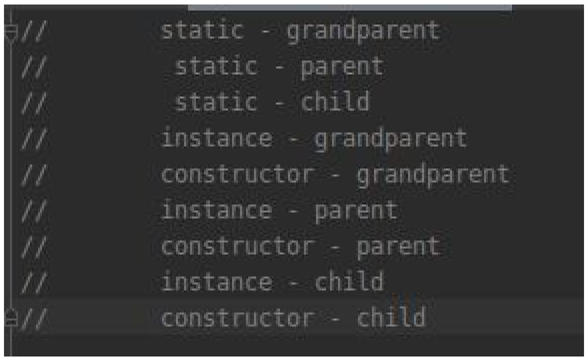
12.What will be the output on new Child(); ?

```
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");

   }
```

```
  }
```

Output



```
//        static - grandparent
//         static - parent
//         static - child
//        instance - grandparent
//        constructor - grandparent
//        instance - parent
//        constructor - parent
//        instance - child
//        constructor - child
```

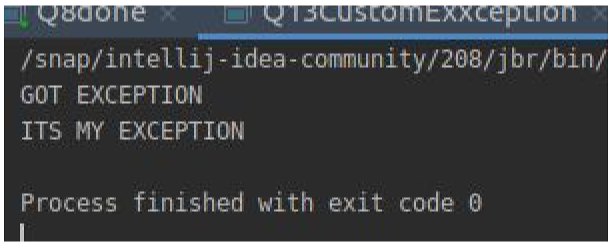
Q13. Create a custom exception that do not have any stack trace.

Code:

<u>Custom EXCEPTION CLASS</u>

```
class MyException extends Exception
{
   MyException(String s)
   {
      super(s);
   }
}
public class Q13CustomExxception {
   public static void main(String args[]) {
      try {
         throw new MyException("ITS MY EXCEPTION");
      } catch (MyException ex) {
         System.out.println("GOT EXCEPTION");
         System.out.println(ex.getMessage());
      }
   }
}
```

<u>Output</u>



```
/snap/intellij-idea-community/208/jbr/bin/
GOT EXCEPTION
ITS MY EXCEPTION

Process finished with exit code 0
```