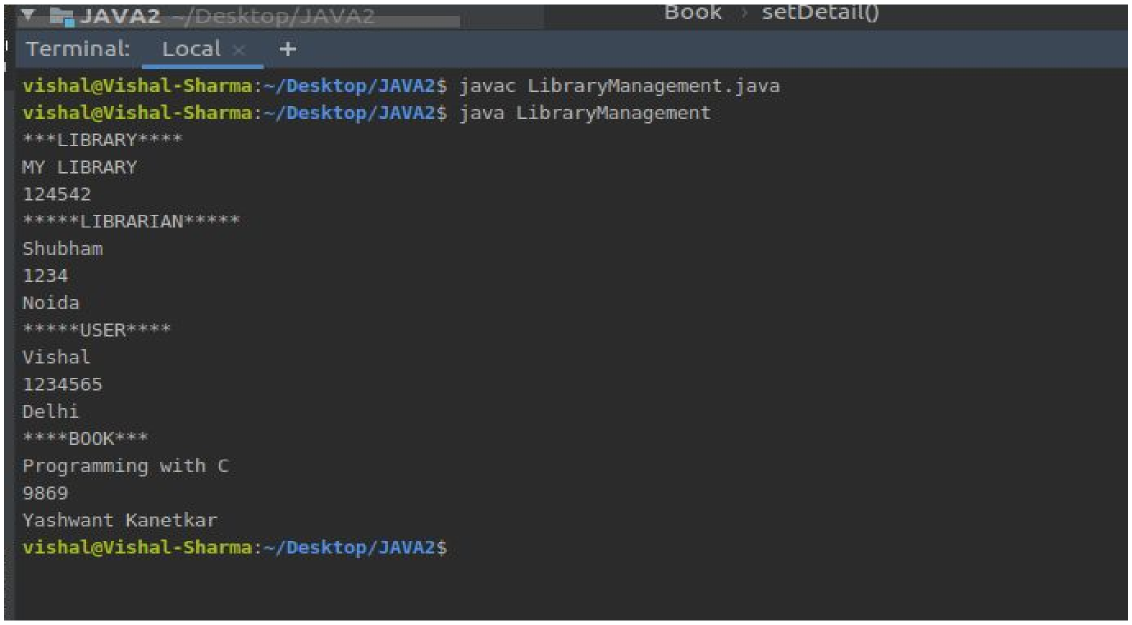
Newer : Vishal Sharma
Newer ID : 4171
Email : vishal.sharma@tothenew.com

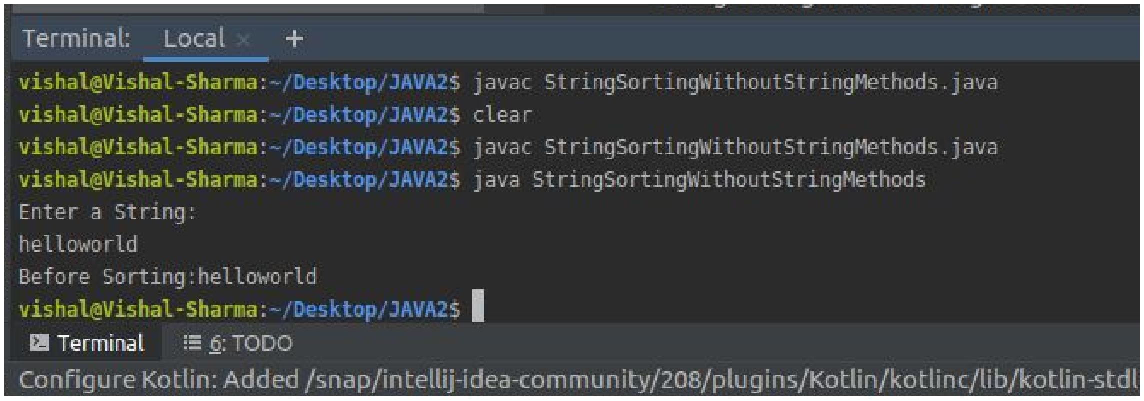
Exercise : Introduction to Java 2

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.
Answer: Filename - **LibraryManagement.java** , **Abstract.java** , **Lib.java**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac LibraryManagement.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java LibraryManagement
***LIBRARY****
MY LIBRARY
124542
*****LIBRARIAN*****
Shubham
1234
Noida
*****USER****
Vishal
1234565
Delhi
****BOOK***
Programming with C
9869
Yashwant Kanetkar
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

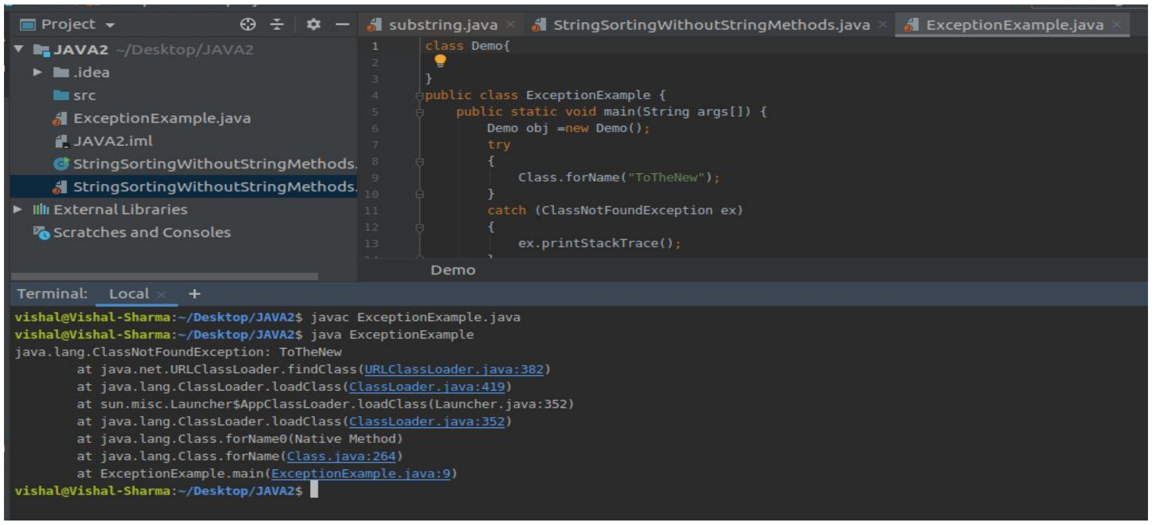
2) WAP to sorting string without using string Methods?
Answer: Filename - **StringSortingWithoutStringMethods.java**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac StringSortingWithoutStringMethods.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ clear
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac StringSortingWithoutStringMethods.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java StringSortingWithoutStringMethods
Enter a String:
helloworld
Before Sorting:helloworld
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

Configure Kotlin: Added /snap/intellij-idea-community/208/plugins/Kotlin/kotlinc/lib/kotlin-stdl

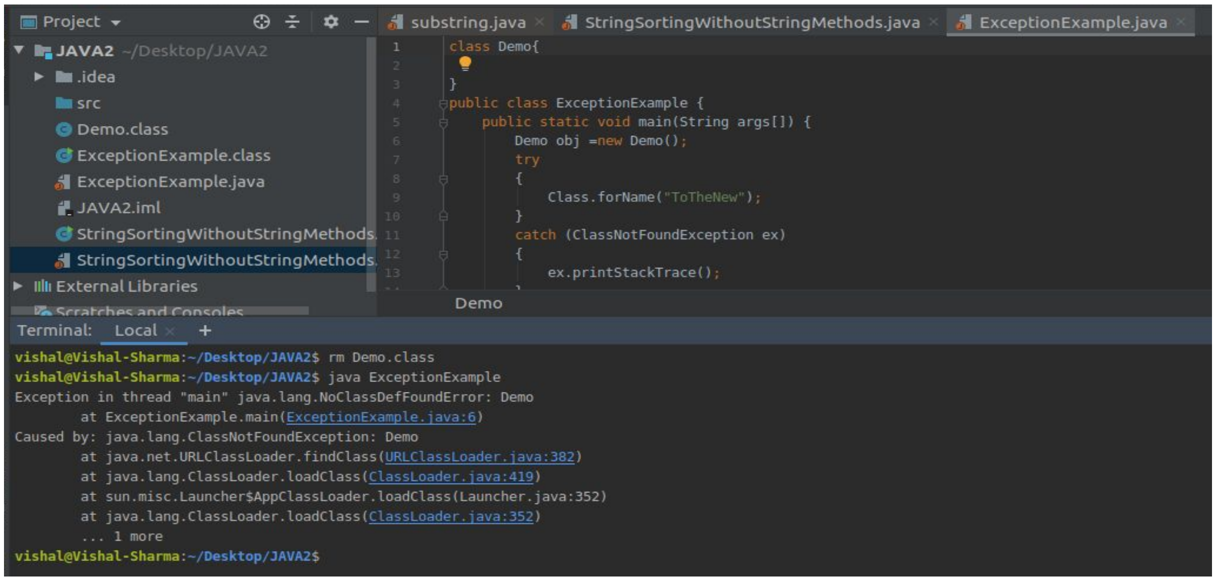
3) WAP to produce NoClassDefFoundError and ClassNotFoundException exception.
Answer: Filename - **ExceptionExample.java**



```java
class Demo{

}
public class ExceptionExample {
    public static void main(String args[]) {
        Demo obj =new Demo();
        try
        {
            Class.forName("ToTheNew");
        }
        catch (ClassNotFoundException ex)
        {
            ex.printStackTrace();
```

```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac ExceptionExample.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java ExceptionExample
java.lang.ClassNotFoundException: ToTheNew
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
        at java.lang.Class.forName0(Native Method)
        at java.lang.Class.forName(Class.java:264)
        at ExceptionExample.main(ExceptionExample.java:9)
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

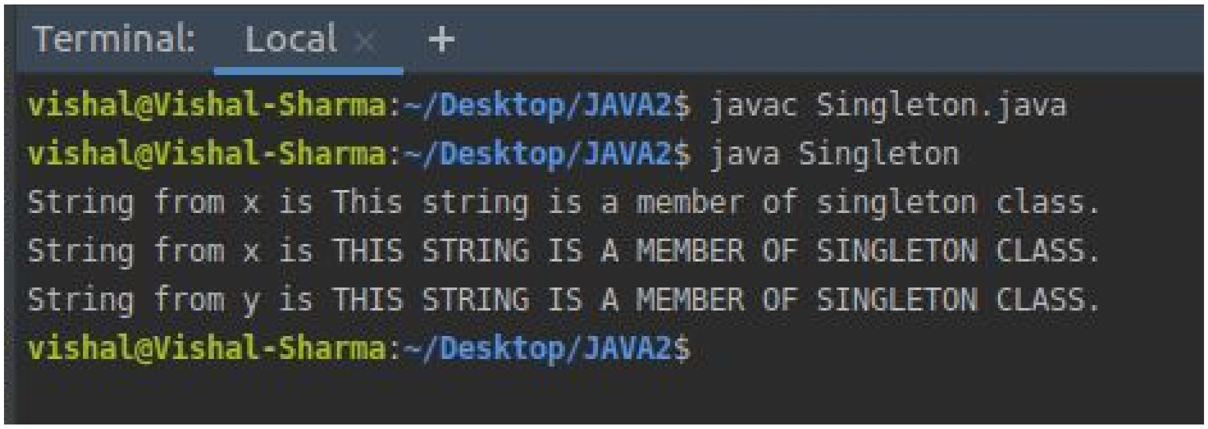


```java
class Demo{

}
public class ExceptionExample {
    public static void main(String args[]) {
        Demo obj =new Demo();
        try
        {
            Class.forName("ToTheNew");
        }
        catch (ClassNotFoundException ex)
        {
            ex.printStackTrace();
```

```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ rm Demo.class
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java ExceptionExample
Exception in thread "main" java.lang.NoClassDefFoundError: Demo
        at ExceptionExample.main(ExceptionExample.java:6)
Caused by: java.lang.ClassNotFoundException: Demo
        at java.net.URLClassLoader.findClass(URLClassLoader.java:382)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:419)
        at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:352)
        at java.lang.ClassLoader.loadClass(ClassLoader.java:352)
        ... 1 more
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

4) WAP to create singleton class.
Answer: Filename - **Singleton.java**

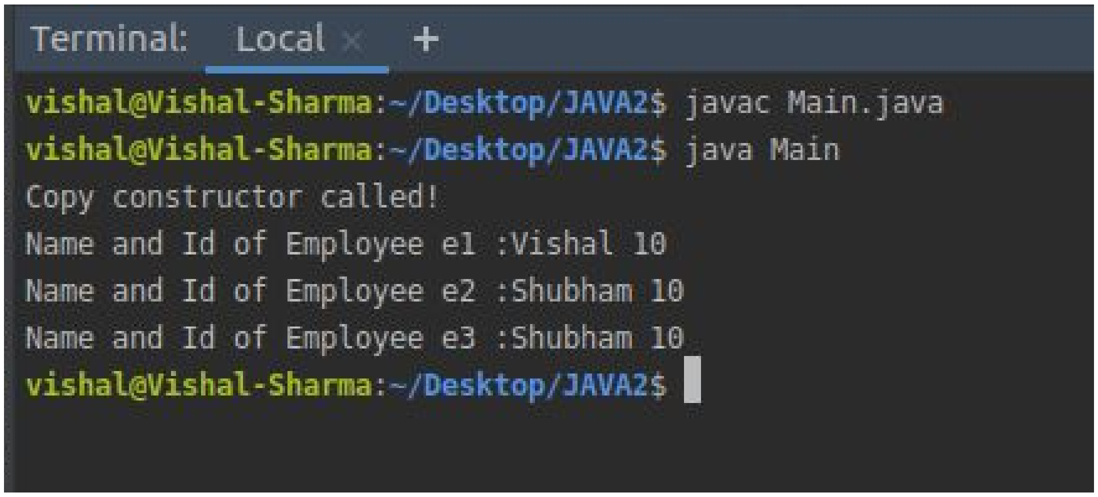


```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac Singleton.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java Singleton
String from x is This string is a member of singleton class.
String from x is THIS STRING IS A MEMBER OF SINGLETON CLASS.
String from y is THIS STRING IS A MEMBER OF SINGLETON CLASS.
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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

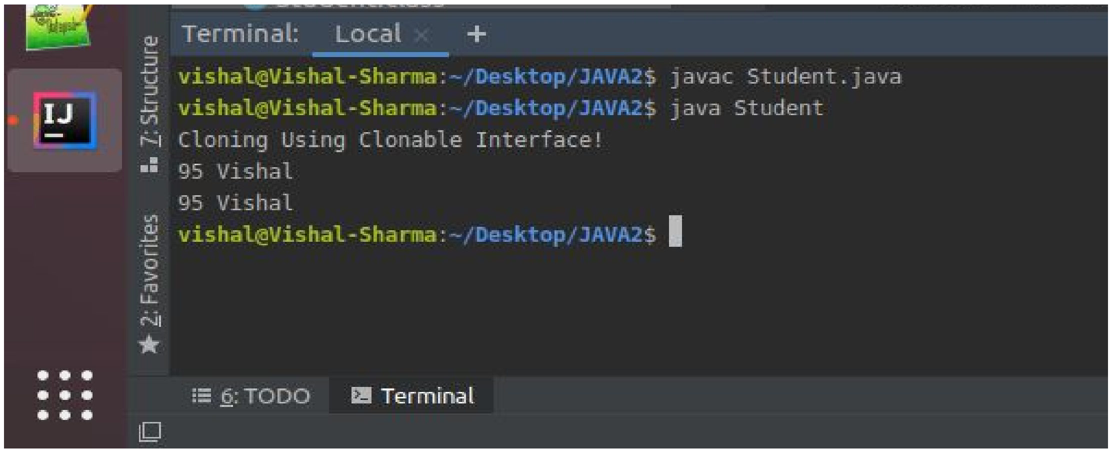
Answer: Filename - **Student.java** and **Main.java**

Using Copy Constructor Filename - **Main.java**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac Main.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java Main
Copy constructor called!
Name and Id of Employee e1 :Vishal 10
Name and Id of Employee e2 :Shubham 10
Name and Id of Employee e3 :Shubham 10
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

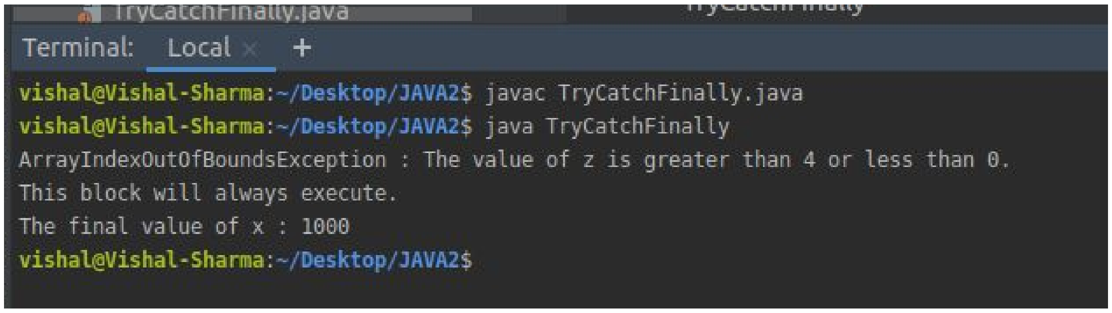
Using Clonable Interface Filename - **Student.java**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac Student.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java Student
Cloning Using Clonable Interface!
95 Vishal
95 Vishal
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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

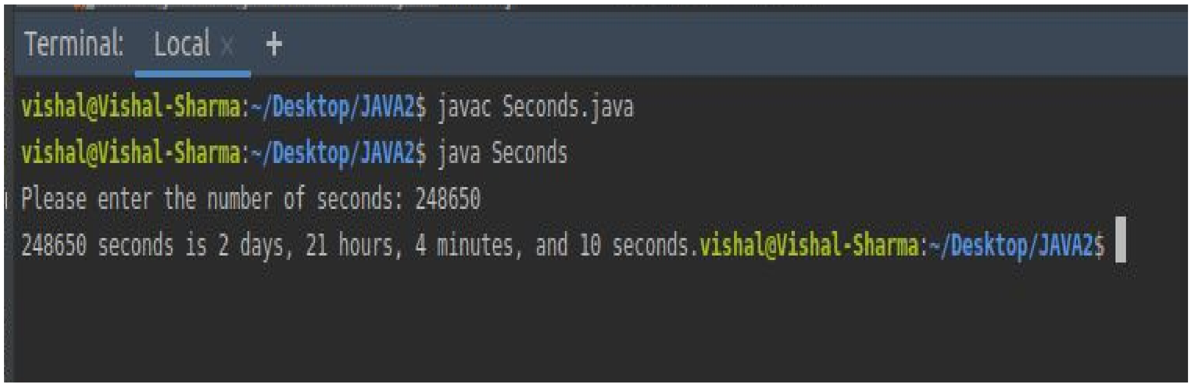
   Answer: Filename - **TryCatchFinally.java**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac TryCatchFinally.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java TryCatchFinally
ArrayIndexOutOfBoundsException : The value of z is greater than 4 or less than 0.
This block will always execute.
The final value of x : 1000
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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

   Answer: Filename - **Seconds.java**



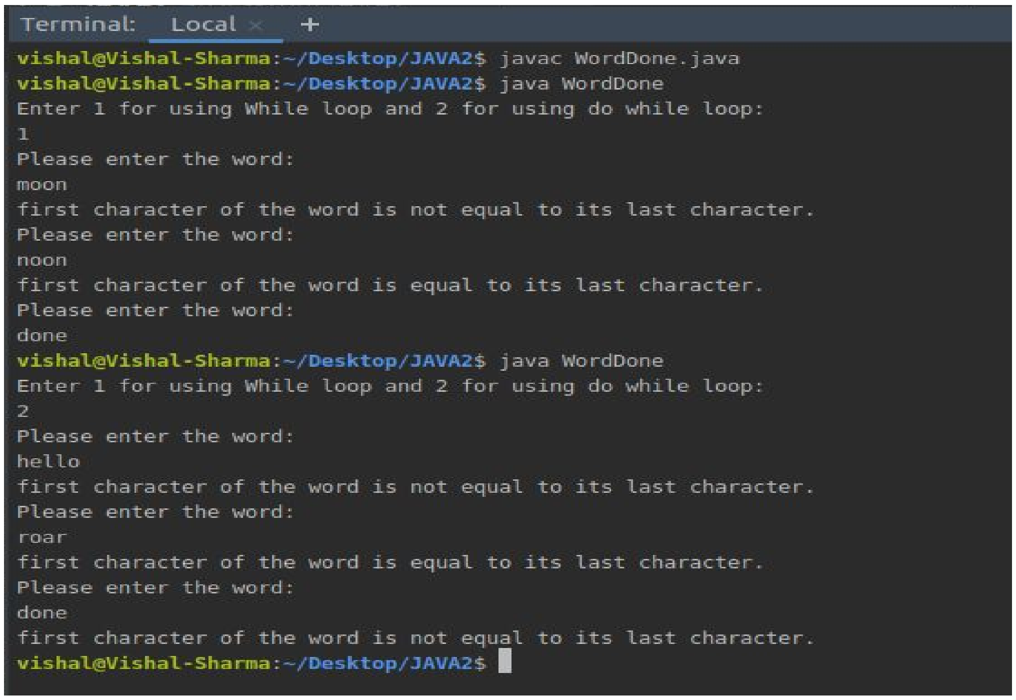
```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac Seconds.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java Seconds
Please enter the number of seconds: 248650
248650 seconds is 2 days, 21 hours, 4 minutes, and 10 seconds.vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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

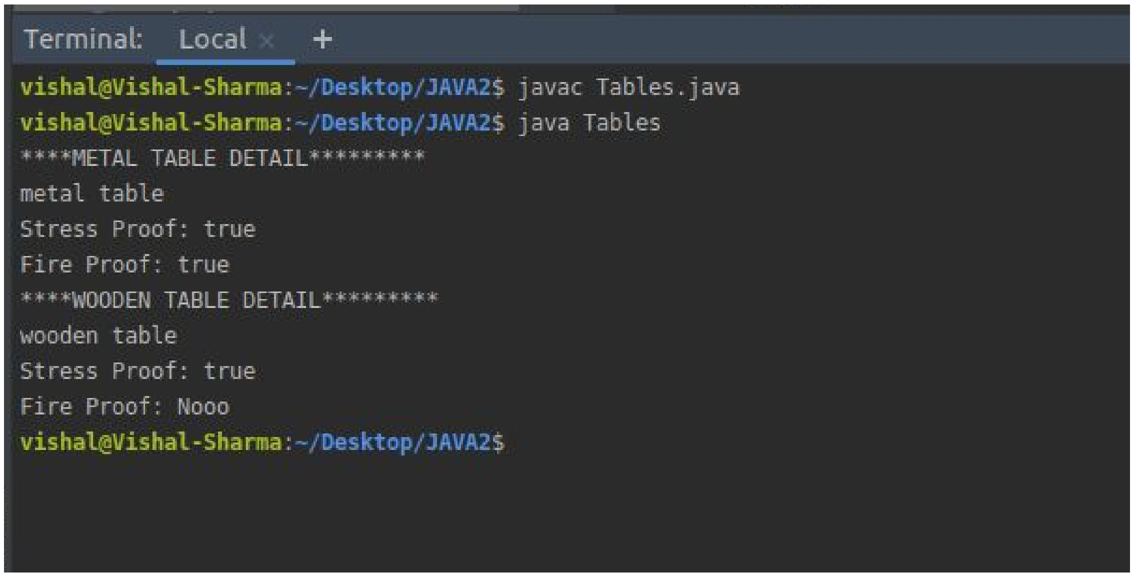
   Answer: Filename - **<u>WordDone.java</u>**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac WordDone.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java WordDone
Enter 1 for using While loop and 2 for using do while loop:
1
Please enter the word:
moon
first character of the word is not equal to its last character.
Please enter the word:
noon
first character of the word is equal to its last character.
Please enter the word:
done
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java WordDone
Enter 1 for using While loop and 2 for using do while loop:
2
Please enter the word:
hello
first character of the word is not equal to its last character.
Please enter the word:
roar
first character of the word is equal to its last character.
Please enter the word:
done
first character of the word is not equal to its last character.
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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.

   Answer: Filename - **<u>Tables.java</u>** , **<u>Table.java</u>** , **<u>Chairs.java</u>** , **<u>Chair.java</u>**



Terminal: Local

```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac Tables.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java Tables
****METAL TABLE DETAIL*********
metal table
Stress Proof: true
Fire Proof: true
****WOODEN TABLE DETAIL*********
wooden table
Stress Proof: true
Fire Proof: Nooo
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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

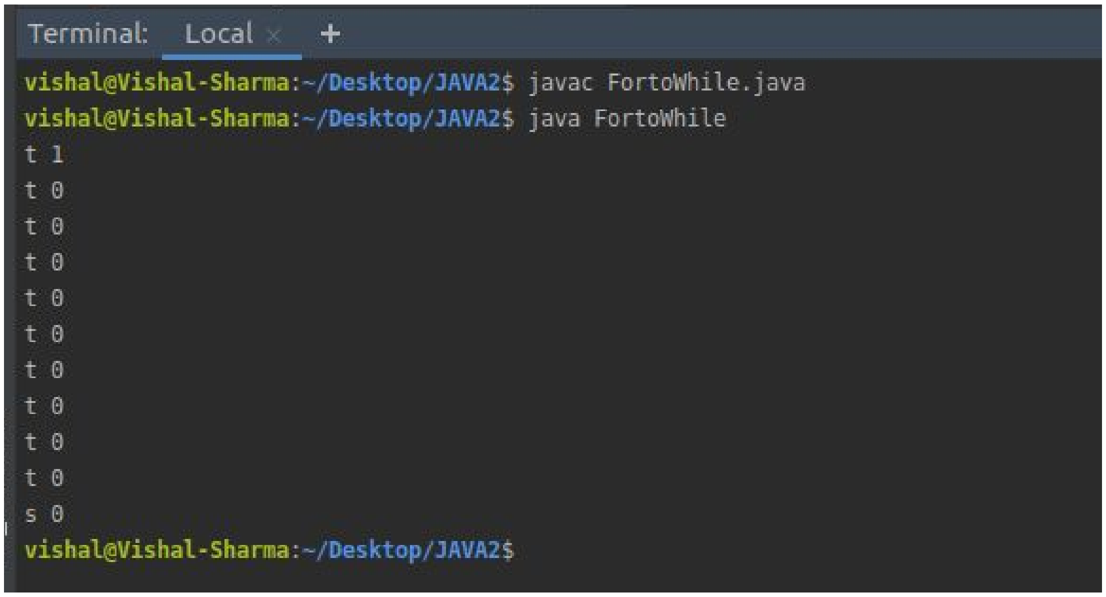
- Gets the next order from the queue
- Prepares the coffee
- Places the coffee in the completed order queue
- Places a notification that order for token is ready

Answer: Filename - **Coffee.java**

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);
```

Answer: Filename - **FortoWhile.java**


```
Terminal:   Local ×   +
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac FortoWhile.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java FortoWhile
t 1
t 0
t 0
t 0
t 0
t 0
t 0
t 0
t 0
t 0
s 0
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```

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");

    }

}
```

Answer:

static - grandparent

static - parent

static - child

instance - grandparent

constructor - grandparent

instance - parent

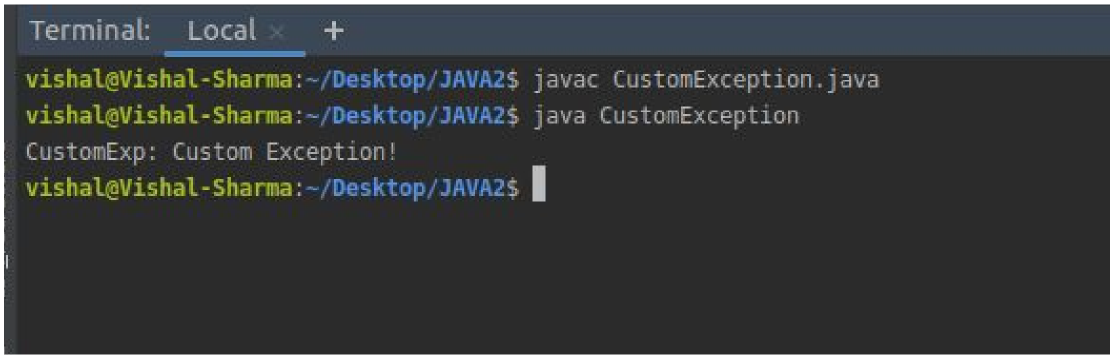
constructor - parent

instance - child

constructor - child

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

Answer: Filename - **<u>CustomException.java</u>**



```
vishal@Vishal-Sharma:~/Desktop/JAVA2$ javac CustomException.java
vishal@Vishal-Sharma:~/Desktop/JAVA2$ java CustomException
CustomExp: Custom Exception!
vishal@Vishal-Sharma:~/Desktop/JAVA2$
```