Introduction To Java2

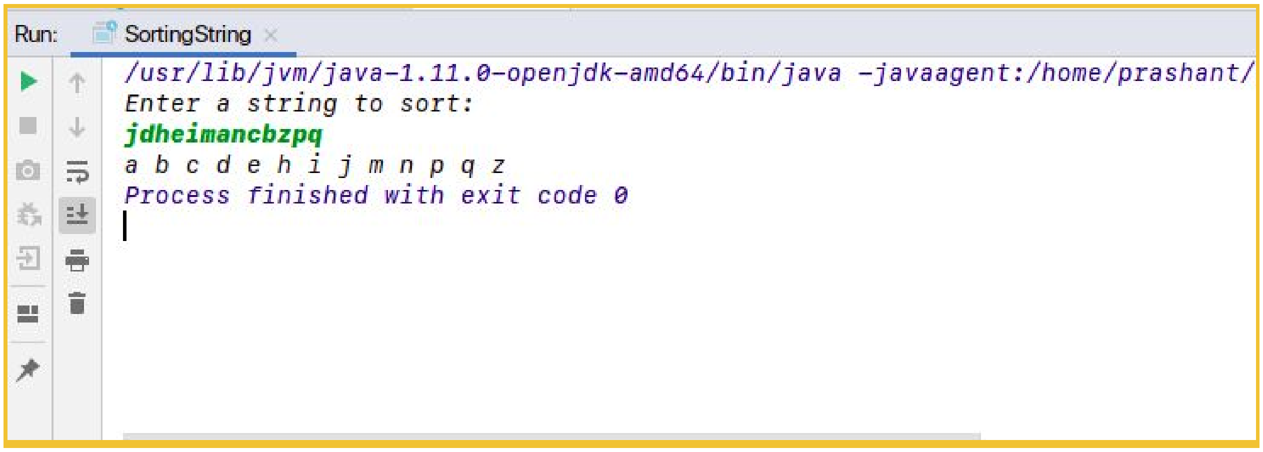
Newer ID - 4092

Email - prashant.brahmbhatt@tothenew.com

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.

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

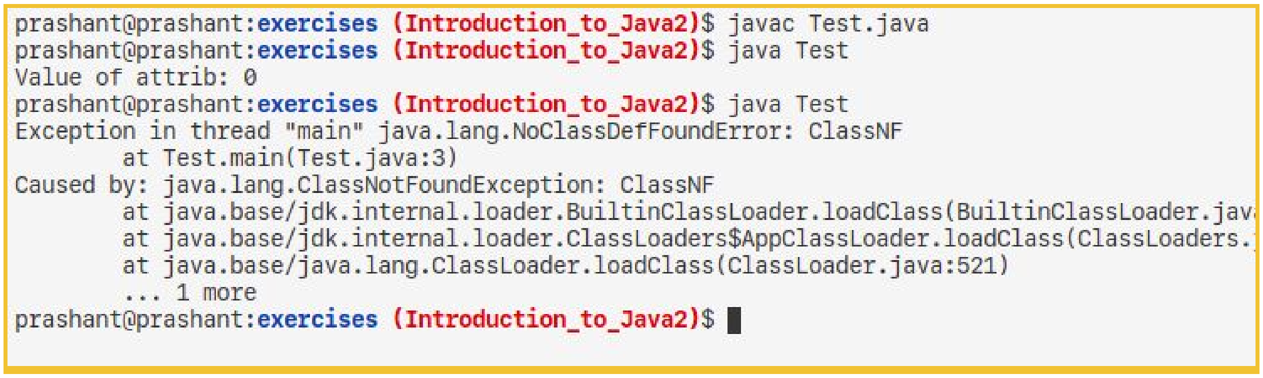
ANS: Filename - SortingString.java



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/
Enter a string to sort:
jdheimancbzpq
a b c d e h i j m n p q z
Process finished with exit code 0
```

3.) WAP to produce NoClassDefFoundError and ClassNotFoundException exception.

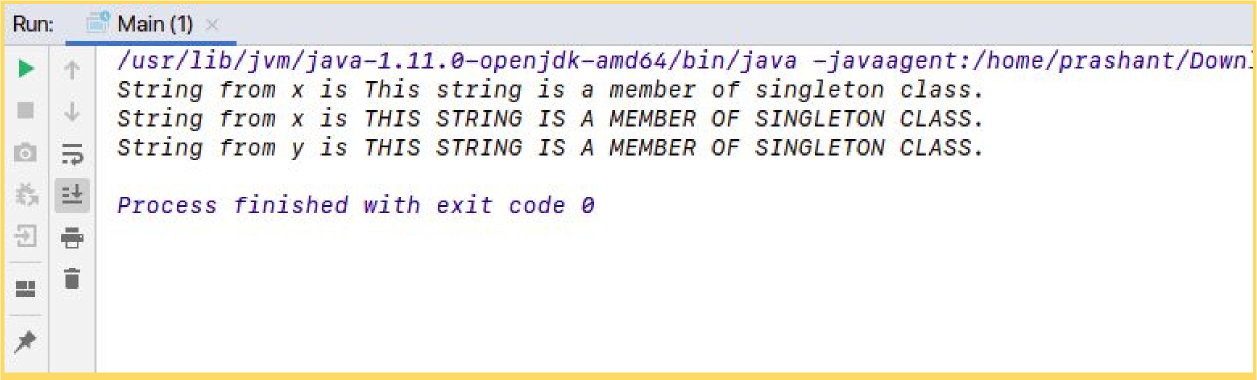
ANS: Filename - ClassExps.java



```
prashant@prashant:exercises (Introduction_to_Java2)$ javac Test.java
prashant@prashant:exercises (Introduction_to_Java2)$ java Test
Value of attrib: 0
prashant@prashant:exercises (Introduction_to_Java2)$ java Test
Exception in thread "main" java.lang.NoClassDefFoundError: ClassNF
        at Test.main(Test.java:3)
Caused by: java.lang.ClassNotFoundException: ClassNF
        at java.base/jdk.internal.loader.BuiltinClassLoader.loadClass(BuiltinClassLoader.jav
        at java.base/jdk.internal.loader.ClassLoaders$AppClassLoader.loadClass(ClassLoaders.
        at java.base/java.lang.ClassLoader.loadClass(ClassLoader.java:521)
        ... 1 more
prashant@prashant:exercises (Introduction_to_Java2)$
```

4.) WAP to create singleton class.

ANS: Filename - Singleton.java



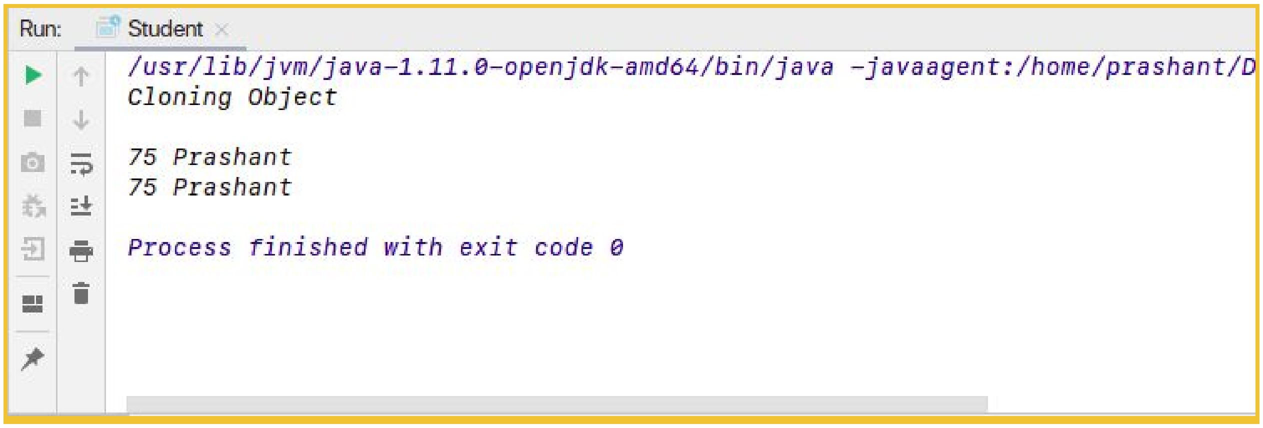
```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/Downl
String from x is This string is a member of singleton class.
String from x is THIS STRING IS A MEMBER OF SINGLETON CLASS.
String from y is THIS STRING IS A MEMBER OF SINGLETON CLASS.

Process finished with exit code 0
```

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

ANS: Filename - Cloning.java

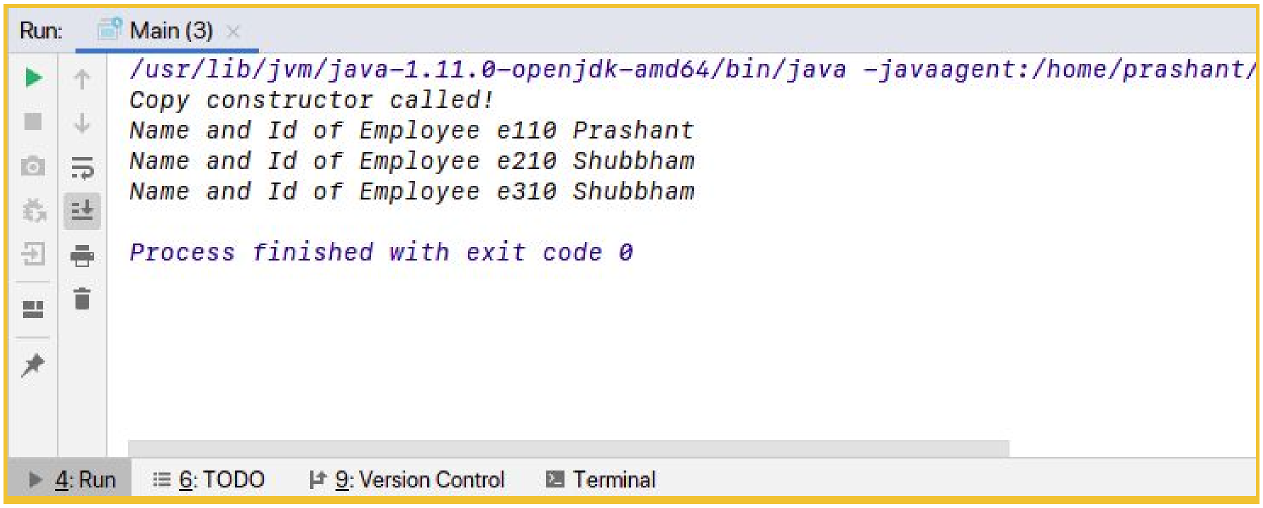
Filename - Main.java



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/D
Cloning Object

75 Prashant
75 Prashant

Process finished with exit code 0
```

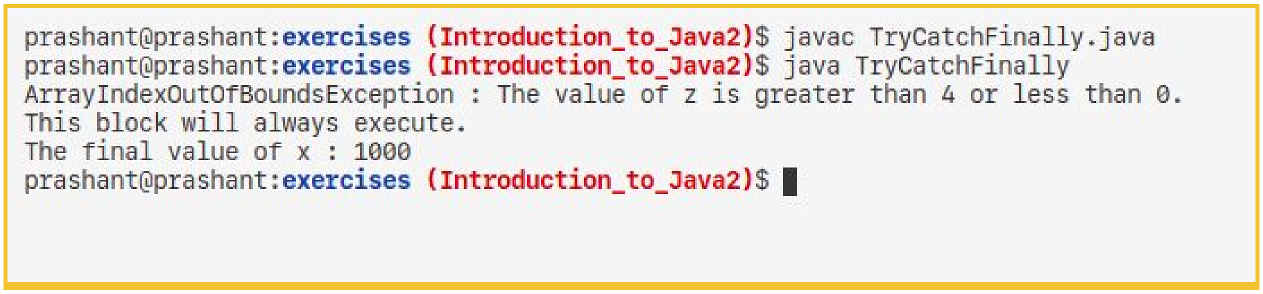


```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/
Copy constructor called!
Name and Id of Employee e110 Prashant
Name and Id of Employee e210 Shubbham
Name and Id of Employee e310 Shubbham

Process finished with exit code 0
```

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

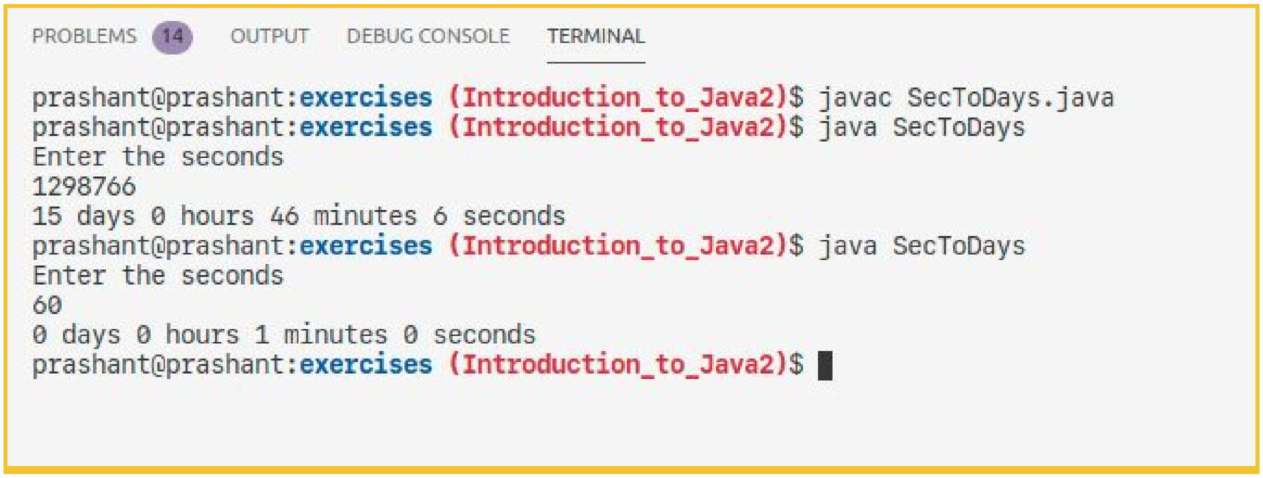
ANS: Filename - TryCatchFinally.java



```
prashant@prashant:exercises (Introduction_to_Java2)$ javac TryCatchFinally.java
prashant@prashant:exercises (Introduction_to_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
prashant@prashant:exercises (Introduction_to_Java2)$ 
```

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

ANS: Filename - SecToDays.java



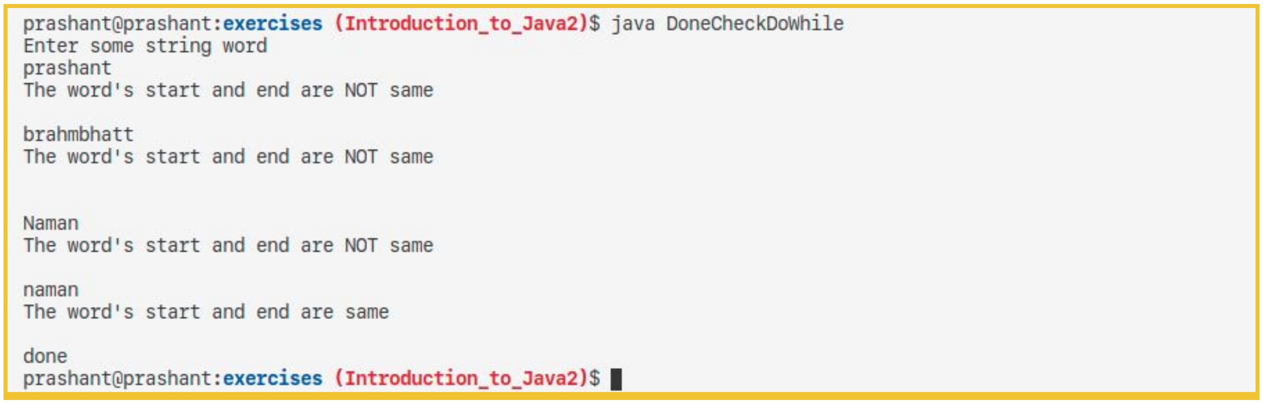
```
prashant@prashant:exercises (Introduction_to_Java2)$ javac SecToDays.java
prashant@prashant:exercises (Introduction_to_Java2)$ java SecToDays
Enter the seconds
1298766
15 days 0 hours 46 minutes 6 seconds
prashant@prashant:exercises (Introduction_to_Java2)$ java SecToDays
Enter the seconds
60
0 days 0 hours 1 minutes 0 seconds
prashant@prashant:exercises (Introduction_to_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

ANS: Filename - DoneCheckDoWhile.java , DoneCheckWhile.java



```
prashant@prashant:exercises (Introduction_to_Java2)$ java DoneCheckDoWhile
Enter some string word
prashant
The word's start and end are NOT same

brahmbhatt
The word's start and end are NOT same


Naman
The word's start and end are NOT same

naman
The word's start and end are same

done
prashant@prashant:exercises (Introduction_to_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.

ANS: Filename - ChairClass.java, Chair.java

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

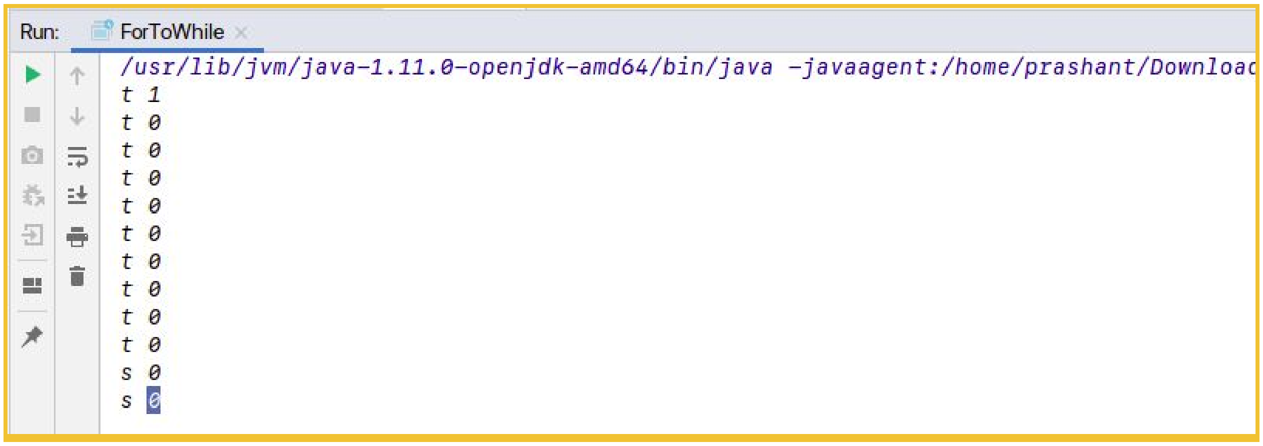
- Places a notification that order for token is ready

ANS: 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);
```

ANS: Filename -ForToWhile.java



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/Downloac
t 1
t 0
t 0
t 0
t 0
t 0
t 0
t 0
t 0
t 0
s 0
s 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");
    }
}
```

ANS: static - grandparent

static - parent

static - child

instance - grandparent

constructor - grandparent

instance - parent

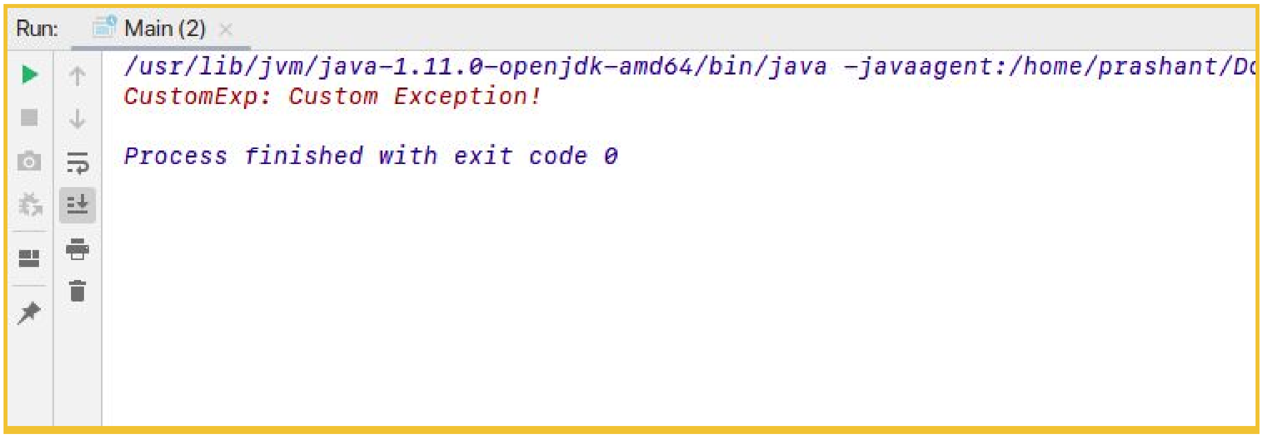
constructor - parent

instance - child

constructor - child

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

ANS: Filename -CustomExp.java



```
/usr/lib/jvm/java-1.11.0-openjdk-amd64/bin/java -javaagent:/home/prashant/Do
CustomExp: Custom Exception!

Process finished with exit code 0
```