LAB EXERCISE ON FILE HANDLING

NAME : SHERAL SIMON WASKAR

REG. NO : 20BCE1182

COURSE CODE : CSE 1007 LAB

COURSE : JAVA PROGRAMMING LAB

SLOT : L13-L14

FACULTY : J V THOMAS ABRAHAM

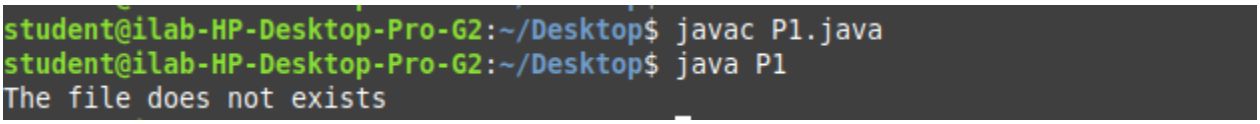
DATE : 06/04/2022

1. Write a program to create a text file in the path c:\java\abc.txt and check whether that file is exists or not.
Code :

```
import java.io.*;

public class P1
{

public static void main(String[] args)
{
```

```
File f = new File("F\\abc.txt");

if(f.exists())

System.out.println("The file exists ");

else
System.out.println("The file does not exists ");

}
}
```

Output :



2) Write a program to accept specified number of characters as input, convert them into uppercase characters and save the data in a text file

Code:

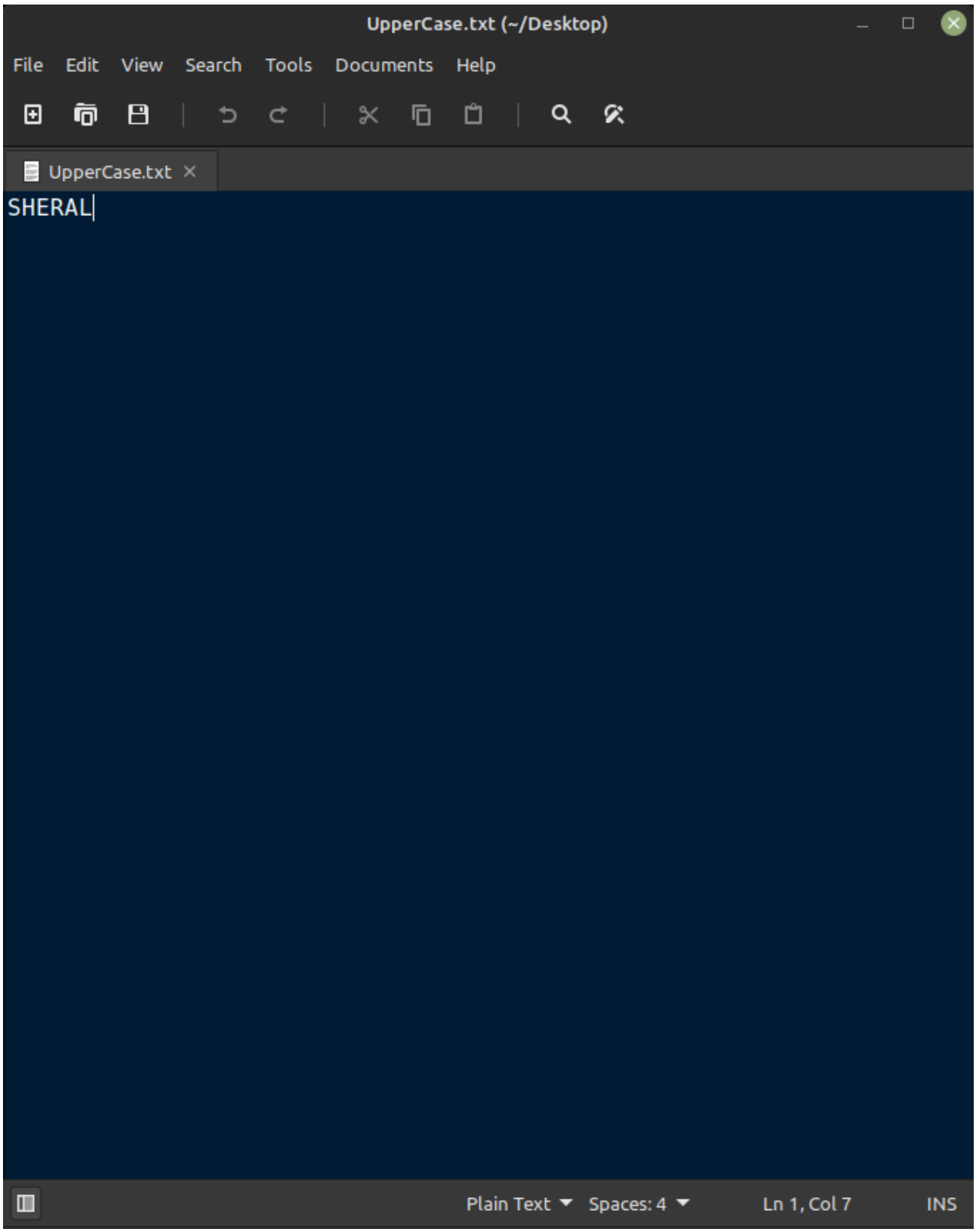
```
import java.util.*;
import java.io.FileWriter;
import java.io.IOException;
```

```
public class P2
{

public static void main(String[] args)
{

Scanner sc = new Scanner(System.in);

System.out.println("Enter a string " );

String str = sc.next();

String str1 = str.toUpperCase();
try
{



FileWriter f = new FileWriter("UpperCase.txt");
f.write(str1);
System.out.println(str1);
f.close();

System.out.println("The a file named UpperCase is created
and the content is saved in the file ");
}

catch(IOException e)
{
System.out.println(e.getMessage());
```

```
}

}

}
```

Output :

```
student@ilab-HP-Desktop-Pro-G2:~/Desktop$ javac P2.java
student@ilab-HP-Desktop-Pro-G2:~/Desktop$ java P2
Enter a string
sheral
SHERAL
The a file named UpperCase is created and the content is saved in the file
student@ilab-HP-Desktop-Pro-G2:~/Desktop$
```



3) Write a program for a mail-order company. The program uses a data-entry screen in which the user types an item number and a quantity. The valid item numbers and prices are as follows:

Item-Number Price

101 100.50
105 175.95
108 220.45
115 280.75
125 300.99
When the user enters an item number, check the number to make sure that it is valid. If it is valid, write the record that includes item number, quantity, price in a text file, otherwise your program should raise a user-defined exception.

Code :

```
import java.io.*;
import java.util.*;
class InvalidItemNumberException extends Exception{

}

public class File {
public static void main(String[] args) throws
IOException,InvalidItemNumberException {
FileOutputStream os = new FileOutputStream("file2.txt");
DataOutputStream dos = new DataOutputStream(os);
int itemNo,quantity;
int Item[]=new int[5];
Item[0]=101;
Item[1]=105;
Item[2]=108;
Item[3]=115;
Item[4]=125;
```

```
double price[]=new double[5];
price[0]=100.50;
price[1]=175.95;
price[2]=220.45;
price[3]=280.75;
price[4]=300.99;
String ch;
Scanner sc = new Scanner(System.in);

int i,c=0;
do
{
c=0;
System.out.println("Enter the item number,quantity:");

itemNo = sc.nextInt();

quantity = sc.nextInt();
try{
for(i=0;i<5;i++){

if(itemNo==Item[i]){

dos.writeInt(itemNo);

dos.writeInt(quantity);

dos.writeDouble(price[i]);
c=1;
}
}
```

```
if(c==0){
throw new InvalidItemNumberException();
}
}

catch(InvalidItemNumberException ie){
System.out.println("Invalid Item Number");


}
System.out.println("Continue(y/n)?");
ch = sc.next();
}while(ch.equals("y"));
dos.close();

}
}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac File.java

D:\SEM 4\CSE1007_LAB>java -classpath .;yourjar.jar File
Enter the item number,quantity:
101
4
Continue(y/n)?
y
Enter the item number,quantity:
105
5
Continue(y/n)?
y
Enter the item number,quantity:
108
4
Continue(y/n)?
n
```

4) Modify the above program such that, if the item number is valid, save the record in a file in object format, otherwise your program should raise a user-defined exception.
Code :

```
import java.util.Scanner;
import java.io.FileWriter;
import java.io.Serializable;

class MyException extends Exception
{
public MyException(int ino)
{
super(ino);
}

}


public class SaveData implements Serializable
{
 private int ino;
 private double price;
//int quantity = sc.nextInt();

SaveData(int ino , double price , int quantity)
{

this.ino = ino;
this.price = price;
this.quantity = quantity;
}
```

```
public void valid(int ino) throws MyException
{
if(ino != 101 || ino != 105 || ino != 108 || ino != 115 || ino != 125)
throw new Valid("The item number is not valid ");


else
{


FileWriter fw = new FileWriter("Details.txt");

if(ino == 101)
{

quantity  = sc.nextInt();

price = 100.50;
fw.write(ino , quantity , price);
}

else if(ino == 105)
{

quantity = sc.nextInt();
price = 175.95;
fw.write(ino , quantity , price);
}
```

```
else if(ino == 108)
{
quantity = sc.nextInt();
price = 220.45;
fw.write(ino , quantity , price);
}

else if(ino == 115)
{
quantity = sc.nextInt();
price = 280.75;
fw.write(ino , quantity , price);
}
else if(ino == 125)
{
quantity = sc.nextInt();
price = 300.99;
fw.write(ino , quantity , price);
}


fw.close();
}
}



public static void main(String[] args)
{
```

```
public double price;
public  int quantity;
Scanner sc = new Scanner(System.in);

try
{
int ino = sc.nextInt();
valid(ino);
SaveData sd1 = new SaveData(ino , price , quantity);
SaveData sd2 = new SaveData(ino , price , quantity);

FileOutputStream f = new FileOutputStream(new
File("SavedData.txt"));
ObjectOutputStream o = new ObjectOutputStream(f);

o.writeObject(sd1);
o.writeObject(sd2);

o.close();
f.close();

}



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

```
}
}
```

5) Write a program that reads the data file created by the Q3 program and displays one record at a time on screen and finally prints the total price.

Code :

```
import java.io.*;

public class Price{

  public static void main(String args[]) throws Exception {

  File file = new File("Price.txt");

  InputStreamReader streamReader =
  new InputStreamReader(new FileInputStream(file));

  BufferedReader br = new BufferedReader(streamReader);

  String line = new String();

  System.out.println(file.getName());
  System.out.println("");
  while (br.ready()) {
    line = br.readLine();

    System.out.println(line);
  }
```

```
  System.out.println("\n");
  System.out.println("Sum is : ");

  System.out.println(line);
  }
}
```

Output :

```
D:\SEM 4\CSE1007_LAB>javac Price.java

D:\SEM 4\CSE1007_LAB>java Price.java
Price.txt
================


100.50
175.95
220.45
280.75
300.99

================
Sum is :
300.99
```

6) Consider a table “Employee” with fields: EmpID, EmpName, department and designation. Write a Java program to perform the following operations:

a) Read employee name, department and designation and insert a new record. EmpID should be generated automatically such that new employee ID is one greater than maximum employee ID.

b) Given an employee ID, your program should display other details.

c) Given a department name, your program should print all employees belong to that department.

d) Given a employee ID, your program should be able to change either her department or designation

Code :

```
import java.util.*;
import java.io.*;

class Empdata
{
Scanner sc = new Scanner(System.in);

public void read()
{
FileReader fr = new FileReader("ReadData");
int i;
while((i = fr.read()) != -1)
System.out.println((char)i);

FileWriter fw = new FileWriter("ReadData");
fw.write("EmployeeName : Reena \n Department :
Marketing \n Designation : Manager ");

        empid = empid + i;
        fw.write(empid);
}
fw.close();
}


int ch;
do
{
```

```
System.out.println("Enter choice 1:read , 2:display ,
3:modifydept , 4:modifydesig);
ch = sc.nextInt();

switch(ch)
{

case 1:
read();
break;

case 2:
display();
break;

case 3:
modifydept();
break;

case 4:
modifydesig();
break;

default:
System.out.println("Invalid choice ");
break;

}
```