Madan Bhandari Memorial College

Department of Computer Science and Information Technology (B.Sc.CSIT)
Ninayak Nagar, New Baneshwor, Kathmandu

Practical Sheet

Submitted By:- Utsav Acharya Sharma | Program No:- 14

Submitted To Debesh Adhikari | Lab Date:- 2080/09/22

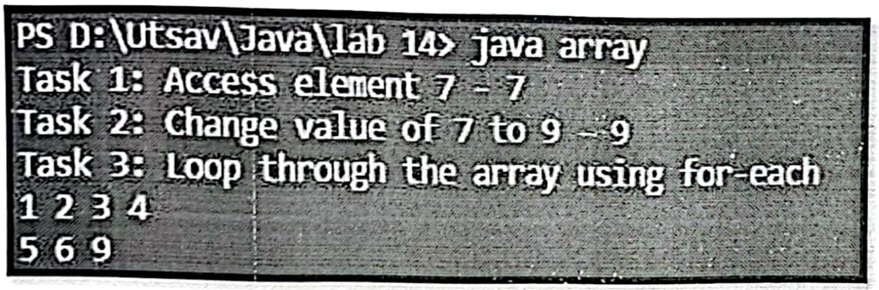
Submission Date:- 2080/09/24 | T.U.Roll.No. :- 24179

Title: Basic Java Programs

Array

Code:

```
public class array{
   public static void main(String[] args){
        int[][] myNumbers = { {1,2,3,4}, {5,6,7}};
        int element7 = myNumbers[1][2];
        system.out.println("Task1: Access element 7 -"+element7);
        myNumbers[1][2]=9;
        System.out.println("Task 2: Change value of 7 to9 -"+myNumbers
                                                              [1][2]);
        System.out.println("Task 3: Loop through the array using for-
                                   each");
        for(int[] row: myNumbers){
              for(int value : row){
                     System.out.print(value+" ");
              }
              System.out.println();
        }
   }
}
```

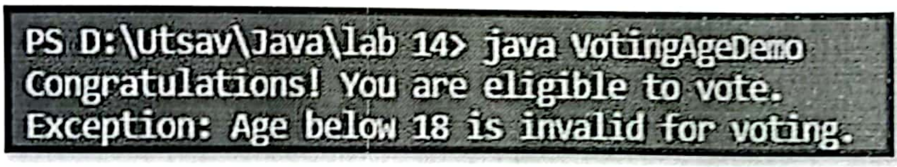


```
PS D:\Utsav\Java\lab 14> java array
Task 1: Access element 7 - 7
Task 2: Change value of 7 to 9 - 9
Task 3: Loop through the array using for-each
1 2 3 4
5 6 9
```

Exception Handeling:

Code:

```
class InvalidAge Exception extends Exception{
    public InvalidAge Exception (String message){
        super(message);
    }
}
public class VotingAgeDemo{
    public static void checkVotingAge(int age) throws InvalidAgeException{
        if(age<18){
            throw new InvalidAgeException("Age below 18 is invalid for voting.");
        }else{
            system.out.println("Congratulations! You are eligible to vote.");
        }
    }
    public static void main(String [] args){
        try{
            checkVotingAge(21);
        }catch (InvalidAgeException e){
            system.out.println("Exception: "+e.getMessage());
        }
        try{
            checkVotingAge(16);
        }catch (InvalidAgeException e){
            System.out.println("Exception: "+e.getMessage());
        }
    }
}
```



```
PS D:\Utsav\Java\lab 14> java VotingAgeDemo
Congratulations! You are eligible to vote.
Exception: Age below 18 is invalid for voting.
```

✓

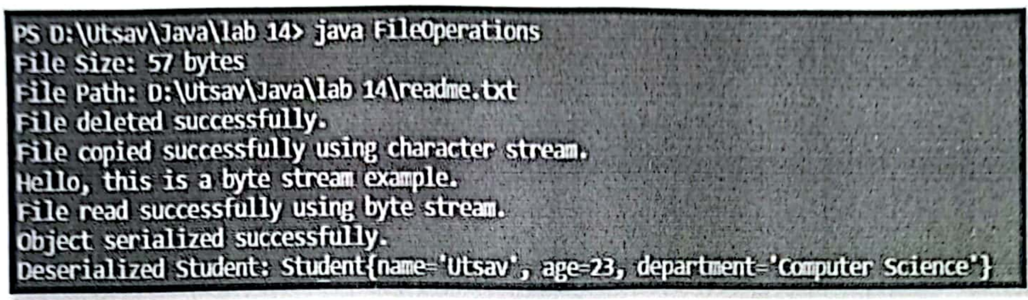
File Handeling:

Code:

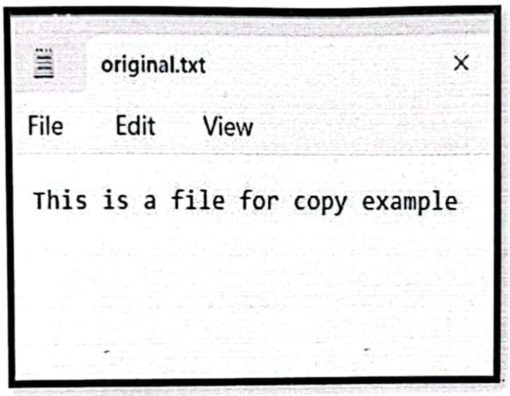
```
import java.io.*;
public class FileOperations {
    public static void main (String[] args){
        try {
            String readmeFilePath = "D:/Utsav/Java/lab14/readme.txt";
            writeToFile(readmeFilePath, "This is a readme file. You
                        have all the information here.");
          readAndDisplayFileInfo (readmeFilePath);
          deleteFile (readmeFilePath);
          String originalFilePath ="original.txt";
          String copyFilePath = "copy.txt";
           writeToFile (originalFilePath, "This is a file for copy
                        example");
           copyFileUsingCharacterStream(originalFilePath, copy
                                        FilePath);
            String byteFilePath = "byteExample.txt";
            writeToFileUsingByteStream(byteFilePath, "Hello,
               this is a byte stream example.");
             readAndDisplayFileContent(byteFilePath);
          // ~~String~~ serialization and deserialization
            String serializedFilePath = "serializedStudent.dat";
            Student student = new Student ("Utsav", 23, "Comp-
                                           uter Science");
            serializeObject (student, serializedFilePath);
            student deserializedStudent =(Student) deserialize
                                    Object(serializedFilePath);
            System.out.println ("Deserialized Student: " +deseria-
                              lizedStudent);
         } catch (IO Exception| classNotFound Exception e){
             e.printStackTrace();
         }
private static void writeToFile (String filePath, String content)
        throws IOException{
        try (BufferedWriter writer = new BufferedWriter (new
                                        FileWriter (filePath))){

                writer.write(content);
        }
}
private static void readAndDisplayFileInfo (String filePath)
{
        File file = new File(filePath);
        System.out.println(" File Size: "+ file.length() +"bytes");
        system.out.println ("File Path: " +file.getAbsalutePath());
}
private static void deleteFile (String filePath){
        File file = new File(filePath);
        if(file.delete()){
                System.out.println("File deleted successfully.");
        }else{
                System.out.println("~~Ed~~Failed to delete the file.");
        }
}
private static void copyFileUsingCharacterStream (String
                sourcePath, String destinationPath) throws IOException
        try (BufferedReader reader = new BufferReader(new
                                        FileReader(source Path));
                BufferedWriter writer = new Buffered Writer(new
                                        FileWriter(destination Path)))
                {
                String line;
                while ((line = reader.readLine()) != null){
                        writer.write(line);
                        writer.newLine();
                }
                System.out.println("File copied successfully using
                                character stream.");
}  }
```

```
private static void writeToFileUsingByteStream(String FilePath,
                                     String content) throws IOException{
    try(FileOutputStream fos= new FileOutputStream(FilePath)){
          byte[] bytes= content.getBytes();
           fos.write(bytes);
    }
}
private static void readAndDisplayFileContent(String FilePath)
                    throws IOException{
    try(FileInputStream fis = new FileInputStream(FilePath)){
          int content;
          while ((content= fis.read()) != -1){
                system.out.print((char) content);
           }
           System.out.println("\nFile read successfully
                                    using byte Stream.");
    }
}
private static void serializeObject(Object object, String
              filePath) throws IOException{
    try (ObjectOutputStream oos = new ObjectOutputStream
                                   (new FileOutputStream(FilePath))){
            oos.writeObject(object);
    }
    System.out.println("Object serialized successfully.");
}
private static Object deserializeObject(String FilePath) throws
           IOException, ClassNotFoundException{
    try (ObjectInputStream ois= new ObjectInputStream
                   (new FileInputStream(FilePath))){
            return ois.readObject();
    }
}
```

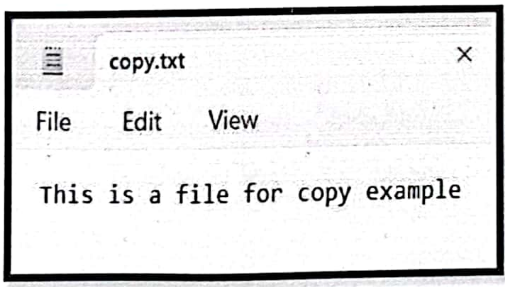
```
static class Student implements Serializable{
    private String name;
    private int age;
     private String department;
     public Student (String name, int age, String department){
            this.name =name;
             this. age = age;
              this.department =department;
       }
       @Override
        public String toString(){
             return "Student {'+
                        "name='"+name+'\''+
                         ", age="+age+
                         ", department='"+department+'\''+
                         '}';
           }
      }
}
```



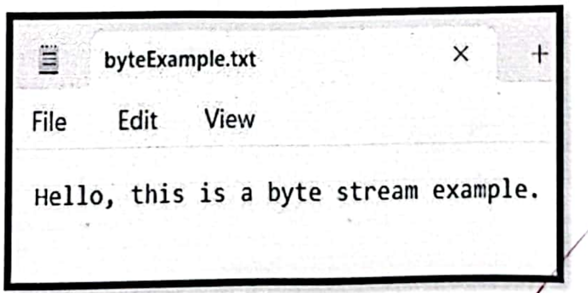
```
PS D:\Utsav\Java\lab 14> java FileOperations
File Size: 57 bytes
File Path: D:\Utsav\Java\lab 14\readme.txt
File deleted successfully.
File copied successfully using character stream.
Hello, this is a byte stream example.
File read successfully using byte stream.
Object serialized successfully.
Deserialized Student: Student{name='Utsav', age=23, department='Computer Science'}
```



```
This is a file for copy example
```



```
This is a file for copy example
```



```
Hello, this is a byte stream example.
```