Day 32

Q. Write a program to read the data from the file and copy it on another file.

Answer:

The FileInputStream is used for reading streams of raw bytes and FileOutputStream is used to write the data on file. The write () method is used to write the data on a text file.

In this example, we read the data from a file and then copy it on another file. First specify the name of the source file and destination file.

```
import java.io.*;
class CopyFile
{
    public static void main(String[] args) throws IOException
    {
        int i;
        FileInputStream fin;
        FileOutputStream fout;
        try
        {
            fin = new FileInputStream(args[0]);
        }
        catch (FileNotFoundException fnfe)
        {
            fnfe.printStackTrace();
            return;
        }
        try
        {
            fout = new FileOutputStream (args[1]);
        }
        catch (FileNotFoundException fofe)
        {
            fofe.printStackTrace();
            return;
        }
        catch (ArrayIndexOutOfBoundsException ai)
        {
            ai.printStackTrace();
            return;
        }
        try
        {
            do
            {
                i = fin.read ();
                if (i != -1)
                {
                    fout.write(i);
                }
```

```
            }
            while (i != -1);
        }
        catch (IOException ioe)
        {
            ioe.printStackTrace();
        }
        fin.close();
        fout.close();
    }
}
```

Q. Write a program to read the data from the file using FileInputStream.

Answer:

FileInputStream class is used to read the data from the file. it should be used to read byte-oriented data for example read image, audio, video.

In this example, we read the data from a text file "test.txt" and print the ASCII value of each character. The printStackTrace() method is used to print error message.

//test.txt
Welcome

```
import java.io.*;
public class FileInputStreamTest
{
    public static void main(String args[])
    {
        try
        {
            FileInputStream fis = new FileInputStream("test.txt");
            int i;
            System.out.println("ASCII value of the character:");
            while((i=fis.read()) != -1)
            {
                System.out.print(i+" : ");
                System.out.println((char)i);
            }
        }
        catch (FileNotFoundException fnfe)
        {
            fnfe.printStackTrace();
        }
        catch (IOException io)
        {
            io.printStackTrace();
        }
    }
```

```
}
```

Q. Write a program for counting number of characters, words and lines in a file.

Answer:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

public class WordCharLineCountInFile
{
   public static void main(String[] args)
   {
      BufferedReader reader = null;
      int charCount = 0;
      int wordCount = 0;
      int lineCount = 0;

      try
      {
         reader = new BufferedReader(new FileReader("test.txt"));
         String currentLine = reader.readLine();
         while (currentLine != null)
         {
            lineCount++;
            String[] words = currentLine.split(" ");
            wordCount = wordCount + words.length;

            for (String word : words)
            {
               charCount = charCount + word.length();
            }
             currentLine = reader.readLine();
         }
          System.out.println("Number of character in file : "+charCount);
          System.out.println("Number of words in a file : "+wordCount);
          System.out.println("Number of lines in file : "+lineCount);
      }
      catch (IOException e)
      {
         e.printStackTrace();
      }
      finally
      {
         try
         {
            reader.close();
         }
         catch (IOException e)
         {
            e.printStackTrace();
```

```
        }
      }
   }
}
```

Q. Write a program to read the file using BufferedReader class.

Answer:

BufferedReader class uses two methods for reading the file. One is readLine() method and another is read() method.

In this example of BufferedReader class, we are using both methods to read the file. The name of the text file is "test.txt".

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;

class BufferedReaderDemo
{
    public static void main(String[] args)
    {
        BufferedReader br1 = null;
        BufferedReader br2 = null;
        try
        {
            br1 = new BufferedReader (new FileReader("test.txt"));
            System.out.println("Using readLine () method");
            String content = br1.readLine();
            while (content != null)
            {
                System.out.println(content);
                content = br1.readLine();
            }
            br2 = new BufferedReader (new FileReader("test.txt"));
            System.out.println("Using read() method");
            int i = 0;
            char ch;
            while ((i = br2.read()) != -1)
            {
                ch = (char)i;
                System.out.print(ch);
            }
        }
        catch (IOException ioe)
        {
            ioe.printStackTrace();
        }
    }
```

```
}
```

Q. Implementing FileReader and FileWriter class to read and write the data from file.

Answer:

The FileReader class creates a Reader that can be used to read the content of the file. The FileWriter class creates a Writer class that can be used to write the content of the file.

Here "test.txt" is a text file. We read the data from the text file by using FileReader class. The BufferdReader class is used to read the text from the input stream.

```
import java.io.*;
class FileReaderFileWriter
{
    public static void main(String[] args)
    {
        String file = "test.txt"; //name of the text file
        String text = null;
        System.out.println("Content of the file: ");
        try
        {
            FileReader fileReader = new FileReader(file);
            BufferedReader bufferedReader = new BufferedReader(fileReader);
            while ((text = bufferedReader.readLine()) != null)
            {
                System.out.println(text);
            }
            bufferedReader.close();
        }
        catch (FileNotFoundException ex)
        {
            ex.printStackTrace();
        }
        catch (IOException ex)
        {
            ex.printStackTrace();
        }
    }
}
```

Q. Write a program to read the username and password from the user and display them using Console class.

Answer:

Console Class:
Console class provides the method to access the character based console device. If any application needs to read the password or other secure data, then we use the console class.

Here, we read the username and password from the user and print them on console. The readPassword () method used to read the password from the user and it doesn’t show at time of entering.

```
import java.io.Console;
public class ConsoleTest
{
    public static void main(String[] args) throws Exception
    {
        Console console = System.console();
        if (console == null)
        {
            System.out.println("Unable to fetch console");
            return;
        }
        System.out.print("Enter username:");
        String user = console.readLine();

        System.out.print("Enter password:");
        char ch[] = console.readPassword();
        String password = String.valueOf(ch);

        System.out.println("Username: "+user);
        System.out.println("Password:"+password);
    }
}
```

Q. Write a program to delete an existing file.

Answer:

The java.io.File.delete() method is used to delete the file. The BufferedReader class reads the data from the file.

In this example, we use delete() method, which deletes the file permanently.

```
import java.io.*;
class DeleteFile
{
    public static void main(String[] args)
    {
        try
        {
            BufferedWriter bufferedWriter = new BufferedWriter(new FileWriter("file.txt"));
            bufferedWriter.write("Welcome");
            bufferedWriter.close();
            boolean status = (new File("file.txt")).delete();
            if (status)
            {
                System.out.println("File deleted successfully");
```

```
            }
            BufferedReader bufferedReader = new BufferedReader(new FileReader("file.txt"));
            String str;
            while ((str = bufferedReader.readLine()) != null)
            {
                System.out.println(str);
            }
            bufferedReader.close();
        }
        catch (IOException ioe)
        {
            System.out.println("File not found");
            ioe.printStackTrace();
        }
    }
}
```

Q. Create a class date with day, month and year as members. Write appropriate member functions. Create another class students, which has id, name, date of birth and marks of 3 subjects as members. Write appropriate constructor for the student which assigns values to the members. Accept the details as command line arguments and create a student object using the arguments. Display the student details in a proper format.

Answer:

Below example has two classes, Date and Students class. We insert the value through command line argument.

```
class Date
{
    int day, month, year;
    Date(int day, int month, int year)
    {
        this.day = day;
        this.month = month;
        this.year = year;
    }
    Date(){}
}
```

```
import java.io.*;
import java.lang.*;
class Students extends Date
{
    int id;
    String name;
    Date d1;
    int marks[] = new int[3];
    Students(int id, String name, Date d, int s1, int s2, int s3)
    {
        this.id = id;
```

```
        this.name = name;
        marks[0] = s1;
        marks[1] = s2;
        marks[2] = s3;
        d1 = new Date(d.day, d.month, d.year);
    }
    public void display()
    {
        System.out.println("\n\nID   Name\tDOB\t  Marks of 3 Subjects");
        System.out.println("===  =====\t=======\t  ==================");
        System.out.println(+id+"  "+name+" \t"+d1.day+"/"+d1.month+"/"+d1.year+"
"+marks[0]+"  "+marks[1]+" "+marks[2]);
        System.out.println("===  =====\t=======\t  ==================");
    }
    public static void main(String ar[])
    {
        Date d = new Date(Integer.parseInt(ar[2]),Integer.parseInt(ar[3]),Integer.parseInt(ar[4]));
        Students  s1 = new
Students(Integer.parseInt(ar[0]),ar[1],d,Integer.parseInt(ar[5]),Integer.parseInt(ar[6]),Integer.parseI
nt(ar[7]));
        s1.display();
    }
}
```