JBK1010-Input Output Stream (File I/O)

Create a Directory in Java

```
import java.io.File;
class CreateDirectory {
public static void main(String args[]) {
  try {
        String one ="e:/jbk";              // one directory
                  String many="e:/hello/hi/say";// multiple directories

          // Create one directory
          File file = new File(one);
          if(!file.exists()) {
                  if(file.mkdir()) {
         System.out.println("Directory : " + one + " created");
                  }  }
          // Create multiple directories
          File files = new File(many);
          if(!files.exists()) {
                  if(files.mkdirs()) {
        System.out.println("Directories : " + many + " created");
                  } }  }
  catch (Exception e) {
          System.err.println("Error : " + e.getMessage());
  } } }
```

Create a File in Java

```
1) import java.io.File;
import java.io.FileOutputStream;
import java.io.FileNotFoundException;
public class CreateFile {
        public static void main(String[] args) {
        String fileName = " e:/jbk /success.txt";
        File file = new File(fileName);
        // If the file doesn't exist
        if (!file.isFile()) {
                // Check the parent directory...
                file = file.getAbsoluteFile();
                File parentDir = new File(file.getParent());
                if (!parentDir.exists()) {   // ... and create it if necessary
                        parentDir.mkdirs();
        System.out.println(parentDir + " parent dir is created..");
                } }
        // Place to store the stream reference
        FileOutputStream outputFile = null;
        try {
            // Create the stream opened to append data
            outputFile = new FileOutputStream(file, true);
            System.out.println(file + " file is created..");
        } catch (FileNotFoundException e) {
            e.printStackTrace(System.err);
        } System.exit(0); }  }
    2) import java.io.File;
    import java.io.IOException;
    public class CreateFileDemo{
     public static void main( String[] args ) {
      try {
      File file = new File("d:\\newfile.txt");
      /*If file gets created then the createNewFile()
       * method would return true or if the file is
       * already present it would return false
       */
     boolean fvar = file.createNewFile();
      if (fvar){
        System.out.println("File has been created successfully");
      } else {
        System.out.println("File already present at the specified
location");
      }  } catch (IOException e) {
        System.out.println("Exception Occurred:");
       e.printStackTrace();
     }  } }
```

Read from a file in Java

```
  1) import java.io.BufferedReader;
  import java.io.File;
    import java.io.FileReader;
    import java.io.IOException;
    public class Read {
    public static void main(String[] args) {
        BufferedReader reader = null;
        try {
            String CurrentLine;
            File file = new
File("/home/balaji/samplecodez/success.txt");
            FileReader fileReader = new FileReader(file);
            reader = new BufferedReader(fileReader);
            while ((CurrentLine = reader.readLine()) != null) {
                System.out.println(CurrentLine);
            } } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (reader != null)reader.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }}}}
```

3) Read file in Java using BufferedReader

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class ReadFileDemo {
  public static void main(String[] args) {
BufferedReader br = null;
BufferedReader br2 = null;
try{
   br = new BufferedReader(new FileReader("d:\\myfile.txt"));
   //One way of reading the file
     System.out.println("Reading the file using readLine() method:");
     String contentLine = br.readLine();
     while (contentLine != null) {
       System.out.println(contentLine);
       contentLine = br.readLine();
     }
     br2 = new BufferedReader(new FileReader("d:\\myfile2.txt"));
     //Second way of reading the file
     System.out.println("Reading the file using read() method:");
     int num=0;
     char ch;
     while((num=br2.read()) != -1){
      ch=(char)num;
        System.out.print(ch);
     } }
catch (IOException ioe) {
     ioe.printStackTrace();
} finally {
     try {
       if (br != null)
            br.close();
       if (br2 != null)
            br2.close();
     }
     catch (IOException ioe)   {
           System.out.println("Error in closing the BufferedReader");
     }}}}
```

Write into a file in Java

1) import java.io.BufferedReader;

```
    import java.io.BufferedWriter;
        import java.io.File;
        import java.io.FileWriter;
        import java.io.IOException;
        import java.io.InputStreamReader;
        public class Write {
        public static void main(String[] args) {
                try {
                        String content = null;
                        String OldContent = "";
                        File file = new
File("/home/balaji/samplecodez/success.txt");
                        if (!file.exists()) {
                                file.createNewFile();
                        }
                        FileWriter fw = new FileWriter(file.getAbsoluteFile());
                        BufferedWriter bw = new BufferedWriter(fw);
                        BufferedReader reader = new BufferedReader
                                        (new InputStreamReader(System.in));
                System.out.println("'end' to quit and save your content..\n");
                        do {
                                try {
                                        content = (String) reader.readLine();
                                } catch (IOException e) {
                                        // TODO Auto-generated catch block
                                        e.printStackTrace();
                                } OldContent =
OldContent.concat(content).concat("\n");
                        } while(!content.equals("end"));
                        bw.write(OldContent);
                        bw.close();
System.out.println("\nOur content is successfully inserted into\n");
                        System.out.println(file);
                } catch (IOException e) {
                        e.printStackTrace();
                }}}
```

2) write to a file in java using FileOutputStream

```
        import java.io.File;
        import java.io.FileOutputStream;
        import java.io.IOException;
        public class WriteFileDemo {
          public static void main(String[] args) {
         FileOutputStream fos = null;  File file;
          String mycontent = "This is my Data which needs" +
           " to be written into the file";
```

```
    try {
   //Specify the file path here
     file = new File("C:/myfile.txt");
     fos = new FileOutputStream(file);
   /* This logic will check whether the file
      * exists or not. If the file is not found
      * at the specified location it would create
      * a new file*/
     if (!file.exists()) {
       file.createNewFile();
     }
     /*String content cannot be directly written into
      * a file. It needs to be converted into bytes
      */
     byte[] bytesArray = mycontent.getBytes();
     fos.write(bytesArray);
     fos.flush();
     System.out.println("File Written Successfully");
     } catch (IOException ioe) {
     ioe.printStackTrace();
    }   finally {
     try {
       if (fos != null) {
            fos.close();
       } }  catch (IOException ioe) {
       System.out.println("Error in closing the Stream");
     }}}}
```

Update (Read + Write) /append a file in Java

```
    import java.io.BufferedReader;
    import java.io.BufferedWriter;
    import java.io.File;
    import java.io.FileReader;
    import java.io.FileWriter;
    import java.io.IOException;
    import java.io.InputStreamReader;
    public class Update {
    public static void main(String[] args) {
            BufferedReader reader = null;
            String OldContent = "";
            File file = new
File("/home/balaji/samplecodez/success.txt");
            try {
                    String Content;
                    FileReader fileReader = new FileReader(file);
            reader = new BufferedReader(fileReader);
            while ((Content = reader.readLine()) != null) {
    OldContent = OldContent.concat(Content).concat("\n");
            }  } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (reader != null)reader.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }}
        System.out.println("Your Old Content : \n"+OldContent);
    System.out.println("Do you need to Udate your Content ? Y/N \n");
        reader = new BufferedReader(new
    InputStreamReader(System.in));
        char c = 'y';
        try {
            c = (char) reader.read();
        } catch (IOException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }switch(c) {
            case 'y' :
                write(file);
                break;
            case 'n' :
                break;
            case 'Y' :
                write(file);
                break;
            case 'N' :
                break;
            default :
                System.out.println("Press 'Y' or 'N' to Continue..");
        }       }
      private static void write(File file) {
        // TODO Auto-generated method stub
        try {
        String content = null;
            String NewContent = "";
            if (!file.exists()) {
                file.createNewFile();
            } FileWriter fw = new
FileWriter(file.getAbsoluteFile());
            BufferedWriter bw = new BufferedWriter(fw);
            BufferedReader reader = new BufferedReader
                (new InputStreamReader(System.in));
System.out.println("'end' to quit and save your content..\n");
            do {
                try {
                    content = (String) reader.readLine();
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                } NewContent =
NewContent.concat(content).concat("\n");
            } while(!content.equals("end"));
            bw.write(NewContent);
            bw.close();
System.out.println("\nOur content is successfully updated into\n");
            System.out.println(file);
        } catch (IOException e) {
            e.printStackTrace();
        }}}
```

Append content to File using FileWriter and BufferedWriter

```
import java.io.File;
import java.io.FileWriter;
import java.io.BufferedWriter;
import java.io.IOException;
class AppendFileDemo{
  public static void main( String[] args ){
    try{
        String content = "This is my content which would be appended " +
        "at the end of the specified file";
      //Specify the file name and path here
        File file =new File("C://myfile.txt");
        /* This logic is to create the file if the
         * file is not already present
         */
        if(!file.exists()){
          file.createNewFile();       }
        //Here true is to append the content to file
        FileWriter fw = new FileWriter(file,true);
        //BufferedWriter writer give better performance
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write(content);
        //Closing BufferedWriter Stream
        bw.close();
System.out.println("Data successfully appended at the end of file");
    }catch(IOException ioe){
```

```
    System.out.println("Exception occurred:");
      ioe.printStackTrace();
   }}}
```

Append content to File using PrintWriterimport java.io.File;

```
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.BufferedWriter;
import java.io.IOException;
FileWriter fw = new FileWriter(file,true);

          BufferedWriter bw = new BufferedWriter(fw);
          PrintWriter pw = new PrintWriter(bw);
      //This will add a new line to the file content
          pw.println("");
      /* Below three statements would add three
       * mentioned Strings to the file in new lines.
       */
          pw.println("This is first line");
          pw.println("This is the second line");
          pw.println("This is third line"); pw.close();
System.out.println("Data successfully appended at the end of file");
    }catch(IOException ioe){
          System.out.println("Exception occurred:");
          ioe.printStackTrace();
   }}}
```

Program that reads the individual words from text file and prints them out, one per line.

```
import java.io.*;
import java.util.Scanner;
public class ScanXan {
  public static void main(String[] args) throws IOException {
    Scanner s = null;
    try {
       s = new Scanner(new BufferedReader(new
FileReader("xanadu.txt")));
       while (s.hasNext()) {
         System.out.println(s.next());
       } } finally {
       if (s != null) {
         s.close();
       }} }}
```

Note: scanner uses white space to separate tokens. (White space characters include blanks, tabs, and line terminators.