Lecture 23

File Handling

In this lecture we will discuss how to read and write data from permanent storage. So far we know:

- how to declare variables of different types
- how to take input & output in variables
- performing different operation using variables
- how to declare arrays to handle list of any data type [single dimension & multi-dimension]

However, we have to start from scratch every time. We may take input from user but that input will be lost with the termination of program, each time program execution is independent from previous execution. However, in real life we do many things in parts. The task we do in one part is available for next part, rather than to take start from scratch each time. With permanent storage we may achieve following advantages:

- data can be made available for next executions
- data can be shared among different programs
- a program can be made that is useable for long term may be years because data can be stored as transactions proceed day by day
- large volume of stored data can be used as input in programs

There may be other uses of file handling like writing softwares that can handle databases, recording logs, saving cookies (hope you know about cookies in terms of websites and not thinking of eatables). Hope this is enough for understanding why we need file handling and what are the uses of file handling. There are several ways to handle files in Java language. There are two main aspects of file handling. One is writing in files and other is reading files. Statistics shows that reading operations are performed most often and there are many possible ways to read same file. Reading and writing is different as you can easily recall that taking input from user and showing output to user on screen is different. Here we will discuss one method each for reading and writing in files.

File Writing

Before we start discussion on file handling just recall MS Word file every one of you must have used. In MS Word spaces are used to separate words and (.) period is used to separate lines. Enter key is used to separate paragraphs. There are all delimiters used to differentiate between two things. Similarly we will use delimiters while writing data. Usually we use space as delimiter however, we may use \n. For file writing we will use java class "***PrintWriter***". For this we have to import package "***java.io.****". We will use ***print*** or ***println*** function for writing, just like we use to write output on monitor. Finally it is required to use ***close*** function after completion of write for successful write operation. Therefore following code will write 10 random numbers in file "**numbers.txt**". You are able to open this file in textpad, notepad, wordpad to see contents:

```
import java.io.*;

   PrintWriter pw=new PrintWriter("numbers.txt");
   int i,n;
   for (i=1;i<=10;i++){
     n=(int)(Math.random()*50);
     pw.println(n+" ");
   }
   pw.close();
```

We can add path with file name to specify folder otherwise file will automatically be created in current folder where java file exist. Like "***d:/data/numbers.txt***". Note either / will be used or \\ will be used. Single \ will be a logical error you may experience. Similarly using ***print*** or ***println*** function you may write any data type double, String etc. Sometimes we add meta data before actual data we will discuss it in coming lectures. Now we will discuss file reading.

For reading file just like input you should know what to read specifically this means you know what is written in files, we mean data type and size. Like file has integer, double, or String etc. For file reading we will use your familiar class Scanner. Just we have to add another class that is ***File***. We will create object of File and we will pass that object to scanner. Luckily next part is just like you take input from user, means we will write functions ***nextInt***, ***nextDouble***, ***next*** and ***nextLine***. See code:

```
    File file=new File("abc.def");
    Scanner input=new Scanner(file);
    ...
    //next we may take input from file just like we take input from user
```

For your convenience here we are giving complete program to write and read 10 random numbers from file:

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

class FileHandling{
  public static void main(String []args) throws Exception{
    PrintWriter pw=new PrintWriter("d:\\numbers.txt");
    int i,n;
    for (i=1;i<=10;i++){
      n=(int)(Math.random()*50);
      System.out.print(n+" ");
      pw.println(n+" ");
    }
    pw.close();
    System.out.println();

    //Reading from file
    File file=new File("d:\\numbers.txt");
    Scanner input=new Scanner(file);
    while (input.hasNext()){
      n=input.nextInt();
      System.out.print(n+" ");
    }
    input.close();
    System.out.println();
  }
}
```

Note: All functions having file handling code must write phrase "throws Exception" before opening curly brace like:

```
public static String[] words(String fn) throws Exception{
```

This program will generate 10 random numbers between 0-50. Simultaneously it will print those numbers as well as write them into file as well. Later it will read and display those numbers. You must have idea about both reading and writing, though it is not at all necessary to do them together. It is possible that a program may have only read or write operation. Like a program may have write operation only to log activities. Similarly a program may taking input from some file instead of user may have read operation only. Now we will discuss another example to read and write different types of data in same file and this time we will use separate program to read and write. First there is a program FileWritingProgram:

```
import java.io.*;

class FileWritingProgram{
  public static void main(String []args) throws Exception{
    PrintWriter pw=new PrintWriter("students.txt");
    String names[]={"Fareed","Nasir","Zaheer","Javed","Zahoor"};
    int semesterNo[]={2,4,2,4,2};
    double cgpa[]={3.2,2.4,2.9,3.3,3.0};
    int i;
    for (i=0;i<names.length;i++){
      pw.print(names[i]+" "+semesterNo[i]+" "+cgpa[i]+" ");
    }
    pw.close();
  }
}
```

Here we have 3 arrays having data of 5 students. We are writing this data into file "students.txt". You may run this program and open this file into notepad to see data written in it. Now there is a second program:

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

class FileReadingProgram{
  public static void main(String []args) throws Exception{
    File file=new File("students.txt");
    Scanner input=new Scanner(file);
    String name;
    int sNo;
    double cgp;
    while (input.hasNext()){
      name=input.next();
      sNo=input.nextInt();
      cgp=input.nextDouble();
      System.out.println(name+" "+sNo+" "+cgp);
    }
    input.close();
    System.out.println();
  }
}
```

In this program we are reading data from file and because we know what is written in this file. We are reading in the same sequence and printing on screen. That’s all for this lecture hope you understand and learn it be practice.

Practice Task

Write a program to read same program say "FileOperations.java". Read this file word by word using ***next()*** function of Scanner. Open another file for writing say "words.txt". Simultaneously write words in file using ***println()*** function.