Uttara InfoSolutions

www.uttarainfo.com

IO Practicals - Lab 1

Info: This is a very important lab. You will be given the zip of .java files. I have mentioned in brackets which example .java file you need to look into for solving the program. First go through the example code and then build the solution. All the questions are important from both project and interviews point of view. Take your time but complete it please. Take the assistance of the Lab Instructors when needed.

1) Test basic methods of Scanner class usage to read inputs from the keyboard (Ex code in TestScanner.java)

Steps:

Create 2 Scanner objects wrapping System.in like this (import java.util.* as well):

```
Scanner sc1 = new Scanner(System.in);
Scanner sc2 = new Scanner(System.in);

System.out.println("Enter a string input");
String s1 = sc1.next(); // to read a single word input
//print s1
System.out.println("Enter int as input");
int n = sc1.nextInt();
System.out.println("n = "+n);
double d = sc1.nextDouble();
//print d
String line = sc2.nextLine(); // sentence input
//print line
```

Check out other methods usage of Scanner class to read a boolean, float, long and print the contents to the monitor. Remember to use sc2.nextLine() on a separate scanner.

2) Test basic methods of File class usage (Ex code in TestFile.java)

Steps:

a) Ask user to input the path and read it using Scanner. Print it out to check if the input is correct. While inputting a path in a windows machine, give path like this: c:/programs/source or give with 2 \\s like this c:\\programs\\source
b) Create a File object using the path input
c) Check out the working of these methods:

```
File f = new File(path);
```

f.exists(),f.isFile(), f.isDirectory(), f.length(), f.getAbsolutePath(), f.getName() => print out the returned values and understand how each method works. Change the path to a folder and run the program. Now what changes to do you in the output and why?

d) Give path to folder then test File[] fa = f.listFiles(). Loop
over File[] and then print the names of its contents. What does each File ref in the array point to?

3) Write a program to read input path of a text file from the user and print its contents to the monitor. (Ex code in TestPathReading.java). Code it directly in main() for now.

4) Take 2 paths as input from user (first to an existing text file and second to one that does not exist). From one, read the file contents and write it to another. (Ex code in TestIO.java, TestReading.java and ReadFileAndPrint.java)

Steps:

```
try
{
FileReader fr = new FileReader(path1);
FileWriter fw = new FileWriter(path2);

int c;

while((c=fr.read())!=-1)
{
   System.out.print((char)c);
   fw.write(c);
}
fw.close();
fr.close();
}
catch(IOException e)
{
        e.printStackTrace();
}
```

Check if the file is being written. Now code with correct code structure with try..catch..finally as discussed in class. Test it.

Now use BufferedReader and BufferedWriter to improve the performance of this. Read one line at a time and print to the console (See example code first if you have a doubt).

```
BufferedReader br = new BufferedReader(new FileReader(path));
String line;
while((line=br.readLine())!=null)
{
        System.out.println(line);
}
```

Now use FileWriter fw = new FileWriter(path2,true); // check if data is getting appended.

5) WAP to take input a path from the user. If the path is pointing to a folder, print all the file names which are .txt files.

6) Print out which word is occurring the max number of times in a file

Pseudocode:

a)Take a file path as input. validate if it exists and if it is a text file.
b)Create a List<String> words = new ArrayList<String>();
c)Using BufferedReader, read one line at a time, split it by space and
add all the words from String[] to the words list.
d)After all lines content has been added to the words list, loop over
each word and find its number of occurence using Collections.frequency().
e)Print the one which has max num of occ.

7) Take 2 file paths from the user and identify which words are not present in both the files.

8) Sort all the words in a file and print out words a) sorted without duplicates b) sorted with duplicates

9) Have a file (friends.txt) with your friend names (one name per line).Read each line from the file and print it in main(). Then ask the user to edit a name (by asking him what name to edit and then what should be the new name). Read all the names into a collection except the name to edit. Add the new name to the collection. Create a FileWriter on the same file (without append being set) and then overwrite the contents. Then do the same to delete a name from the file by asking the user. This is an important example to understand how to edit/delete contents of a file.

10) WAP to take the input from the user a path and a word. If the path is a text file, search the word in the file and print the number of occurrences. If the path is a folder, read the files in the folder and print the number of occurrences in all the text files of the word (important).

11) Build a Menu and ask the user to add/update/list/delete a phone book of contacts. Look into the demo example and then start this.

Other Collection based problems:

1) Create a Map<String,String>. Test basic working of methods of maps - put(key,val), get(key), remove(key), entrySet(), keySet(), values(), containsKey(), containsValue(). (Look at TestMaps.java)

2) Given a sentence as input, find the number of occurrences of every word in it and print it out (Use Maps).

3) Create a program to take in the country and capitals from the user as inputs. Then a menu should be given to the user to be able to a) search for the capital given a country name b) list the countries and their respective capitals. c) Sort the list output based on country d) *very important* sort the list output based on capitals

4) A Student has a name, date of birth, email and id. Create a menu for the user to allow adding students, get Student info based on id, sort students based on date of birth. Perform date validation correctly for checking date of birth input is a valid date and whether the date is in the past.

5) How sort a Map based on its values?? -> common interview programming question.

6) Create a Stack as an interface with following code
```
public interface Stack
{
public void push(int element);
public int pop();
```

```
public int peek();
public void printElements()
}
```
Create a class called ArrayStack that implements Stack interface like this:
ArrayStack has ints -> 1..n multiplicity which means you can store them in an array instance
variable.
```
public class ArrayStack implements Stack
{
int[] arr = new int[10]; // instance variable where elements will be stored
int count; // to keep a track of how many elements are filled into the stack
public void push(int element)
{
1. check if count is < length of array
2. store element into arr[count].
3. increment count.
}
public int pop()
{
1. check if count >0
2. return arr[count].
3. decrement count
}
public int peek()
{
1. check if count > 0
2. return arr[count] without decrementing count.
}
public void printElements()
{
1. Loop over arr count num of times
2. print each element via SOP
}
}
```

Build a tester class to test ArrayStack something like this:

```
main(..)
{
ArrayStack myStack = new ArrayStack();
myStack.push(5);
myStack.push(10);
myStack.push(15);
myStack.printElements(); -> should print 15 10 5
SOP(myStack.peek()); -> should print 15
SOP(myStack.pop()); -> should print 15
SOP(myStack.peek()); -> should print 10
myStack.printElements(); -> should print 10 5
myStack.push(20);
myStack.printElements(); -> should print 20 10 5
// test all boundary conditions!
}
```

7) Implement a Stack using LinkedList as its data structure. Use the Stack to implement reversing of words in a given sentence.