JAVA Programming Language Homework VI: I/O

ID: **Name:**

1. When comparing java.io.BufferedWriter to java.io.FileWriter, which capability exists as a method in only one of the two?

A. Closing the stream

B. Flushing the stream

C. Writing to the stream

D. Marking a location in the stream

E. Writing a line separator to the stream

ANS: E

- A：兩者皆有關閉串流的方法
- B：兩者皆有清除串流的方法
- C：兩者皆有寫出串流的方法
- D：兩者皆沒有標記串流位置的方法，通常是用在讀入串流使用
- E：BufferedWriter 有寫出換列 newLine() 方法；FileWriter 則無

2 Chain these constructors to create objects to read from a file named "in" and to write to a file named "out".

1.	Reader = [1. place here] [2. place here] "in"));
2.	Writer = [3. place here] [4. place here] [5. place here] "out"));

Constructors:

A. new FileReader (	B. new PrintReader (	C. new BufferedReader (
D. new BufferedWriter (	E. new FileWriter (	F. new PrintWriter (

Which sequence is correct?

A. CAFDE

B. ACDFE

C. CAEDF

D. CBFDE

E. BCDFE

ANS: A

Reader = new BufferedReader (new FileReader ("in"));
Writer = new PrintWriter (new BufferedWriter (new FileWriter ("out"));
備註：並沒有 PrintReader 的類別，相關資訊可洽：
http://java.sun.com/j2se/1.4.2/docs/api/java/io/package-summary.html

3 Place the code fragments into position to use a BufferedReader to read in an entire text file.

```
1.        class PrintFile {
2.          public static void main (String[] args) {
3.            BufferedReader buffReader = null;
4.            // more code here to initialize buffReader
5.            try {
6.              String temp;
7.              while( [1. place here] [2. place here] ) {
8.                System.out.println(temp);
9.              }
10.           } catch [3. place here]
11.             e.printStackTrace();
12.           }
13.         }
14.       }
```

Code Fragments:

A. (temp = buffReader.readLine ())	B. && buffReader.hasNext ()
C. (temp = buffReader.nextLine ())	D. (IOException e) {
E. ! = null	F. (FileNotFoundException e) {

Which sequence is correct?

A. AED

B. AEF

C. ABD

D. CBF

E. CED

ANS: A

```
while( (temp = buffReader.readLine () ) != null ) {
   System.out.println(temp);
}
} catch (IOException e) {
```

- temp 很明顯的是用來讀取資料的變數;之前只有 BufferedReader,所以 是利用 BufferedReader 的 method 來讀取,其中可選的答案只有 readLine(),因為並沒有 nextLine() 這個 method 存在。
- 放在 while 中,需要的是一個邏輯運算式,例如表示讀不到資料或是結 尾時,由於 readline() 會 return null,因此利用 != null 作為判斷的依據。
- readLine() 會丟出 IOException(); 而不是 FileNotFoundException。

4 Place the Fragments into program, so that the program will get lines from a text file, display them, and then close the resources.

```
1.    import java.io.*;
2.    public class ReadFile {
3.        public static void main (String[] args) {
4.        try {
5.            File x1 = new File("MyText.txt");
6.            [1. Place here] x2 = new [2. Place here](x1);
7.            [3. Place here] x4 = new [4. Place here](x2);
8.            String x3 = null;
9.            while(( x3 = x4.[5. place here]()) != null ) {
10.               System.out.println(x3);
11.           }
12.           x4.close();
13.       }
14.       catch (Exception ex) {
15.           ex.printStackTrace ();
16.       }
17.       }
18.  }
```

Code Fragments:

A. BufferedReader	B. StreamReader	C. FileReader	D. readLine

E. readLn	F. read	G. closeFile	F. close

Try to fill them:

1.＿＿＿＿＿　　2.＿＿＿＿＿　　3.＿＿＿＿＿　　4.＿＿＿＿＿　　5.＿＿＿＿＿

ANS:

1.　C　　　2.　C　　　3.　A　　　4.　A　　　5.　D

- 本意是讀取文件檔 **MyText.txt** 的資料，並將之呈現出來。
- 其中先利用 **FileReader** 讀取在將之放入 **BufferedReader** 中，以減低不斷讀取硬碟的負擔。
- **BufferedReader** 並無提供 **readLn** 方法
- **read()** 用來讀取單一字元，回傳型態為 int，而 **readLine** 則一次讀取一行且回傳型態為 **string**。