University of Cape Town ~ Department of Computer Science

Computer Science 1016S ~ 2007

Theory Test 1A

Marks : 30

Time : 40 minutes

Instructions:

    a) Answer all questions.

    b) Write your answers in the space provided.

    c) Show all calculations where applicable.

Question 1: Recursion and Exceptions [15]

```java
import java.util.Scanner;
   public class RecursiveTest {
     public static void main(String[] args) {
       Scanner keyboard = new Scanner(System.in);
       System.out.print("Enter the height:");
       int height = keyboard.nextInt();
       Tri(height,"");
     }

     public static String Line(int n) {
     //...

     //returns a line of characters of length n
     }

     public static void Tri(int n,String offset) {
      if (n==1)
         System.out.println(offset+Line(1));
       else {
         Tri(n-1, offset+" ");
         System.out.println(offset+Line(n*2-1));
       }
     }
   }
```

For different values of the height parameter, the program listed above produces output as follows.

```
Enter the height:1
L


Enter the height:2
 L
LOL


Enter the height:3
  L
 LOL
LOLOL
 etc.
```

a) What changes to the program above would make it infititely recursive?

[1]

_____

removing the stopping case/base case – if (n==1)

b) Infinitely recursive programs usually generate a `StackOverflowException`. Explain what the stack is and why it overflows with infinite rec.

[2]

_____

_____

A stack is a memory structure analogous to a stack of paper which computers use to keep track of recursion.For each recursive call, a stack frame is placed on the stack. [1] If there are too many, then the stack will attempt to grow beyond its limit, resulting in an error condition known as a stack overflow [1]

c) Is a `StackOverflowException` a checked exception? Justify your answer.

[2]

_____

_____

No.[1] It is not caught and did not have to be specified in the RecursiveTest method. [or some valid justification] [1]

d) Write a **recursive definition** for the method `Line`.

[4]

_____

_____

_____

_____

```
public static String Line(int n) {
    if (n==1)  // 1 mark
      return("L");
    if (n==2)
      return("LO");  // 1 mark
    else
      return("LO"+Line(n-2));  // 2 marks
}  // note no marks for an iterative solution!
```

e) Now, using the Tri method above, write a **recursive method** Tree(int n, String offset) that , for an interger n, produces output as follows:

Tree(1,"") displays:

 L
LOL

Tree(2,"") displays:

 L
 LOL
 L
 LOL
LOLOL

Tree(3,"") displays:

 L
 LOL

 L
 LOL
 LOLOL
 L
 LOL
 LOLOL
LOLOLOL

And so on... You are given the method header, just fill in the body of the method. (Note that no marks will be given for iterative solutions!)

public static void Tree(int n, String offset) {

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

}

[5]

```
public static void Tree(int n, String offset) {
    if(n==1)
        Tri(n+1,offset); // 1 mark
    else {
        Tree(n-1, offset + " "); // 2 marks
        Tri(n+1,offset); // 2 marks
    }
```

Question 2: File I/O and Exceptions [15]

```java
import java.io.*;
import java.util.Scanner;
public class IOTest1 {
  public static void main(String[] args) throws
                        FileNotFoundException {
    int [] arr = {1,5,8,57,100,119,300,401,1000,12000};
    PrintWriter printW = null;
    printW = new PrintWriter(new FileOutputStream("fileA.txt"));
    Scanner scan = new Scanner(new FileInputStream("fileB.txt"));
    int k = scan.nextInt();
    int p = look(arr,0,10,k);
    printW.println(p);
  }

  public static int look(int[] a, int first, int last, int key)  {
    int mid,result=0;
    boolean found = false;
    while ( (first <= last) && !(found) ) {
       mid = (first + last)/2;
       if (key == a[mid]) {
          found = true;
          result = mid;
       }
       else if (key < a[mid]) last = mid - 1;
       else if (key > a[mid]) first = mid + 1;
    }
    if (first > last) result = -1;
    return result;
  }
```

a) List three conditions under which a `FileNotFoundException` will be thrown by the above program.

[3]

_____

_____

_____

fileB.txt does not exist [1], fileA.txt exists and has ppermissions set so that you can't write to it [1] , fileB.txt exists and has permission set so that you can't read from it [1]

f) Rewrite the main method so that the `FileNotFoundException` is handled.

[4]

```java
public static void main(String[] args) {
int [] arr = {1,5,8,57,100,119,300,401,1000,12000};
```

_____

_____

_____

_____

_____

_____

_____

_____

```
}
     PrintWriter printW = null;
          try {  // 1 mark
             printW = new PrintWriter(new FileOutputStream("fileA.txt"));
             Scanner scan = new Scanner(new FileInputStream("fileB.txt"));
             int k = scan.nextInt();
             int p = look(arr,0,10,k);
             printW.println(p);

          }
           catch (FileNotFoundException e) { //2 marks
           System.out.println(e.getMessage()); //1 mark
           }
```

b) What is a stream in Java IO?

[2]

_____

_____

*A stream is **an object** that enables the flow of data between a program and some I/O device or file [2]*

c) Given an example of an output stream object from the program listed above.

[1]

_____

d) Describe **clearly** in English what this program does.

[3]

_____

_____

_____

_____

The program reads in a number to seaqrch for in the array arr from fileB.txt.[1] It then performs a binary search [1] for the number. The position of the number [1] is the written for fileA.txt

e) If, before the program is run, "`fileA.txt`" contains the data:

`100`

and , "`fileB.txt`" contains the data:

`57`

Write down the **exact** contents of each of these files after the program is run. [2]

_____

_____

fileA: 3 fileB: 57