CITS1001 exam 2013
SAMPLE SOLUTIONS
Official cover page to be substituted here

May 10, 2013

Question 1 (10 marks)

Write a Java class, called `Book`, to manage information about a particular book: its title, author, publication year, ISBN number and the text of the book (as lines of text).

a) Write skeleton code for fields and a constructor of the `Book` class. The types of the fields and signatures of the constructor should be shown clearly. But you are *not* required to write the code for the constructor: just show its signature.

b) Write a method `wordCount` for the `Book` class that returns the total number of words in the text of a book. Use a simple definition for a “word”, such as any white space separated string.

Answer Question 1a and 1b here and on the next page

Answer Question 1a and 1b here

```java
// SAMPLE SOLUTION CLASS
import java.util.HashMap;
import java.util.ArrayList;

public class Book
{
// 6 marks for suitable choice of types and
// for correct class structure (with imports)
// and a sensible constructor sig
        String author;
        String title;
        int publicationYear;
        String isbn;
        ArrayList<String> text;

        public Book(String author, String title)
        {
            // assuming other fields can be set later
            // code not required for this question
        }

// 4 marks for this method: signature and body − use foreach
// and use of local ArrayList
    public int wordCount() {
        int count = 0;
        for (String line : text) {
            count = count + line.split(” ”).length;
        }
        return count;
    }
}
```

Question 2 (10 marks)

2a) Write a method to input a list of words and returns a string in which all but the last word are replaced with the initial letter. The input represents one or more given names, and ends with the family name. For example, given the input string `Hilda Mary Primrose Smith` the method should return the string `H. M. P. Smith`.

2b) Suppose you have a table of internet movies and the number of downloads for each movie, defined by a variable `HashMap<Movie, Integer> downloads`. Write the signature and body for a method `mostDownloaded` that returns the `Movie` object that has the most downloads.

Answer question 2a here

```
public String nameSummary(String namelist)
   {
       StringBuffer name = new StringBuffer();
       String[] parts = namelist.split(” ”);
       for (int i=0; i < parts.length-1; i++) {
           name.append(parts[i].charAt(0) + ”. ”); // initials
       }
       name.append(parts[parts.length-1]); //last name
       return name.toString();
   }
```

Answer Question 2b here

```
    //4 marks for sensible signature choices and correct code
    public String mostFrequent(HashMap<String, Integer> wordtable)
    { // assume at least one word in the table, and
      // if equal tops, will return the last one
        int maxval = 0;
        String maxString = ””;

        for (String s : wordtable.keySet())
        {
            int v = wordtable.get(s);
            if (v >= maxval) {
                maxString = s;
                maxval = v;
            }
        }
        return maxString;
    }
```

Question 3 (10 marks)

A method with signature `boolean palindrome(String str)` decides whether a given string is a palindrome: that is, whether the string reads the same backwards as forwards.

3a) Write 4 (four) JUnit `assertEquals` statements to test the correctness of an implementation of this method. Include a brief Java comment for each test to justify the chosen case.

3b) Write Java code to implement the `boolean palindrome(String str)` method.

Answer Question 3a here

```java
// 1 mark each for contrasting example - only 4 required

assertEquals(true, palindrome("abba")); // even length
assertEquals(true, palindrome("deacaed")) ; //odd length
assertEquals(false , palindrome("cEddeC")) ;
//mixed case - this could also be true if toLowerCase is used
assertEquals(true, palindrome("a")) ; //single
assertEquals(true, palindrome("")) ; //empty
assertEquals(false , palindrome("Abc")) ; // neg case
```

Answer Question 3b on the next page

Answer Question 3b here

```
//5 marks: 3 for main loop, 2 for exception conditions and efficiency
public boolean palindrome(String str)
    {
        int len = str.length();
        if (len > 1) {
            int mid = len/2 - 1;
            for (int i = 0; i <= mid; i++) {
                if (str.charAt(i) != str.charAt(len-1-i)) {
                    return false; //mismatch
                }
            }
            return true; //all chars match
        } else {
            return true; //0 or 1 chars only
        }
    }
```

Question 4 (10 marks)

Compare and contrast the methods `fib1` and `fib2` given below. Each method calculates part of the mathematical Fibonaci sequence. Comment on the strengths and weakness of each implementation, using the criteria of correctness, design, efficiency, and readability. Use the line numbers to refer to parts of the code in your answer.

```
 1      private int fibonacci(int n)
 2      {
 3          int val;
 4          if (n==0 || n==1) {
 5              val = 1;
 6          } else {
 7              val = fibonacci(n-1) + fibonacci(n-2);
 8          }
 9          return val;
10      }
11
12      public int[] fib1(int n)
13      {
14          int[] xy = new int[n];
15          for (int i=n; i>0; i--)
16          {
17              xy[i] = fibonacci(i);
18          }
19          return xy;
20      }
21
22      public int[] fib2(int n)
23      {
24          int[] fibs=new int[n]; fibs[0]=1; fibs[1]=1;
25          for (int i=2; i<n; i++)
26          {
27              fibs[i] = fibs[i-1] + fibs[i-2];
28          }
29          return fibs;
30      }
```

Answer Question 4 here

sample answer: 1 mark per reasonable point,
organisation should address the given criteria

correctness:
line 15: runtime error will try to access xy[n] causing
intoverflow exception run time error
(should start at n-1)
line 15: loop bound error: stops one place too early
(never sets xy[0]
fib2 is correct

design:
line 17: use of helper method fibonacci is good design (and necessary)
line 17: perhaps it should not be private because other classes might use it
line 17: maybe all methods should be static because they are utility methods
- do not affect any object state through field variables
line 3, 5,7,9: explanatory variable val is good
cube 15: no reason to fill the array from highest to lowest, so unnecessary
complexity here

efficiency:
lines 1 to 10 : this recursive fib function is very inefficient since
each fib call makes n*2 method calls
lines 22 to 29 : fib2 is an efficient implementation of fib function
because values are calculated once and then the results accessed
from the array

readability:
line 14: poor choice of variable name
line 23: poor layout with lack of space and newline
all lines: layout brackets on the right lines and
correct space around operators
all methods: no javadoc comments - does not help readability

Du’s comment: Bonus points for students who notice that fib(n) is the first 5 fibs
not the fibonacci number for n=5