

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

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

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

Question 4

(10 marks)

Compare and contrast the methods `fib1` and `fib2` given below. Each method calculates part of the mathematical Fibonacci 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    }
```