UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

APS105 — Computer Fundamentals
Final Examination — April, 2006

Examiner: John Carter

Duration: 2.5 h

Exam Type: A
This is a "closed book" examination; no aids are allowed.

Calculator Type: 4
No calculators are allowed.

All questions are to be answered on the examination paper. If the space provided for a question is insufficient, extra space is provided at the end of the examination. If you use this extra space, please indicate clearly which question(s) you have answered there.

The examination has 13 pages, including this one.

The marks allocated to the questions, out of a total of 135, are shown in the question headings.

You must use the Java programming language to answer programming questions. Unless specifically directed otherwise, you may use any of the methods from the `Math`, `String`, and `In` classes.

Name _____

Student Number _____ ecf login _____

MARKS

1	2	3	4	5	6	7	8	9	10	Total
/10	/20	/15	/15	/15	/15	/5	/10	/15	/15	/135

1. [10 Marks]

Circle the correct answer for each of the following statements. Each correct answer is worth one mark. Each incorrect answer will result in a deduction of one mark.

(a) **True** or **False:** A Java compiler will recognize all syntax errors in a program.

(b) **True** or **False:** An `int` value is stored using 32 bits.

(c) **True** or **False:** At the time of a method call, an argument automatically receives a value from the corresponding parameter.

(d) **True** or **False:** The declaration `int[][] table = new int[10][];` is valid.

(e) **True** or **False:** If we fail to write a `toString` method for a class, Java supplies one that returns the string `""`.

(f) **True** or **False:** The value of the expression `"Canada".toUpperCase()indexOf('A')` is 2.

(g) **True** or **False:** The `String` class has an instance method that returns the length of a string.

(h) **True** or **False:** Recursive methods are usually more efficient than non-recursive ones.

(i) **True** or **False:** An ordered linked list can be searched using a binary search.

(j) **True** or **False:** An inorder traversal of a binary search tree visits the nodes in ascending order.

2. [20 Marks]

Each part of this question is worth two marks.

(a) Show what would be printed by the statement `System.out.println("Sum is:\n" + 3.4 + 5);`

(b) Write $\left|\frac{4}{3}\pi r^3\right|$ as a Java expression.

(c) Write a statement that will assign to the `int` variable `choice` a random value from the set $\{2, 4, 6, 8, 10\}$.

(d) Simplify the expression `!(x >= y || x != z)` as much as possible.

(e) What do we mean when we say that a method is *overloaded*?

(f) Trace a binary search as it seeks the value 61 in the array called `list` shown below. To show your trace, print the value of the array element examined at each stage of the search.

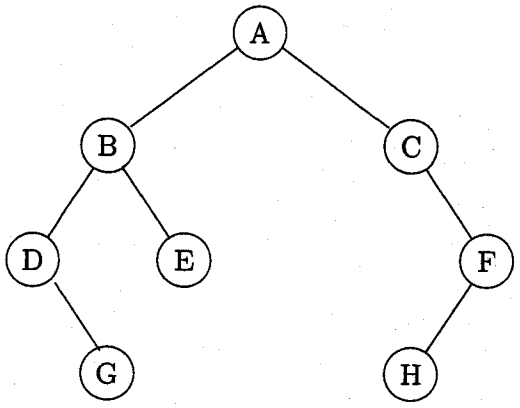
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
`list`	31	35	40	41	52	55	58	63	68	71	79	82	87	90	94

(g) Suppose that an array initially contains the values `{7,3,5,2,8}`. If the array is to be sorted into ascending order using insertion sort, show the contents of the array after each of the first two passes of the sort.

(h) Suppose that an array initially contains the values `{28,36,27,32,25,40,24}`. If the array is to be sorted into ascending order using quicksort as discussed in class, show the contents of the array after the first pass of the sort.

(i) What is a stack?

(j) The diagram shows a binary tree. In what order would the nodes be visited by a preorder traversal?



3. [15 Marks]

Write a complete Java program that first prompts the user for a value of x, reads x (as a `double`) and then computes and prints the value of e^x without using either of the methods `exp` or `pow` from the `Math` class. The value of e^x is given by the following *power series*.

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots$$

In computing the value of e^x, your program should continue to add terms of the power series until it reaches a term whose absolute value is less than 10^{-15}.

4. [15 Marks]

Suppose that points in the plane are represented by objects of a `Point` class with the following fields:

```
private double x;
private double y;
```

The `Point` class also contains accessor methods `getX` and `getY` that return the values of `x` and `y`. Suppose also that line segments are represented by objects of the `Segment` class with the following fields:

```
private Point p;
private Point q;
```

(a) Write a constructor method for the `Segment` class that could be called by the statement

```
Segment s = new Segment(p1,p2);
```

where `p1` and `p2` are `Point` objects. The constructor would create a new `Segment` object with endpoints `p1` and `p2`.

(b) Write an instance method `length` for the `Segment` class. The method should return, as a `double`, the length of a segment.

(c) Write an instance method `slope` for the `Segment` class. The method should return, as a `double`, the slope of a segment. If the segment is vertical or of length zero, the method should return `NaN`.

(d) Write an `equals` method for the `Segment` class. Two segments should be considered equal if their lengths differ by less than 10^{-15} and their slopes either differ by less than 10^{-15} or are both undefined.

5. [15 Marks]

Complete the definition of the method `firstWord` whose header is shown below. The method should return the word in the string `s` that would be first if the words of the string were placed in alphabetic order. Assume that `s` consists only of lower-case words separated by single blanks and that there are no blanks at the beginning or end of `s`. You may *not* use methods of the `StringTokenizer` class in your solution.

```
public static String firstWord (String s)
```

6. [15 Marks]

Complete the definition of the method `closest` whose header is shown below. The method should return the smallest non-negative difference between values in the `int` array `list`.

As examples,

if `list = {3,13,20,10}` the method should return the value 3 (13 and 10 are closest)
if `list = {-4,5,2,-8,7,-1}` the method should return the value 2 (5 and 7 are closest)
if `list = {-5,7,-5,4}` the method should return the value 0 (−5 and −5 are closest)

```
public static int closest (int[] list)
```

7. [5 Marks]

The Tchebyshev polynomials are defined as follows:

$$T_n(x) = \begin{cases} 1 & \text{if } n = 0 \\ x & \text{if } n = 1 \\ x \cdot T_{n-1}(x) - T_{n-2}(x) & \text{if } n > 1 \end{cases}$$

(a) Find $T_2(x)$ and $T_3(x)$ in simplified form

(b) Find the value of $T_4(2)$ by hand.

8. [10 Marks]

Write a recursive method `count` that has two parameters: an `int` value `n` and an `int` value `digit`. You may assume that `n` is non-negative and that `digit` is a single digit in the range 0 to 9. The method should return a count of the number of times that `digit` occurs in `n`. For example, `count(7088785,8)` should return the value 3. Note that your method *must* be recursive; no credit will be given for a non-recursive solution.

9. [15 Marks]

Suppose that linked lists are maintained in the usual way seen in class using the class `List` and the inner class `Node` whose fields are shown below.

```
class List
{
  private Node head;

  class Node
  {
    int info;
    Node link;
  }
}
```

Suppose further that all linked lists are maintained in non-decreasing order. Write an instance method `simplify` that deletes any duplicate items in a list. If, for example, before calling `simplify`, a list contains

13 15 15 17 17 17 19 22 25 25 28

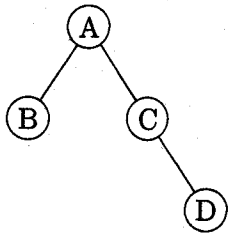
then, after `simplify` has been called, the list should contain

13 15 17 19 22 25 28

```
public void simplify ()
{
```

10. [15 Marks]

Suppose that we define the *level* of a node in a binary tree as the number of edges on the path from the node to the root. For example, in the binary tree shown in the diagram, the levels of A, B, C, and D are, respectively, 0, 1, 1, and 2.



We can then define the *height* of a binary tree as the maximum level of any node in the tree. Thus the height of the binary tree shown above is 2. Using this definition of height, complete the definitions of a pair of methods for the `Tree` and `Node` classes that return the height of a binary tree. If the tree is empty, the value -1 should be returned.

```
class Tree
{
  private Node root;

  public int height ()
  {




  }
  class Node
  {
    int info;
    Node lChild;
    Node rChild;

    int height ()
    {
```

Extra Space *Please specify which question(s) you are answering on this page.*