UNIVERSITY OF TORONTO
FACULTY OF APPLIED SCIENCE AND ENGINEERING

APS105F — Computer Fundamentals
Final Examination — December, 2001

Examiners: John Carter, Baochun Li, James MacLean

Duration: 2.5 h

- Examination Type A: This is a "closed book" examination; no aids are permitted.
- Calculator Type 4: No electronic or mechanical computing devices are permitted.
- Write your answers in the spaces provided. You must answer using a pen.
- If more space is required, blank pages are provided at the back of the examination.
- Rough work, if necessary, can be done on the backs of the pages.
- This examination has 15 pages.
- You must use the Java programming language to answer programming questions.
- You may use any methods from the `Math` and `String` classes.
- You may assume that the following `Stdin` methods are available:
  `getInt()`, `getFloat()`, `getDouble()`, `getString()`, and `getChar()`.

Circle your lecture section:	**L0101** Carter WRF	or	**L0102** Li	or	**L0103** MacLean	or	**L0104** Carter MTR

Name _____

Student Number _____ ECF Login _____

MARKS

Question	1	2	3	4	5	6	7	8	9	10	Total
Value	20	15	15	15	15	15	15	15	5	10	140
Mark											

1. [20 Marks]

Each part of this question is worth two marks. Answers should be brief; point form is permitted.

(a) Write the simplest possible expression that is equivalent to the expression

```
!(x <= y && y <= z)
```

(b) What does it mean to declare a field as being `private`?

(c) What two properties are common to a recursive description of any process?

(d) Trace a binary search as it seeks the value 70 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`	34	39	42	45	49	53	58	61	62	66	72	77	84	86	95

(e) Suppose that an array initially contains the values `{8, 4, 2, 7, 3}`. If the array is to be sorted into ascending order using an insertion sort, show the contents of the array as it would appear after each of the first two passes of the insertion sort.

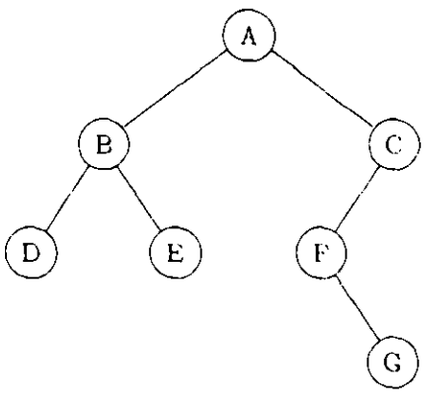
(f) Use big oh notation to write expressions for the time complexity of

selection sort: binary search:

(g) Find the base 10 representation of $1AE_{16}$.

(h) Write 58 as a base 3 numeral.

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



(j) Draw the binary search tree that would be produced by inserting the values Jerry, Elaine, George, Cosmo, and Newman (in the given order) into an initially empty tree. (Assume that the tree is to be maintained so that its entries are in dictionary order.)

2. [15 Marks]

Complete the definition of the method `maxCount` whose header is shown below. The method should return the number of *local maxima* in the array `list`. A local maximum is defined to be either a single value or a sequence of equal values greater than both the preceding and following values. For example, the sequence $3, 7, 5, 9, 9, 9, 8, 2, 4$ has *two* local maxima: 7 and 9.

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

3. [15 Marks]

Complete the definition of the method **delete** so that it returns a string in which any occurrences of any characters in the string **pattern** that appear in the string **source** have been deleted. For example, if the string **s** had the value "Toronto", then, after the call

```
s = delete(s,"to");
```

s would have the value "Trn".

```
public static String delete (String source, String pattern)
```

4. [15 Marks]

Assume that we have an array, a, whose elements are of type `int`. Using *recursion*, complete the definition of the method `butterflySearch` to search for a specific integer `key` in the array. If the key exists, the method returns `true`; otherwise it returns `false`.

The strategy of searching for the key is as follows: we first compare the first integer with the key, then compare the last integer, then the second, then the second last, and so on. We progress in this order until the key is found or all integers have been examined. In other words, for a list with n items, we use the following search sequence:

`a[0], a[n-1], a[1], a[n-2], a[2], a[n-3], a[3], a[n-4], a[4], a[n-5], ...`

Hint: You may need to write a supplementary method for this purpose. Marks will not be given if you do not use recursion.

```
public static boolean butterflySearch (int [] a, int key)
```

5. [15 Marks]

Write the complete definition of class `Circle`. Within this class, define an *inner class*, `Point`, that contains the two-dimensional coordinates of a point `(x,y)`, which are declared as instance fields of type `double`. In the `Circle` class, the centre of the circle is an object of the class `Point`, and the radius of the circle is represented as a field of type `double`. Both fields are private instance fields. Complete the definition of the inner class `Point` and the class `Circle` with the following methods. Give your answer on both this and the next page.

- Write a constructor of the class `Point`, taking two double parameters to initialize the coordinates of the point.
- Write a method `distance` of the class `Point`, taking an object of class `Point` as a parameter. This method should return a double value that shows the distance between the two points.
- Write a constructor of the class `Circle`, taking three `double` parameters to initialize a circle object.
- Write a `double` valued instance method `area`, that returns the area of its implicit `Circle` object.
- Write a class method `selectSmaller` that can be called by a statement such as

  ```
  c3 = Circle.selectSmaller(c1, c2);
  ```

  where `c1`, `c2` and `c3` are objects of type `Circle`. The method should make `c3` represent the smaller of the circles represented by `c1` and `c2` (or `c1` if `c1` and `c2` are the same size).
- Write a boolean valued instance method `isInside` that may be called by a statement such as

  ```
  boolean contained = c1.isInside(c2);
  ```

  The method should return `true` if `c1` is entirely inside `c2` and return `false` otherwise. You may take advantage of the `distance` method that is already defined in the class `Point`.

This page should be used for your answer to Question 5.

6. [15 Marks]

Suppose that linked lists are maintained 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

7. [15 Marks]

A priority queue is a queue in which each element has a priority. Two rules govern the removal of elements from a priority queue:

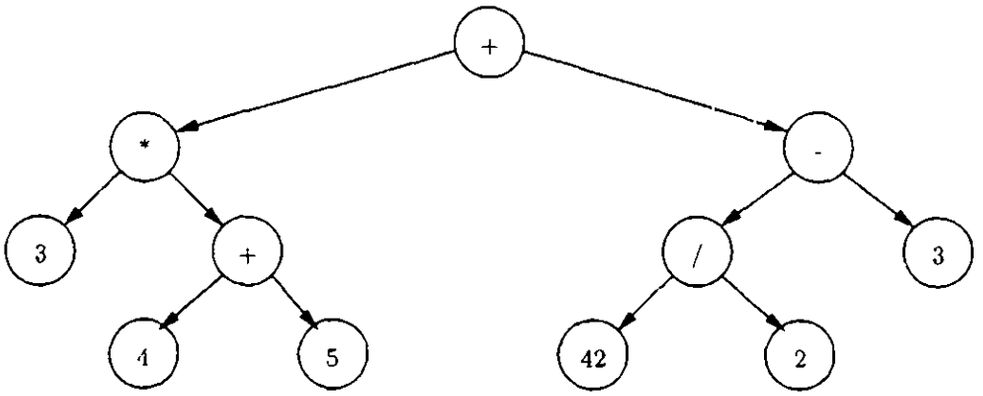
- elements having the same priority are removed from the queue in the order in which they were added to the queue and
- the removed element will always have the highest priority of all the elements waiting in the queue.

Write a class **PQueue** that implements a priority queue where each element stores a string and has a priority value in the range of 0 through 4 (inclusive), with 4 being the *highest* priority. Specifically, you must write methods to enqueue an element, dequeue an element, and check if the queue is empty. You may also require a constructor.
You may make use of the class **Queue**, which has the following ADT, in your solution:

```
class Queue
{
  public void enqueue (String s);
  public String dequeue ();
  public boolean isEmpty ();
}
```

8. [15 Marks]

A *parse tree* is a binary tree used to store an arithmetic expression to be evaluated. For example,



represents the expression $3*(4+5)+42/2-3$. Write a method named `evaluate()` in the class `parseTree` to evaluate the expression stored in a parse tree (*e.g.* 45 for the parse tree shown above) assuming:

1. each leaf node contains a `String` representing an integer (you can convert the string to an actual integer using `int Integer.parseInt(String s)`), and
2. each internal node contains one of "+", "-", "*" or "/" and internal nodes always have two children, and
3. you may declare additional methods if needed, and
4. the class `ParseTree` is defined as

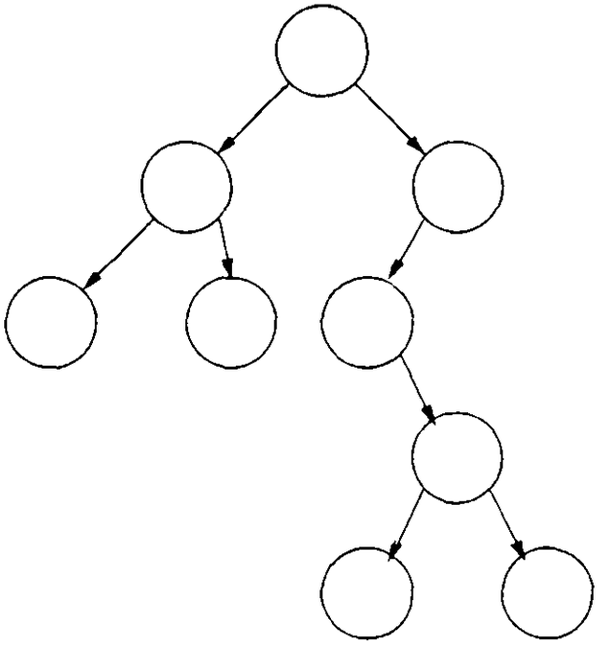
```
class ParseTree
{
  class Node
  {
    String data  ;
    Node   left  ;
    Node   right ;
  }

  private Node root ;
}
```

You may use this space and the top of the next page for your answer to this question. If your solution contains more than one method, indicate clearly the class to which each method belongs.

This space can be used for your answer to Question 8.

9. [5 Marks]

Shown below is a tree where the nodes have not yet been labelled. A post order traversal of this tree gives the output 1 2 3 4 5 6 7 8 9. Label the tree nodes with the appropriate numbers.



10. [10 Marks]

A *Trinary Tree* is a tree in which each node can have 0, 1, 2 or 3 children. Complete the method `printPostOrder` below so that it prints a postorder traversal of the elements stored in a trinary tree defined by `class Trinary`. If your solution contains more than one method, indicate clearly the class to which each method belongs.

```
class Trinary
{
  class Node
  {
    int  data   ;
    Node left   ;
    Node middle ;
    Node right  ;
  }

  private Node root ;

  public void printPostOrder()
```

Extra space *Please indicate clearly which question(s) you are answering on this page.*

Extra space *Please indicate clearly which question(s) you are answering on this page.*