University of Toronto
Faculty of Applied Science and Engineering

ECE 244F

PROGRAMMING FUNDAMENTALS

Fall 2009

Final Examination

Examiner: C. Gibson, M. Stumm, and W. Aboelsaadat

Duration: Two and a Half Hours

No aids allowed. No books. No notes. No calculators. No computers. No communicating devices.

Do not remove any sheets from this test book. Answer all questions in the space provided. No additional sheets are permitted.

There are 15 questions. The weight of each question is the same. Work independently.

Write your name and student number in the space below. Do the same on the top of each sheet of this exam book.

Last Name: ____________________ **First Name:** __________________________

Student Number: ______________________

Q1. __________	**Q9.** __________
Q2. __________	**Q10.** _________
Q3. __________	**Q11.** _________
Q4. __________	**Q12.** _________
Q5. __________	**Q13.** _________
Q6. __________	**Q14.** _________
Q7. __________	**Q15.** _________
Q8. __________	

Total $\boxed{}$

Name: ______________________________ Student Number: __________________

1. **Warmup Questions**

a. What is the output of the following code segment:

```
for (int i =1; i <10; i++)
{
      cout << i;
      if (i >=5)
            break;
      cout << i;
}
```

A: 112233445566778899
B: 12345
C: 11223344
D: 112233445

Answer:

b. The order of adding 1 to each element in a one dimensional array of N integers is

A: $O(1)$
B: $O(logN)$
C: $O(N)$
D: $O(NlogN)$
E: $O(N^2)$

Answer:

c. Which of the following C++ class member functions should be provided if class objects point to dynamic data ?

A: a destructor
B: a constructor
C: a copy-constructor
D: A and B
E: A, B, and C

Answer:

d. Given the function definition

```
void Twist(int& a, int b) {
      int c;
      c = a + 2;
      a = a * 3;
      b = c + a;
}
```

what is the output of the following code fragment that invokes *Twist* ?

```
r = 1;
s = 2;
t = 3;
Twist(t, s);
cout << r << ' ' << s << ' ' << t << endl;
```

A: 1 14 9
B: 5 14 3
C: 1 10 3
D: 1 14 3
E: none of the above

Answer:

2. True or False Questions

Identify whether the following statements are true or false by circling the correct answer:

a) The running time of the program fragment shown below is O(N): true or false?

```
sum = 0 ;
for( i=0; i<N; i++ )
    if( i>j )
        sum += 1 ;
    else
        for( k=0; k<N; k++ )
            sum -= 1 ;
```

b) The parameter to a copy constructor must be passed by reference: true or false?

c) An $O(logN)$ algorithm is generally slower than an $O(N)$ algorithm: true or false?

d) An $O(logN)$ algorithm is always slower than an $O(N)$ algorithm: true or false?

e) Given the declarations `int x=5;` and `int *ptr;` then the statement `*ptr = &x;` will cause an error: true or false?

f) The maximum number of nodes in a tree that has L levels is 2^L: true or false?

g) The largest value of a binary search tree is always stored at the root of the tree: true or false?

h) In a binary tree, every node has exactly two children: true or false?

i) Inserting an element into an unsorted list takes $O(1)$ time: true or false?

j) Tree operations typically run in $O(d)$ time where d is the number of nodes in the tree: true or false?

Name: ______________________________ Student Number: __________________

3. **Value and reference parameters**

a. Identify all errors in the code below:

```
int f( int i, int& j ) {
  …
 …
 return( j ) ;
}

…
int y ;
int x = f( 1, y ) ;
int z = f( 1, 2 ) ;
```

b. Consider the following code fragment:

```
int a = 2 ;

class A
{
  private:
    int a ;
    int b ;
  public:
    A() { a = 5 ; b = 10 ; }
    void f( int& ) ;
};
A* ap = new A ;

A::f( int& c ) {
    int a = b + c ;
    a += 5 ;
    c = a + b + c ;
}
 …
int c = 15 ;
ap->f( c ) ;
cout << c << endl ;
```

What is the output?

Answer:

4. Recursion

You are to write a function that takes in a pointer to a character array and uses recursion to print the null-terminated string out to the screen in reverse order, starting with the first character of the string but skipping every second character. For example, a call to:

```
print_reverse ("This is a sample string: 123456789.");
```

should print:

```
.8642 git lmsas iT
```

You can assume that any strings are properly-terminated C-style strings, but you are responsible for all other error handling. You are not to use any string manipulation functions (*i.e.*, strlen(), etc.), or loops of any kind.

```
void print_reverse (char * array)
{











}
```

5. Inheritance

You are given the following class definition:

```
class myDerivedClass : public myBaseClass
{
     private:
          char *     derivedArray;

     public:
          myDerivedClass();
          ~myDerivedClass();
          // ...
};
```

Write a complete copy constructor function for this class that provides a deep copy for all variables in the class (including any inherited member variables):

Name: ______________________________ Student Number: ___________________

6. Tracing

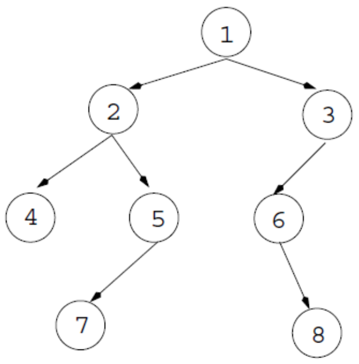
What will this program print?

```
class A {
public:
  A()
    { cout << "A ctor" << endl; }
  A(const A& a)
    { cout << "A copy ctor" << endl; }
  virtual ~A()
    { cout << "A dtor" << endl; }
  virtual void foo()
    { cout << "A foo()" << endl; }
  virtual A& operator=(const A& rhs)
    { cout << "A op=" << endl; }
};

class B : public A {
public:
  B()
    { cout << "B ctor" << endl; }
  virtual ~B()
    { cout << "B dtor" << endl; }
  virtual void foo()
    { cout << "B foo()" << endl; }
protected:
  A mInstanceOfA; // don't forgetme!
};

A foo(A& input) {
  input.foo();
  return input;
}

int main() {
  B myB;
  B myOtherB;
  A myA;
  myOtherB = myB;
  myA = foo(myOtherB);
}
```

Answer:

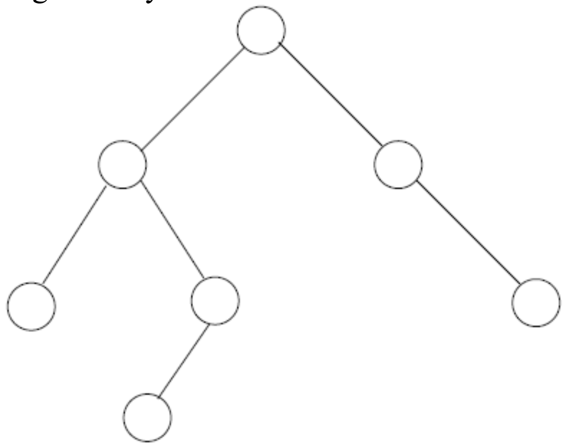
7. Trees

a. Given the tree shown below, show the order in which nodes in the tree are processed by preorder traversal.



Answer:

b. Label the following binary tree with the numbers from the set {6,22,9,14,13,1,8} so that it is a legal binary search tree. You can select the numbers from this set in any order.



c. If one is provided with the *inorder* and *postorder* traversals of a given binary tree, it is always possible to construct the tree. Draw **the** tree (i.e., only one tree) that has the following inorder and postorder traversals:

Inorder:	C B A E D F H I G K J
Postorder:	A B C D E F I K J G H

Name: ______________________________ Student Number: __________________

8. Class Design and Data Encapsulation

Complex numbers occur in many engineering applications. A complex number `c` consists of an ordered pair of real (floating point) numbers `(x,y)`. The first number is called the *real* part of the complex number. The second number of called the *imaginary* part. The addition/subtraction of two complex numbers $c_1 = (x_1,y_1)$ and $c_2 = (x_2,y_2)$ is defined as

$c_1 + c_2 = ((x_1+x_2),(y_1+y_2))$, and
$c_1 - c_2 = ((x_1-x_2),(y_1-y_2))$.

Design and write ***<u>the header file</u>*** for a `C++` class for representing complex numbers and their arithmetic. Call your class `Complex`. Your class should allow for the declaration of complex numbers only with initialization of both real and imaginary parts. It should allow for the accessing (read & write) of the real and imaginary parts. It should also overload the + and − operators to perform the addition and subtraction of complex numbers. Finally, your class should contain a function `print()` which prints the complex number as `(x,y)` to the standard output.

Most importantly, your class should enforce data encapsulation, allowing its designer to change its implementation without affecting its use.

Name: ______________________________ Student Number: __________________

9. Complexity Analysis

Determine the ***worst-case*** time complexity (expressed in big-O notation) for each of the program segments below as a function of the size of the input n. Show the details of your analysis and clearly indicate your final result.

a. The size of the input is n.

```
w=0;
for (int i=0; i < n; ++i) {
  for (int j=0; j < n*n*n; ++j) {
    for (int k=0; k < n*n*n*n; ++k) {
       w = w + 1;
    }
  }
}
```

Answer:

b. The size of the input is n.

```
for (int i=0; i < n; ++i) {
    for (int j=0; j < i*n ; ++j) {
         O(1)
    }
}
```

Answer:

c. The size of the input is n, which is a power of 2.

```
int mystery (int k, int n) {
   int x;
   int y;
   if (n <= 1) {
      return(0);
   } else {
      for (int i=0 ; i < n ; ++i) {
          O(1) // to determine the value of k
      }
      x = mystery (k, n/2); // integer division
      y = mystery (n-k, n/2); // integer division
      return (x+y);
   }
}
```

Answer:

Name: _______________________________ Student Number: ___________________

2. Scope and parameters

Show the output of the following program in the box provided below. Show your work for partial credit.

```
#include <iostream>
using namespace std;

char confuse (char c1);
void mixup (char &c1, char c2, char c3);

char c4 = 'F';

int main () {
   char c1 = 'H', c2 = 'C', c3 = 'E';
   mixup( c2, c1, c3);
   {
      char c3 = 'G';
      cout << c1 << c2 << c3 << c4 << endl;
   }
   cout << confuse(c4) << c1 << c2 << c3 << c4 << endl;
   return 0;
}

char confuse (char c1) {
   const char c2 = 'O';
   char c3 = c1;
   if ( c1 >= 'N' ) return c3;
   else             return c2;
}

void mixup (char &c1, char c2, char c3) {
   cout << c1 << c2 << c3 << c4 << endl;
   c1 = confuse(c3);
   c2 = 'S';
   c3 = 'I';
   c4 = confuse(c2);
}
```

Output line 1:	
Output line 2:	
Output line 3:	
Output line 4:	

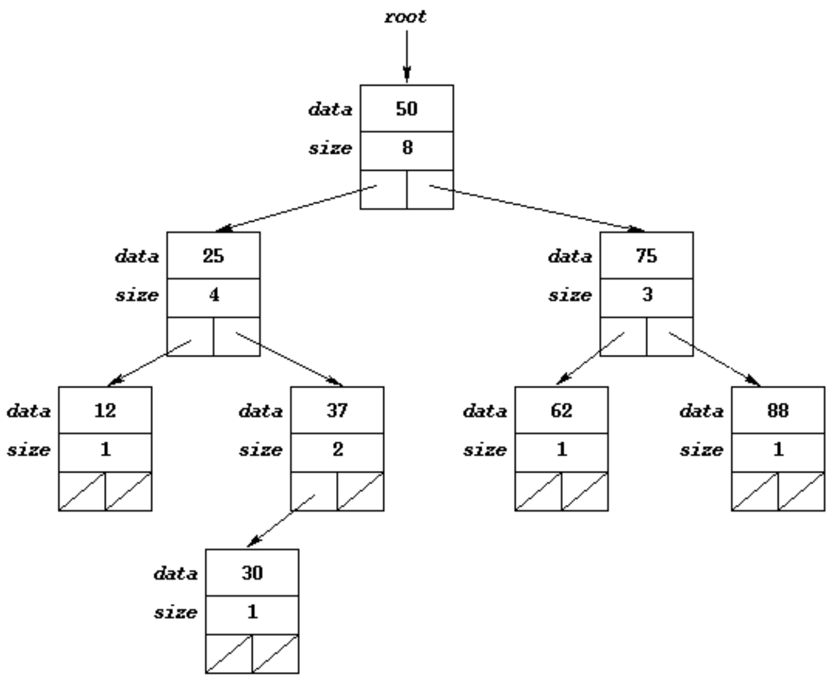
Name: ______________________________ Student Number: __________________

10. Binary Trees II

Assume that binary trees are implemented using the following declarations.

```
struct TreeNode
{
    int        info;     // data stored
    int        size;     // size of subtree (root included)
    TreeNode * left;     // left subtree
    TreeNode * right;    // right subtree

    TreeNode(int val, TreeNode * lptr, TreeNode * rptr)
          : info(val), size(1), left(lptr), right(rptr) { }
};
```

a. Write a function `SetSize()` in the space provided on the next page where its header is given. `SetSize()` should set the `size` field of every node in its tree parameter to the number of nodes in that node's sub-tree (including itself). The number of nodes in a tree is equal to the number of nodes in its left sub-tree plus the number of nodes in its right sub-tree plus one. For example, the following picture shows the result of the call `SetSize(root)`:



14. Hash Tables

Assume you are given a hash table with size M = 11, and the following hash function:

h(key) = (key % M) (Note: the "%" is the modulus operator in C++)

Assume that **quadratic probing** is used to resolve key collisions, and that the hash table is initially empty. If i is the initial hash index generated by the hash function above, quadratic probing examines locations $i + (k)^2$, where $k = 0, 1, 2, 3, \ldots$ until an empty location is found.

Name: ______________________________ Student Number: ___________________

Show the final contents of the hash table after all of the following operations have been performed, in the sequence shown. Next to each operation, indicate the number of probes performed when inserting that value.

insert 7

insert 3

insert 18

insert 29

insert 14

insert 40

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	