CS 275	**Advanced Programming Techniques in C**	Test 3
Name:	Fall 1997	Max Pts: 40

Important: This is an open book test. You can use any books, notes, or paper. *Do not log into the computer during the test.* Any calculations and rough work can be done on the back side of the test pages. If there is a syntax error in any program segment, just write it down and you will get full credit for the problem. You will lose five points for not writing your name.

1. [5 pt] It is considered good style to put the code for templates in C++ in a `.h` file to be included into a `.c` file, rather than compiling the code in a `.c` file. Why? Why do you get a linking error when you compile the template code in `.c` file and try to link it with the other code that uses the template?

2. [5 pt] When overloading the operator `<<` or `>>`, why must we declare the functions as friends? Why must these functions *not* be declared as the member functions of the class? Can you write either of these functions as *not friends*?

3. [5 pt] What is the purpose of a `protected` data member? Why is it required?

4. [5 pt] Why is it essential to pass the parameter for a copy constructor by reference?

5. [5 pt] What is the scope of a `private` and a `public` class member?

6. [5 pt] What is encapsulation? Why is it useful?

7. [10 pt] What is the output of the following program?

```
class A
{
    int n;
    public:
        A ( void );
        A ( int a );
        void f ( void );
        void g ( void );
        int h ( void ) const;
        void k ( void ) const;
};

A::A ( void ) { n = 0; }
A::A ( int a ) { n = a; }
void A::f ( void ) { n++;  }
void A::g ( void ) { f(); n *= 2; f(); }
int A::h ( void ) const { return n; }
void A::k ( void ) const { cout << n << endl; }

main()
{
    A x;
    A y ( 2 );
    A z = y;
    A w = A(3);
    x.f(); y.g(); z.f(); w.g();
    w.k()
    A e ( x.h() + y.h() + z.h() );
    e.k();
    return 0;
}
```

8. [6 pt] Consider the following class definition:

```
class employee
{
    char * name;
    int age;

    public:
employee ( void )
{
    name = new char[30];
    strcpy ( name, "John Doe" );
    age = 35;
}

~employee ( void )
{
    delete[] name;
}
};

main()
{
    employee e1;
    employee e2;
    employee *e3;

    e1 = e2;
    e3 = &e1;

    return ( 0 );
}
```

Is there any memory leak in this code? Dangling pointers?