ECE2036	**Fall Semester, 2012**
Midterm Exam 2	
Nov 7, 2012	*NAME:*____

OPEN BOOK, OPEN NOTES, NO INTERNET ACCESS PLEASE!.

1. Fourier Transform Lab **15 points**.

   (a) In the “efficient” implementation of the FFT lab, we used three nested loops. for the main computation of the transform. Explain in three short sentences what each of the three loops did. (Not pseudo code, explanation please).

   (b) The “inefficient” version used only two nested loops, and ran in time proportional to N^2, where N is the total number of time-domain samples. Explain why the efficient version (which had three loops, not two) runs in $Nlog_2N$ time

   (c) Assume that $N = 256$, and the inefficient vesion ran in 10 seconds. How long would the efficient version take for the same input dataset?

2. Path Loss with Threads **25 points**.

   (a) Normal multi-threaded applications typically require the use of a *mutex* to protect against simultaneous update of shared variables. Did you need any mutexes in your multi-threaded path loss solution? Explain why or why not.

   (b) Suppose we had transmitter antenna gain (G_t) of 5dBm and receiver antenna gain (G_r) of 5dBm. What would be different in the final picture of the signal strength image?

   (c) Suppose we had 0 for both (G_t) and (G_r) (which we did in the assignment) but used a carrier frequency f of 24GHz instead of 2.4GHz. What would be different in the final picture of the signal strength image?

3. Vector container implementation **20 points**

   A copy of the GFRVec implementation is attached.

   (a) Not sure we need questions on this, what do you think? Probably long enough already.

   (b) I think your short answer questions should replace this. Perhaps about the Vector/Deque usage or the GFRVec implementation.

4. Inheritance and Virtual Functions **25 points**.

   What is printed by the attached `inheritance` program. Hint, there are 11 "Hello from" outputs.

```
#include <iostream>

using namespace std;

class Base
{ // Define a base class
public:
  virtual void Sub1() = 0;
  virtual void Sub2();
  virtual void Sub3();
          void Sub4();
};
class A : public Base
{ // Class A derives from Base
public:
  void Sub2();
  void Sub4();
};

class B : public A
{// Class B derives from A
public:
  virtual void Sub1();
          void Sub2();
};

class C : public Base
{ // Class C derives from Base
public:
  virtual void Sub1();
  virtual void Sub4();
};

// Base Class Methods
void Base::Sub2(){ cout << "Hello from Base::Sub2()" << endl;}
void Base::Sub3()
{
  cout << "Hello from Base::Sub3()" << endl;
  Sub1();  // DON'T MISS THIS CALL IN YOUR ANSWER
}
void Base::Sub4(){ cout << "Hello from Base::Sub4()" << endl;}

// Class A Methods
void A::Sub2() { cout << "Hello from A:Sub2()" << endl; }
void A::Sub4() { cout << "Hello from A:Sub4()" << endl; }
// Class B Methods
void B::Sub1() { cout << "Hello from B:Sub1()" << endl; }
void B::Sub2() { cout << "Hello from B:Sub2()" << endl; }
// Class C Methods
void C::Sub1() { cout << "Hello from C:Sub1()" << endl; }
void C::Sub4() { cout << "Hello from C:Sub2()" << endl; }


// A Helper Subroutine
void  Sub(Base& x)
{
```

Program inheritance.cc

```
  x.Sub1();
  x.Sub2();
  x.Sub3();
}

void AnotherSub(A& a)
{
  a.Sub1();
  a.Sub2();
  a.Sub4();
}

int main()
{
  //A a; won't compile
  B b;
  C c;
  Sub(b);
  Sub(c);
  AnotherSub(b);
}
```

Program inheritance.cc (continued)