ECE 462 Fall 2011, Second Exam

DO NOT START WORKING ON THIS UNTIL TOLD TO DO SO.

You have until 9:20 to take this exam.
Your exam should have 10 pages total (including this cover sheet). *Please let Prof. Midkiff know immediately if it does not.*

This exam is open book, open notes, but no electronics. If you have a question, please ask for clarification. If the question is not resolved, state on the test whatever assumptions you need to make to answer the question, and answer it under those assumptions. *Check the front board occasionally for corrections.*

Name:

Student ID:
Consider the Java program for the next two questions:

class B {
    public B( ) { }

    public void foo(int i, double j, int k, B b1) {
        System.out.println("foo int double B2");
    }

    public void foo(int i, int j, B b2) {
        System.out.println("foo int int B2");
    }

    public void foo(double j, int i, B b2) {
        System.out.println("foo double int B2");
    }
}

public class Main {
    public static void main(String args[ ]) {
        B b = new B( );
        b.foo(3, 4.0, b);
        b.foo(4.0, 3, b);
        b.foo(4, 3, b);
        b.foo(4.0, 3.0, b);
    }
}

Q1 (6 pts): The compiler complains about not being able to find a matching symbol for two of the calls. Which call [sic] does the compiler complain about?
   *b.foo(3, 4.0, b); and b.foo(4.0, 3.0, b);*

Q2 (6 pts): Change the calls the compiler complains about to make it a legal call.
One possibility is to change *b.foo(3, 4.0, b);* to *b.foo(3.0, 4, b);* and to convert *b.foo(4.0, 3.0, b);* to *b.foo(4, 2, b);* Other conversions could be done.
Consider the C++ program below for the next two questions:

class User {
    int ii1, ii2;

public:
    User(int, int);
    User( );
    void print(int, double);
    void print(int, double, double);
    void print(double, int);
    void print(int, int);
    void print(int, int, double);
    ~User( );
};

int main(int argc, char * argv[ ])
{
    int i1 = 1;
    double d1 = 1.0;
    User u(1,2 );
    u.print(i1, i1);
    u.print(d1, d1);
    u.print(i1, i1, i1);
    u.print(i1, d1, i1);
}

Q3 (6 pts): The compiler complains that a call to one or more overloaded functions is ambiguous. Which function calls are ambiguous?

u.print(i1, i1); // uniquely matches void print(int, int);
u.print(d1, d1); // matches both void print(double, int); and void print(int, double); with one conversion and
u.print(i1, i1, i1); // uniquely matches void print(int, int, double); with one conversion
u.print(i1, d1, i1); // uniquely matches void print(int, double, double); with one conversion.

Q4 (6 pts): Change the calls so that they will call one of the declared functions.
One possibility is to change it to u.print((int) d1, d1);

Q5 (6 pts): B is a class, and the class D inherits from B. In D.h, there is the following declaration:

void foo(B*);

and in main there is the code:

D* d1 = new D( ); D* d2 = new D( );
d.foo(d2);

Is it possible for the call d.foo(d2); to call the void foo(B*); declared in D.h? Yes — downcasting the D
pointer to d2 will allow foo(B*) to be matched.
Q6 (6 pts): What is printed by the program below?

```java
public class C1 {
    public int i;
    public C1() {i = 20;}
}
public class C2 {
    public C2() {}
    public C1 foo(C1 z1) {
        z1.i = 30;
        z1 = new C1();
        return z1;
    }
}
public class Main {
    public static void main(String args[]) {
        C1 x1 = new C1();
        C2 x2 = new C2();
        C1 y1 = x2.foo(x1);
        System.out.println(x1.i + " " + y1.i);
    }
}
```

30 20

Q7 (6 pts): What is printed by the program below?

```java
public class C1 {
    public C1() {}
    public void foo(int i) {
        i = -5;
    }
}
public class Main {
    public static void main(String args[]) {
        int i = 50;
        C1 x1 = new C1();
        x1.foo(i);
        System.out.println(i);
    }
}
```

50
Q8 (6 pts): What is printed by the program below?

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

using namespace std;

class C {
pubic:
    C( ) ;
    C(int);
    ~C( );
    int f ;
};

class B {
pubic:
    B( );
    ~B( );
    void foo(C&);
};

B::B( ) { }
B::~B( ){ }
void B::foo(C& c) {
    c.f = -10;
}

C::C(int i) {f=i;}
C::~C( ){ }

int main(int argc, char * argv[ ])
{
    B* b = new B( );
    C c1(2);
    C& c = (C&) c1;
    b->foo(c);
    cout << c.f << endl;
}

-10
```
The following program is used for Questions 9 and 10 below.

```cpp
#include <string>
#include <iostream>
#include <sstream>
using namespace std;

using namespace std;

class B {
public:
    B( );
    ~B( );
    void foo(int*);
};

B::B( ) { }
B::~B( ) { }

void B::foo(int* p) {
    int j=90;
    *p = 20;
    p = &j;
}

int main(int argc, char * argv[]) {
    int i = 50;
    int* ptr = &i;
    B* b = new B( );
    b->foo(ptr);
    cout << i << endl;
    cout << *ptr << endl;
}
```

Q9 (6 pts): What is printed for i?
20

Q10 (6 pts): What is printed for *ptr (circle the right answer)

1. nothing – there will be a segmentation fault because *ptr points to something on foo’s stackframe.
2. 90 if foo’s stackframe is available, a random variable if that location on the stack is overwritten, and maybe a segmentation fault.
3. 20 this is what is printed.
4. 50

Note that 90 is not printed, and no segmentation faults occur because the value of ptr, and not the address of pointer is passed into foo, and consequently the value of ptr in main is unchanged.
Q11 (9 pts):

1. In the code below, “+” is invoked in main, with \( u \) and \( v \) being the operands of the “+” operation. Is \( u \) or \( v \) the argument of the MyComplex ‘‘+’’ function? \( v \)

2. The MyComplex “+” operation is a member function, and must be invoked by an object. In the same line in main, is it invoked on object \( u \) or object \( v \)? \( u \)

3. Could the “<<” friend function be made a member function of the MyComplex class? Why or why not (and try to keep the answer short.)? no, because it would be invoked on the object \texttt{cout}, which is an output stream and not a MyComplex object, and so the MyComplex definition would not be used and we would get an error because the class for \texttt{cout} would not define a “<<” operator with an argument of type MyComplex.

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

class MyComplex {
  private:
    double re, im;
  public:
    MyComplex(double r, double i) : re(r), im(i) { };
    MyComplex operator+(MyComplex) const;
    friend ostream& operator<<(ostream&, const MyComplex&);
  };

MyComplex MyComplex::operator+(const MyComplex arg) const {
  double d1 = re + arg.re;
  double d2 = im + arg.im;
  return MyComplex(d1, d2);
}

ostream& operator<<(ostream& os, const MyComplex& c) {
  os << "(" << c.re << "," << c.im << ")" << endl;
  return os;
}

int main( ) {
  MyComplex u(3,4);
  MyComplex v(2,9);

  cout << u + v << endl;
  return 0;
}
```
Q12 (6 pts): What is printed in the program below?

public class C1 {
    public static int i = 0;
    public C1() {i++;}
}

public class Main {
    public static void main(String args[]) {
        C1 x1 = new C1();
        C1 x2 = new C1();
        System.out.println(x1.i + " "+ x2.i);
    }
}

2 2
Q13 (8 pts):

1. Wilhelma has a set of data consists of key and data pairs, each key is unique, and the data is an object of type D. There are 1000 members of the set, and there are key values from 0…999. What would be a good container to hold data to allow fast insertion and lookup? A Vector or an Array would be good.

2. Ralph has a similar set of data, except his has 1000 elements that have values from 0…1,000,000. What container should he use? Because the keys are very sparse, a Vector or an Array would waste a lot of space, and a List would have relatively slow lookups. A Map would be ideal. Given that there are many implementations and variations on these, talk to me if you think I took too many points off from you.

Q14 (6 pts): In the program below,

1. circle the function that is called by “-second”.

2. circle the declaration of the object that would be pointed to by this in the operator- function.

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

// private is optional since attributes default to private.
class MyComplex {
private:
    double re, im;
public:
    MyComplex(double r, double i);
    MyComplex operator-( );

    friend ostream& operator<< (ostream&, const MyComplex&);
};

MyComplex::MyComplex(double r, double i) : re(r), im(i) { }

MyComplex MyComplex::operator-( ) {
    double d1 = -re;
    double d2 = -im;
    return MyComplex(d1, d2);
}

ostream& operator<< (ostream& os, const MyComplex& c) {
    os << "(" << c.re << ", " << c.im << ")" << endl;
    return os;
}

int main( ) {
    MyComplex first(3, 4);
    MyComplex second(2, 9);

    cout << -second << endl; //

    return 0;
}
```

This was easier than I intended, but that might not be such a bad thing.
**Q15 (10 pts):** Use the following program for the following three questions

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

class X {
    int x;
public:
    X( ) { }  
    X(int xx) : x(xx) { }
    virtual void print( ) { cout << x << endl; }
};
class Y : virtual public X {
public:
    Y(int xx) : X(52) { }
};
class W : virtual public X {
public:
    W(int xx) : X(54) { }
};
class Z : public W, public Y {
public:
    Z(int xx) : W(54), Y(52), X(55) { }
};

int main( ) {
    Z* z = new Z(55);
    z->print( );
}
```

1. What is printed? 55

2. Draw the object layout showing the relationship of the Y, W, and Z in memory.

3. What is the purpose of the text “virtual public X” in the declarations of Class Y and class W? **This ensures that only one X base class is created.**

4. If “virtual public X” is changed to “public X” in the declarations of Class Y and class W, what will the object layout look like?