ECE 461 Fall 2011, First Exam

DO NOT START WORKING ON THIS UNTIL TOLD TO DO SO. LEAVE IT ON THE DESK.

You have until 8:20 to take this exam.

Your exam should have 12 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:

Question 1 (5.5 pts) Consider the program:

```
class X {
   int m;
   int n;
public:
   X(int mn, int nn) {m = mm; n = nn;}
}

void print (X* p) {
   cout << p->m << " " << p->n << endl;
}
```

The compiler complains that `m` and `n` cannot be accessed because they are private. Circle the answer that solves the problem and maintains the highest degree of encapsulation (that is, that allows the smallest amount of code outside the `X` class to directly access `m` and `n`..

a. Add the line `friend void print(X*);` to the declaration of the class `X`.

b. Make `m` and `n` private.

c. Make `m` and `n` public.

d. Make `m` and `n` protected.

e. Add the line `public void print( );` to the declaration of the class `X`, and make the definition of the function `print` be:

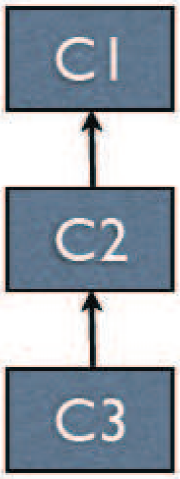
```
void X::print ( ) {
   cout << m << " " << n << endl;
}
```

Question 2 (5.5 pts) Fill in the underlined areas in the Java and C++ programs outlined below to reflect the inheritance shown in the drawing to the right. Leave it blank if that is what is needed.

```
A Java program outline:
class C1  __________________________________{
   . . .
}


class C2  __________________________________{
   . . .
}

class C3  __________________________________{
   . . .
}
```



```
A C++ program outline:
class C1  __________________________________{
   . . .
}


class C2  __________________________________{
   . . .
}

class C3  __________________________________{
   . . .
}
```

Question 3 (5.5 pts)

```
class B {
   int i = 0;
   public B(int i) {this.i = i;}
   public void incr(int j) {i += j;}
   public int getI( ) {return i;}
   public void setI(int j) {i = j;}
}
class D extends B {
   public D(int i) {super(i);}
   public void mul(int j) {
      int m = super.getI( ) * j;
      super.setI(m);
   }
   public int getVal( ) {return super.getI( );}
}
class T {
   public static void main(String [] str) {
      D d = new D(5);
      d.incr(5);
      d.mul(5);
      System.out.println("val is "+d.getVal( ));
   }
}
```

The outcome of this program is:

a. `val is 50`.

b. An error, because there is no function `incr` defined in class D.

c. An error, because there is no function `mul` defined in class B.

Question 4 (5.5 pts)
In C++, to exhibit polymorphism it is necessary to

a. Declare all methods that will exhibit polymorphic behavior as virtual.

b. Access the methods by a pointer or reference.

c. Declare the class as `abstract`.

e. Declare the class as `polymorphic`.

f. **a** and **b**.

g. **a** and **c**.

h. **a** and **e**.

Question 5 (5.5 pts)

```
cat B.java D.java T.java
class B {
   int i = 0;
   public B(int i) {this.i = i;}
   public void incr(int j) {i += j;}
   public int getVal( ) {return i;}
   public void setVal(int j) {i = j;}
}
class D extends B {
   public D(int i) {super(i);}
   public void incr(int j) {
      int m = super.getVal( ) * 2*j;
      super.setVal(m);
   }
   public int getVal( ) {return super.getVal( );}
}
class T {
   public static void main(String [] str) {
      D d = new D(5);
      B b = (B) new D(5);
      d.incr(5);
      b.incr(5);
      System.out.println("values are "+d.getVal( )+", "+b.getVal( ));
   }
}
```

What is printed?

a. `values are 10, 10`

b. `values are 50, 50`

c. `values are 10, 50`

d. `values are 50, 10`

Question 6 (5.5 pts)

The program is essentially the same as the one in Question 5, but has been simplified as indicated in the comments below.

```
class B {
   int i = 0;
   public B(int i) {this.i = i;}
   public void incr(int j) {i += j;}
   public int getVal( ) {return i;}
   public void setVal(int j) {i = j;}
}
class D extends B {
   public D(int i) {super(i);}
   public void incr(int j) { // THIS METHOD WAS CHANGED FROM QUESTION 5.
      super.incr(j);
   }
   public int getVal( ) {return super.getVal( );}
}
class T {
   public static void main(String [] str) {
      // DECLARATIONS AND CODE RELATED TO b WERE DELETED FROM QUESTION 5.
      D d = new D(5);
      d.incr(5);
      System.out.println("value is "+d.getVal( ));
   }
}
```

What is printed?

a. `value is 10`

b. `value is 50`

c. error because `super` is not a defined method in class `D`.

Question 7 (5.5 pts) In a C++ program class D inherits from class B. Both B and D implement a member function (i.e. the function is not static) `foo(int)`. The code, including D's `foo` function, looks like:

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

class B {
private:
   int i;
public:
   virtual int foo(int);
};

class D : B {
public:
   virtual int foo(int);
};

int B::foo(int j) {
   return 2*j;
}

int D::foo(int j) {
   // Would like to call B's foo here, passing j as an argument to B's foo,
   // and assigning the return value from B's foo to j.
   return 2*j;
}
```

What should be placed in the line "// would like to call B's foo here ... " to call B's foo, passing `j` as an argument and assigning the return value from B's foo to j?

Question 8 (5.5 pts)
Make the following Java class abstract.

```
class B {
   int i = 0;
   public void incr(int j);
}
```

Question 9 (5.5 pts)
Change the declarations below as needed to make the C++ class `B` abstract.

```
class B {
   int i = 0;
   public void incr(int j);
}
```

Question 10 (5.5 pts)
A programmer reads in an ordered list of doubles of the form (key, str), where *key* is an integer key and *str* is a string, and where all keys from l to u are present. The program then reads in an *unordered* list of integers whose values are greater than or equal to l, and less than or equal to u. The program will then print out the integer and the corresponding *str*.

What is a good container to use for this problem and why?

Question 11 (5.5 pts)
A programmer reads in a list of strings, some of which might be duplicates. The programmer then wants to print the strings without duplicates. What is a good container to use for this and why?

Question 12 (5.5 pts)
A programmer has created an abstract class A from which many other classes are derived. Change the C++ code below to create a vector which will hold any class that is derived from A (directly or indirectly) and no other classes.

```
vector v(5);
```

Question 13 (5.5 pts)
Let *listI* be a list of integers. Show a line of code that will place a 4 somewhere in the list.

Question 14 (5.5 pts)
Consider the following program:

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

class B {
public:
   B( );
};

class C : public B {
public:
   C();
};

B::B( ) {
   cout << "B ";
}

C::C( ) {
   cout << "C";
}

int main( ) {
   C* c = new C( );
   cout << endl;
}
```

What is printed?

a. `C`

b. `B`

c. `B C`

d. `C B`

Question 15 (5.5 pts)

Show the code requested in the comment in `foo`.

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

class B {
private:
   int i;
public:
   virtual int foo(int);
   // add a declaration here that allows global function bar to access i.
};

int B::foo(int j) {
   return 2*j;
}

int bar(B b) {
   b.i = 5011;
   return b.i;
}
```

Question 16 (5.5 pts)

Consider the program below:

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

class B {
private:
   int i;
public:
   virtual int foo(int);
};

int B::foo(int j) {
   return 2*j;
}

int main( ) {
   B* b = new B( );
}
```

Pick the correct answer:

a. There will be a compile time error because no constructor for `B` is given.

b. There will be a run-time error when the constructor for `B` is called in m`main`.

c. The compiler will create a default zero-arg constructor for `B`

d. The semantics for the call to the `B` constructor are undefined and depend on the particular compiler used.

Question 17 (5.5 pts)

Consider the code:

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

class B {
public:
   int i;
   B( );
   virtual int foo(int);
};

class D : B {
private:
public:
   D(int);
   virtual int foo(int);
};

B::B( ) {
   i = -1;
}

int B::foo(int j) {
   return 2*j;
}

int D::foo(int j) {
   return 2*j;
}

D::D(int j) {
   B::i = j;
}

int main( ) {
   D* d = new D(5 );
}
```

Pick the correct answer as to what happens when the constructor call `D(5)` in `main` occurs.

a. The constructor `D::D(int)` is called, which calls the zero-arg constructor for `B` to create the `B` object that is part of the constructed `D` object.

b. The constructor `D::D(int)` is called, and no other constructors are called.

c. The constructor `D::D(int)` is called, and the C++ compiler would like to invoke a `B::B(int)` constructor, but since none exists an error is given.