ECE 30862 Fall 2015, Second Exam

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

THE LAST PAGE IS THE ANSWER SHEET. TEAR IT OFF AND PUT ALL ANSWERS THERE. TURN IN BOTH PARTS OF THE TEST WHEN FINISHED.

You have until 12:20PM to take this exam. The total number of points should be 100, and each question is worth 2.5 points. After taking the test turn in both the test and the answer sheet.

Your exam should have 9 (nine) pages total (including this cover page and theanswer sheet, one almost entire blank page and the answer sheet). As soon as the test begins, check that your exam is complete and *let one of the proctors know immediately if it does not.*

This exam is open book, open notes, but absolutely 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 toknow answer the question, and answer it under those assumptions. *Check the front board occasionally for corrections.*

I have neither given nor received help during this exam from any other person or electronic source, and I understand that if I have I will be guilty of cheating and will fail the exam and perhaps the course.

Name (must be signed to be graded):

Name (printed, worth 2 pt):

Last four digits of your ID:

For each statement below which has a question number (e.g., **Q7**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put "E" on the answer sheet.

```
#include <string>
#include <iostream>
class B {
public:
   B( ) { }
   void m1( ) {std::cout << "B::m1" << std::endl;}
   virtual void m2( ) {
      std::cout << "B::m2" << std::endl;
   }
private:
   virtual void m3( ) {
     std::cout << "B::m3" << std::endl;
   }
   virtual void m4( ) {
      std::cout << "B::m4" << std::endl;
   }
};

class D1 : public B {
public:
   D1( ) { }
   ~D1( ) { };
   void m1( ) {
      std::cout << "D1::m1" << std::endl;
   }
   void m2( ) {
      std::cout << "D1::m2" << std::endl;
   }
private:
   virtual void m3( ) {
      std::cout << "D1::m3" << std::endl;
   }
   virtual void m4( ) {
      std::cout << "D1::m4" << std::endl;
   }
};

class D2 : public D1 {
public:
   D2( ) { };
   ~D2( ) { };
   void m1( ) {
      std::cout << "D2::m1" << std::endl;
      }
   virtual void m2( ) {
      std::cout << "D2::m2" << std::endl;
      }
private:
   void m3( ) {
      std::cout << "D2::m3" << std::endl;
      }
   virtual void m4( ) {
      std::cout << "D2::m4" << std::endl;
      }
};
```

```
int main(int argc, char * argv[ ]) {
   // reminder: D1 d1; uses 0 arg const. for D1
   B b; B* bP = &b; B& bR = b;
   D1 d1; D1* d1P = &d1; D1& d1R = d1;
   D2 d2; D2* d2P = &d2; D2& d2R = d2;

   d1.m1( ); // Q1
   d1R.m4( ); // Q2

   bP = d1P; bR = d1R; b = d1;
   b.m1( ); // Q3
   b.m2( ); // Q4
   bP->m1( ); // Q5
   bP->m2( ); // Q6
   bR.m2( ); // Q7

   B& bR2 = d1;
   bR2.m2( ); // Q8

   d1P = d2P;
   B& bR3 = d2R;
   d1P->m1( ); // Q9
   d1P->m2( ); // Q10
   bR3.m1( ); // Q11
   bR3.m2( ); // Q12
}
```

For each statement below which has a question number (e.g., **Q13**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put "E" on the answer sheet.

```
#include <string>
#include <iostream>

class B {
public:
   B(int i) { }
   virtual ~B() { }
   virtual void m1(double d) {
      std::cout << "B::m1(f)" << std::endl;
   }
   virtual void m1(int i) {
      std::cout << "B::m1(i)" << std::endl;
   }
   virtual void m2( ) {
      std::cout << "B::m2( )" << std::endl;
   }
   virtual void m4( ) {
      std::cout << "B::m4( )" << std::endl;
   }

};

class D : public B {
public:
   D( ) : B(3) { }
   D(int i) : B(i) { }
   virtual ~D() { }
   virtual void m1(double d) {
      std::cout << "D::m1(d)" << std::endl;
   }
};

class D2 : private D {
public:
   D2(int i) : D(i) { }
   virtual ~D2() { }
   virtual void m3( ) {
      m1(2.0);
   }
};

int main(int argc, char * argv[ ]) {
   B b(4); B* bP = &b;
   D d(3); D* dP = &d;
   D2 d2(3); D2* d2P = &d2;

   b.m1(3.0); // Q13
   d.m1(3); // Q14
   d.m1(3.0); // 516
   d.m2( ); // Q16
   dP->m1(3); // Q17
   dP->m1(3.0); // Q18
   d2P->m1(3); // Q19
   d2P->m3( ); // Q20
}
```

Q21. (Unrelated to the program above.) Programmer Bob would like to create a data structure that will allow him to quickly retrieve the last item added (a Last In First Out, or LIFO queue), yet occassionally be able to efficiently add items that will be at the end of the current list of items to be taken out (like a First In First Out, or FIFO, queue).What container can Bob use to do this?

a. a List

b. a Hashmap or Hashtable

c. a Tree

d. none of the above.

For each statement below which has a question number (e.g., **Q22**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put "E" on the answer sheet.

```
#include <string>
#include <iostream>

class L {
private:
   int feet;
   int inches;

   static L& adjust(L& m) {
      /* not important to the problem */
   }

public:
   L(int f, int i) : inches(i), feet(f) { }
   virtual ~L() { }
   L& operator+ (const L& m) const {
      L* res = new L(feet + m.feet, inches + m.inches);
      *res = adjust(*res);
      return *res;
   }

   L& operator- (const L& m) const {
      L* res = new L(feet - m.feet, inches - m.inches);
      *res = adjust(*res);
      return *res;
   }

   L& operator! ( ) const {
      L* res = new L((feet < 0) ?
         -feet : feet, (inches < 0) ? -inches : inches);
      return *res;
   }

   friend L& operator- (const L&m);
   friend std::ostream& operator<< (std::ostream& os, const L& m);

};

L& operator- (const L&m) {
   L* t = new L(-m.feet, -m.inches);
   return *t;
}

std::ostream& operator<< (std::ostream& os, const L& m) {
   os << "(" << m.feet << ", " << m.inches << ")";
   os << std::endl;
   return os;
}

int main(int argc, char * argv[ ]) {
   L m1(-4,-7);
   std::cout << m1 << std::endl; // Q22

   L m2(-1,-8);
   std::cout << m2 << std::endl; // Q23
   std::cout << !m2 << std::endl; // Q24

   L m4 = m1-m2;
   std::cout << m4 << std::endl; // Q25
}
```

Q26: Answer true or false: Could the `operator<<` function be a member function of the L class?

Q27: In the line "**L m4 = m1-m2**", is **m1**, **m2**, or neither passed in as the *this* pointer?

Q28 : In the line `L& operator- (const L& m) const {` does the bold **const** mean (pick the best):

a. The parameter `L&` will not be changed in the function.

b. the function will only change parameters, not other global variables.

c. the function will not change what is pointed to by the `this` pointer.

Q29 In the line `L& operator- (const L& m) const {` does the bold **const** mean (pick the best using the same choices as in Q28):

For each statement below which has a question number (e.g., **Q30**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put "E" on the answer sheet.

```
#include <string>
#include <iostream>

class L {
public:
   int feet;
   int inches;

   L(int f, int i) : inches(i), feet(2*inches) { }
   L(int f, int i, char c) : inches(i), feet(f) { }
   L(const L& m) : inches(4), feet(4) { }
   virtual ~L() { }

};

L& operator- (const L m) {
   L* t = new L(m.feet, m.inches, 'f');
   return *t;
}

int main(int argc, char * argv[ ]) {
   L m1(5,5);
   std::cout << "m1(" << m1.feet << ", " << m1.inches << ")" << std::endl; // Q30

   L m2 = -m1;
   std::cout << "m2(" << m2.feet << ", " << m2.inches << ")" << std::endl; // Q31

}
```

```
#include <string>
#include <iostream>

class L {
public:
   int data;

   L() : data(0) { }
   L(int i) : data(i) { }
   virtual ~L() { }

   static void swap(L lO1, L lO2, L& lR1, L& lR2, L* lP1, L* lP2) {
      L tmpO = lO1;
      L& tmpR = lR1;

      tmpO = lO1; lO1 = lO2; lO2 = tmpO;
      tmpR = lR1; lR1 = lR2; lR2 = tmpR;
      tmpO.data = lP1->data; lP1->data = lP2->data; lP2->data = tmpO.data;
      lP1 = lP2;
   }
   friend std::ostream& operator<< (std::ostream& os, const L& m);
};

int main(int argc, char * argv[ ]) {
   L lO1(1); L lO2(2);
   L* t = new L(1); L lR1 = *t; t = new L(2); L lR2 = *t;
   L* lP1 = new L(1); L* lP2 = new L(2);
   std::cout << "lO1: " << lO1.data << ", lO2: " << lO2.data << std::endl; // lOa
   std::cout << "lR1: " << lR1.data << ", lR2: " << lR2.data << std::endl; // lRa
   std::cout << "lP1->data: " << lP1->data << ", lP2->data2: " << lP2->data << std::endl; // lXa
   std::cout << "lP1: " << lP1 << ", lP2: " << lP2 << std::endl; // lPa

   L::swap(lO1, lO2, lR1, lR2, lP1, lP2);
   std::cout << "lO1: " << lO1.data << ", lO2: " << lO2.data << std::endl; // lOb
   std::cout << "lR1: " << lR1.data << ", lR2: " << lR2.data << std::endl; // lRb
   std::cout << "lP1->data: " << lP1->data << ", lP2->data2: " << lP2->data << std::endl; // lXb
   std::cout << "lP1: " << lP1 << ", lP2: " << lP2 << std::endl; // lPb
}
}
```

Q32. Answer true or false. Do lines **lOa** and **lOb** print the same thing?

Q33. Answer true or false. Do lines **lRa** and **lRb** print the same thing?

Q34. Answer true or false. Do lines **lXa** and **lXb** print the same thing?

Q35 Answer true or false. Do lines **lPa** and **lPb** print the same thing?

For each statement below which has a question number (e.g., **45**), write the output that results from executing the statement on the answer sheet. If the line would produce an error at either compile or run time put "E" on the answer sheet.

```
public class B implements Cloneable {

   public int i;
   public int j;

   public B( ) {i=4; j=0;}

   public B(int ii, int jj) {i = ii; j = jj;}

   public Object clone( ) {
      B newObj = new B(j, i);
      return newObj;
   }

   public String toString( ) {
      return " "+i+", "+j;
   }
}
public class D implements Cloneable {

   public int i;
   public int j;

   public D( ) {i=4; j=0;}

   public String toString( ) {
      return " "+i+", "+j;
   }
}
class Main {

   public static void main(String args[]) {

      B b1 = new B( );
      D d1 = new D( );

      B b2 = (B) b1.clone( );
      D d2 = (D) d1.clone( ); // Q36

      System.out.println("b1: "+b1); // Q37
      System.out.println("b2: "+b2); // Q38
      System.out.println("d1: "+d1);
      System.out.println("d2: "+d2);
   }
}
```

This page intentionally left almost blank

ECE 30862 Fall 2015 Second Exam Answer Sheet

All answers should be on this sheet. Both this sheet and your test must be signed and turned in. You may detach this sheet from the rest of the test to make it easier to write your answers on it. Each question is worth 4 points.

I promise that I have neither Given nor received disallowed aid on this test.

Putting your name on this is worth 2 pt.

Name (Printed):

Name (Signed):

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.

16.

17.

18.

19.

20.

21.

22.

23.

24.

25.

26.

27.

28.

29.

30.

31.

32.

33.

34.

35.

36.

37.

38.