ECE 30862 Fall 2018, Test 2

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 7:30 to take this exam. The total number of points should be 106. Each of the 53 questions is worth 2 points. After taking the test turn in both the test and the answer sheet. You should remove the answer sheet from the rest of the test when taking it.

Your exam should have 10 (ten) pages total (including this cover page, one almost entirely blank page, and the answer sheet). As soon as the test begins, check that your exam is complete and *let a proctor know immediately if it is 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 to answer the question, and answer it under those assumptions. *Check the front board occasionally for corrections.*

Programs may be given without “#include” statements, and without “std::” for brevity, and to allow them to fit on a page. Assume these are present where needed.

For questions that are in comments at the ends of lines, e.g., “foo( ); // Q23”, you should:

- Answer what is printed if something is printed;
- if nothing is printed and the statement is legal at both compile time and at runtime answer “Ok”;
- and if nothing is printed by the statement gives either a compile time or runtime error, answer “Error”, “Err” or something similar. *If the statement is an error, answer questions on following lines in the program as if the statement did not exist in the program.*

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 (signed):

Name (printed):

Last four digits of your ID:

The code below is used for **C++** questions 1 - 6.

```
// Err.h
class Err {
public:
   Err(float, float);
   virtual ~Err( );
protected:
   float num, denom;
};

// Err.cpp
Err::Err(float n, float d) {num=n; denom = d;}
Err::~Err( ) { }

// ErrD.h
class ErrD : public Err {
public:
   ErrD(float, float);
   virtual ~ErrD( );
};

// ErrD.cpp
ErrD::ErrD(float n, float d) : Err(n,d) { }
ErrD::~ErrD( ) { }

// NAR.h
class NaR {
public:
   NaR(int);
   virtual ~NaR( );
   virtual NaR& sub(const NaR&);
public:
   int val;
};

// NaR.cpp
NaR::NaR(int n) {val = n;}
NaR::~NaR( ) { }
NaR& NaR::sub(const NaR& r) {
   if ((r.val < 0) || (val < 0))
      throw Err(val, r.val);
   if (r.val <= val) {
      NaR *p = new NaR(val - r.val);
      return *p;
   }
   throw ErrD(val, r.val);
}

// main.cpp
NaR& sub(NaR& p1, NaR& p2) {
   try {
      p1.sub(p2);
   } catch (ErrD e) {std::cout << "-1" << std::endl;}
     catch (Err e) {std::cout << "-2" << std::endl;}
}
```

```
int main (int argc, char *argv[]) {
   NaR nm(-1);
   NaR n0(0);
   NaR n1(1);

   try { // Q1 what is printed by the Try Catch?
      NaR& nt1 = nm.sub(n0);
      NaR& nt2 = n0.sub(n1);
   } catch (Err e) {
        std::cout << "1" << std::endl;
   } catch (ErrD e) {std::cout << "2" << std::endl;}

   try {// Q2 what is printed by the Try Catch?
      NaR& nt1 = n1.sub(n0);
      NaR& nt2 = n0.sub(n1);
   } catch (Err e) {
       std::cout << "3" << std::endl;
   } catch (ErrD e) {std::cout << "4" << std::endl;}

   try { // Q3 what is printed by the Try Catch
      NaR& nt1 = nm.sub(n0);
      NaR& nt2 = n0.sub(n1);
   } catch (ErrD e) {
       std::cout << "5 << std::endl";
   } catch (Err e) {std::cout << "6" << std::endl;}

   try { // Q4 what is printed by the Try Catch
      NaR& nt1 = n1.sub(n0);
      NaR& nt2 = n0.sub(n1);
   } catch (ErrD e) {
       std::cout << "7" << std::endl;
   } catch (Err e) { std::cout << "8" << std::endl;}

   try { // Q5 what is printed by the Try Catch
      NaR& nt1 = sub(n1,n0);
      NaR& nt2 = sub(n0,n1);
   } catch (ErrD e) {
       std::cout << "9" << std::endl;
   } catch (Err e) { std::cout << "10" << std::endl;}

   try { // Q6 what is printed by the Try Catch
      NaR& nt1 = sub(n0,nm);
      NaR& nt2 = sub(nm,n0);
   } catch (ErrD e) {
       std::cout << "10" << std::endl;
   } catch (Err e) { std::cout << "11" << std::endl;}
}
```

The code below is used for **C++** questions 7 - 14.

```
// B.h
class B {
public:
   B( );
   B(int);
   B(B&);
   virtual ~B( );
   virtual B& operator= (const B& b);
   int v;
};

B::B( ) {v=0;}

B::B(int i) {v=1;}

B::B(B& b) {
   v = 4;
   b.v = -b.v;
}

B::~B( ) { }

B& B::operator= (const B& b) {
   B* bP = new B(-b.v);
   return *bP;
}

// main.cpp
void xchange(B bx, B by) {
   B tmp;
   tmp = bx;
   bx = by;
   by = tmp;
}

void xchangeR(B& bx, B& by) {
   B t;
   B& tmp = t;
   tmp = bx;
   bx = by;
   by = tmp;
}

void xchange(B* bx, B* by) {
   B* tmp;
   tmp = bx;
   bx = by;
   by = tmp;
   by->v = -100;
}
```

```
// main.cpp
int main (int argc, char *argv[]) {

   B t1(2);
   B t2(3);
   B& bR1 = t1;
   B& bR2 = t2;
   B b1;
   B b2(2);
   B* bP1 = new B( );
   B* bP2 = new B(2);

   std::cout << bR1.v << " " << b1.v << " " << b2.v << " "
             << bP1->v << std::endl; // Q7

   b1 = b2;
   bR1 = bR2;
   bP1 = bP2;

   std::cout << bR1.v << " " << t1.v << std::endl; // Q8
   std::cout << b1.v << " " << b2.v << " "
             << bP1->v << std::endl; // Q9

   bR1.v = 7;
   std::cout << bR1.v << " " << bR2.v << std::endl; // Q10

   xchange(b1, b2);
   std::cout << b1.v << " " << b2.v << std::endl; // Q11

   xchangeR(b1, b2);
   std::cout << b1.v << " " << b2.v << std::endl; // Q12

   xchangeR(bR1, bR2);
   std::cout << bR1.v << " " << bR2.v << std::endl; // Q13

   xchange(bP1, bP2);
   std::cout << bP1->v << " " << bP2->v << std::endl; // Q14
}
```

The code below is used for **C++** questions 15 - 19.

```
// Nat.h
class Nat {
public:
   Nat(int);
   virtual ~Nat( );
   virtual Nat& operator*(const Nat&) const; // L1
   virtual Nat& operator/(const Nat&) const;
   virtual void abs( );
   friend Nat& operator+(const Nat&, const Nat&);
   friend Nat& operator-(const Nat&, const Nat&);
   friend std::ostream& operator<<(std::ostream&, const Nat&);

   int val;
};

// Nat.cpp
Nat::Nat(int i) { val = i; abs( );}

Nat::~Nat( ) { }

Nat& Nat::operator*(const Nat& n) const {
   Nat* nP = new Nat(val * n.val);
   return *nP;
}

Nat& Nat::operator/(const Nat& n) const {
   Nat* nP = new Nat(n.val / val);
   return *nP;
}

void Nat::abs( ) {if (val < 0) val = -val;}

Nat& operator+(const Nat& n1, const Nat& n2) {
   Nat* nP = new Nat(n1.val + n2.val);
   return *nP;
}


Nat& operator-(const Nat& n1, const Nat& n2) {
   Nat* nP = new Nat(n2.val - n1.val);
   nP->abs( );
   return *nP;
}

std::ostream& operator<<(std::ostream& os, const Nat& n) {
   os << " " << n.val << " ";
}
```

```
// main.cpp
int main (int argc, char *argv[]) {
   Nat n1(3);
   Nat n2(6);
   Nat n3(9);

   n3 = n1+n2;
   std::cout << n3.val << std::endl; // Q15

   n3 = n1-n2;
   std::cout << n3.val << std::endl; // Q16

   n3 = n1*n2;
   std::cout << n3.val << std::endl; // Q17

   n3 = n1/n2;
   std::cout << n3.val << std::endl; // Q18

   std::cout << n1 << std::endl; // Q19
}
```

The code below is used for **C++** questions 20 - 30.

```
// B.h
class B {
public:
   B( );
   B(int);
   virtual ~B( );
   virtual void f1( );
   void f3( );
   virtual void f4(B&);
private:
   virtual void f2( );
};

// B.cpp
B::B( ) { }
B::B(int) { }
B::~B( ) { }

void B::f1( ) {std::cout << "B::f1" << std::endl;}
void B::f3( ) {std::cout << "B::f3" << std::endl;}
void B::f4(B&) {std::cout << "B::f4" << std::endl;}
void B::f2( ) {std::cout << "B::f2" << std::endl;}

// C.h
class C : public B {
public:
   C(int);
   virtual ~C( );
   virtual void f2( );
   virtual void f5( );
   virtual void f3( );
};

// C.cpp
C::C(int i) { }
C::~C( ) { }
void C::f2( ) {std::cout << "C::f2" << std::endl;}
void C::f5( ) {std::cout << "C::f6" << std::endl;}
void C::f3( ) {std::cout << "C::f3" << std::endl;}
```

```
// main.cpp
int main (int argc, char *argv[]) {

   B b1(1);
   C c1(2);
   B& bd = c1;
   B* bP = &c1;
   C* cP = &c1;
   C* dQ = new C( ); // Q20

   bP->f1( ); // Q21
   bP->f2( ); // Q22
   bP->f3( ); // Q23
   cP->f2( ); // Q24
   cP->f3( ); // Q25
   bP->f4(c1); // Q26
   bP->f5( ); // Q27

   bd.f1( ); // Q28
   b1 = c1; // Q29
   b1.f1( ); // Q30
}
```

The code below is used for **Java** questions 31 - 44.

```
class B {

   public static void f1( ) {
      System.out.println("B::f1");
   }
   public B( ) {
      System.out.println("B");
   }
   public void f2( ) {
      System.out.println("B::f2");
   }
   public void f3(B b) {
      System.out.println("B::f3");
      b.f4( );
   }
   private void f4( ) {
      System.out.println("B::f4");
   }
   public static int i = 0;
}

class D extends B {

   public static void f1( ) {
      System.out.println("D::f1");
   }

   public D( ) {
      super( );
      System.out.println("D");
   }

   public void f2( ) {
      System.out.println("D::f2");
   }

   public void f3(D d) {
      System.out.println("D::f3");
      d.f4( );
   }

   public void f5( ) {
      System.out.println("D::f5");
   }

   private void f4( ) {
      System.out.println("D::f4");
   }
}
```

```
class Main {

   public static void main(String args[])
      throws Exception {
      D d = new D( ); // Q31
      B b = d;

      b.i = b.i + 2;

      b.f1( ); // Q32
      b.f2( ); // Q33
      b.f3(b); // Q34
      b.f4( ); // Q35
      b.f5( ); // Q36

      d.f1( ); // Q37
      d.f2( ); // Q38
      d.f3(d); // Q39
      d.f4( ); // Q40
      d.f5( ); // Q41

      B b1 = new B( ); // Q42
      System.out.println(b.i + " " + b1.i); // Q43
      System.out.println(B.i); // Q44
   }
}
```

The code below is used for **Java** questions 45 - 49.

```
interface I1 {
   int i = 0;
   int j = 1;

   void f1( );
   void f2( );
}

interface I2 {
   int i = 3;
   int j = 4;

   void f1( );
   void f2( );
}

class D implements I1, I2 {

   public D( ) { }

   public void f1( ) {System.out.println("D::f1");}

   public void f3( ) {System.out.println("D::f3");}
}
```

```
class E implements I1, I2 {

   public E( ) { }

   public void f1( ) {System.out.println("E::f1");}

   public void f2( ) {System.out.println("E::f2");}

   public void f3( ) {System.out.println("E::f3");}

}
class Main {

   public static void main(String args[])
      throws Exception {

      E e = new E( ); // Q45

      int i = D.j; // Q46

      e.f1( ); // Q47
      e.f3( ); // Q48
   }
}
```

Q49: What is the most correct statement about class D?

1. This is a legal class. Although *void f2( )* is not implmented, that is ok as long as *void f2( )* is never called on a D object.
2. This is an illegal class because *void f2( )* is not implemented.
3. This is a legal class, but D objects cannot be created. However, if a class X extends D, and class X defines *void f2( )*, X objects can be created.

The code below is used for **Java** questions 50 - 53.

```
class Main {

   public static void f1(float f, double d) {
      System.out.println("f1(f,d)");
   }

   public static void f1(float f, int i) {
      System.out.println("f1(f,i)");
   }

   public static void f1(double d, short l) {
      System.out.println("f1(d,l)");
   }


   public static void main(String args[]) throws Exception {
      float f = (float) 1.0;
      double d = 2.0;
      int i = 1;
      long l = 2;
      short s = 0;

      f1(d, f); // Q50
      f1(f, d); // Q51
      f1(f, s); // Q52
      f1(i, i); // Q53
   }
}
```

This page intentionally left almost blank

Fall 2018 Second Exam Answer Sheet – print your name on this sheet.

1.	**24.**	**47.**
2.	**25.**	**48.**
3.	**26.**	**49.**
4.	**27.**	**50.**
5.	**28.**	**51.**
6.	**29.**	**52.**
7.	**30.**	**53.**
8.	**31.**	
9.	**32.**	
10.	**33.**	
11.	**34.**	
12.	**35.**	
13.	**36.**	
14.	**37.**	
15.	**38.**	
16.	**39.**	
17.	**40.**	
18.	**41.**	
19.	**42.**	
20.	**43.**	
21.	**44.**	
22.	**45.**	
23.	**46.**	

Fall 2018 Second Exam Key Sheet – print your name on this sheet.

1.	1
2.	3
3.	6
4.	7
5.	-1
6.	-2 -2
7.	1 0 1 0
8.	1 1
9.	0 1 1
10.	7 1
11.	0 -1
12.	0 -1
13.	7 1
14.	-100 -100
15.	9
16.	3
17.	18
18.	2
19.	3
20.	Err
21.	B::f1
22.	Err
23.	B::f3
24.	C::f2
25.	C::f3
26.	B::f4
27.	Err
28.	B::f1
29.	Ok
30.	B::f1
31.	B D
32.	B::f1
33.	D::f2
34.	B::f3 B::f4
35.	Err
36.	Err
37.	D::F1
38.	D::f2
39.	D::f3 D::f4
40.	Err
41.	D::f5
42.	B
43.	2 2
44.	2
45.	Ok
46.	Err
47.	E::f1
48.	E::f3
49.	2
50.	Err
51.	f1(f,d)
52.	Err
53.	f1(f,i)

ECE 30862 Fall 2017, 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 9:00PM to take this exam. There are 50 questions and each is worth two points. After taking the test, turn in both the test and the answer sheet.

Your exam should have this sheet, 7 pages with 50 questions, and the answer sheet. As soon as the test begins, check that your exam is complete and *let Prof. Midkiff 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 to answer the question, and answer it under those assumptions.

If a statement is illegal, assume it is not executed when answering other questions in the test.

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, at the instructor's discretion.

Name (must be signed to be graded):

Name:

Last four digits of your ID:

C++Questions. For each question **Q1 - Q9** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer "Err". If the statement is legal and nothing is printed, answer "Ok". If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

A.h

class A {

```
class A {
public:
   static int count; // Q1 Assume the line
                     // at Q2 does not exist.
   static int count = 0; // Q2 Assume
                     // the line at Q1 does not
                     // exist:

   int counter;
   A( );
   static void incr( );
   virtual void print( );
};
```

A.cpp

```
int A::count = 0; // Q3
int A::counter = 0; // Q4

void A::incr( ) {
   count++; // Q5
   counter++; // Q6
}

void A::print( ) {
   std::cout << count << std::endl; // Q7
   std::cout << counter << std::endl; // Q8
}
```

E.h

class E {

```
public:
   std::string msg;
   E(std::string);
   virtual void print( );
};
```

E.cpp

```
E::E(std::string m) {
   A::count++;
   msg = m;
}

void E::print( ) {
   std::cout << "E: " << msg << " ";
   std::cout << A::count << std::endl;
}
```

main.cpp

```
void foo(int j) {
   if (j < 0) throw E("Err" );
   if (j == 0) throw 1;
}

int main( ) {
   for (int i = -1; i < 1; i++) { // A
      try {
         foo(i);
      }
      catch (E e) {e.print( );}
      catch (int i) {std::cout << i << std::endl;}
   }

   std::cout << A::count << std::endl; // Q9
}
```

Q10. What is printed during the entire execution of the loop at the statement marked with **A**?

C++ questions. For each question **Q11 - Q15** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer “Err”. If the statement is legal and nothing is printed, answer “Ok”. If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

Base.h
```
class Base {
public:
  int* x;
  Base( );
  virtual ~Base( );
};
```

Base.cpp
```
Base::Base( ) {
  std::cout << "Base" << std::endl;
  x = new int[3];
  x[0]= 0; x[1] = 1; x[2] = 2;
}

Base::~Base( ) {
  std::cout << "~Base" << std::endl;
  delete x;
}
```

A.h
```
class A : public Base {
public:
  A( );
  virtual ~A( );
};
```

A.cpp
```
A::A( ) {
  std::cout << "A" << endl;
}

A::~A( ) {
  cout << "~A" << std::endl;
};
```

main.cpp
```
int main( ) {
  A a; // Q11
  Base b; // Q12

  b = a;
  b.x[1] = -1;

  std::cout << a.x[1] << std::endl; // Q13
  std::cout << b.x[1] << std::endl; // Q14

}
```

Q15. When exiting the main routine, how many times is the array freed that is allocated when the object held in the variable "a" is constructed?

C++ questions. For each question **Q16 - Q22** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer “Err”. If the statement is legal and nothing is printed, answer “Ok”. If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

Int.h
```
class Int {
public:
  Int( );
  Int(int);
  Int(const Int&);
  virtual ~Int( );
  Int operator=(const Int&);
  Int operator+(const Int);
  Int operator-(const Int&);
  Int operator-( ); // A
  std::ostream& operator>>(std::ostream&);
  friend Int operator-(const Int&); // B
  friend std::ostream& operator<<(
       std::ostream&, const Int&);  // C
private:
  int val;
};

Int operator-(const Int&); // D

std::ostream& operator<<(std::ostream&, const
Int&);
```

Int.cpp
```
Int::Int( ) {
  val = 0;
}

Int::Int(int i) {
  val = i;
}

Int::Int(const Int& src) {
  val = src.val*src.val;
}


Int::~Int( ) { }

Int Int::operator=(const Int& i) {
  Int n = i;
  return n;
}

Int Int::operator+(const Int i) {
  Int n;
  n.val = this->val + i.val;
  return n;
}
```

Int.cpp continued
```
Int Int::operator-(const Int& i) {
  Int n;
  n.val = i.val - this->val;
  return n;
}

Int Int::operator-( ) {  // E
  Int n;
  n.val = -this->val;
  return n;
}

Int operator-(const Int& i) {  // F
  Int n;
  n.val = -i.val;
  return n;
}

std::ostream& Int::operator>>(std::ostream& os) {
  os << val;
  return os;
}

std::ostream& operator<<(std::ostream& os, const Int& i) {
  os << i.val;
  return os;
}

int main( ) {
  Int i1(1);
  Int i2(2);
  Int i3;
  std::cout << i1 << " " << i2 << " " << i3 << std::endl; // Q16

  std::cout << i1 << " " << i2 << " " << i3 << std::endl; // Q17
  std::cout << "i1: ";
  i1 >> std::cout; // Q18
  std::cout << std::endl;

  i3 = i1 + i2; // G
  std::cout << i1 << " " << i2 << " " << i3 << std::endl; // Q19

  Int i4 = i3 = i1 - i2;
  std::cout << i1 << " " << i2 << " "; // Q20
  std::cout  << i3 << " " << i4 << std::endl; // Q21

  i4 = -i3;
  std::cout << i3 << std::endl; // Q22
}
```

C++ question. The questions below refer to the program on the previous page.

Q23. Pick all that are true. For the two functions declared at A, B, C and D, and defined at E and F (answer all that are true)
(a) They both do the same thing and only one can legally be in the program.
(b) They both do the same thing and both can legally be in the program at the same time.
(d) They do different things and both can legally be in the program.
(e) They do different things and only one can legally be in the program.
(f) The function declared at A is legal but the one declared at D is not.
(g) The function declared at D is legal but the one declared at A is not.
(h) None of the above.

Q24. Could the overloaded "<<" be a member function? Answer T or F.

Q25. what does the *this* pointer point to when executing "std::cout << i1" in the line of Question Q21. Give the name of the variable pointed to.

The following two questions have nothing to do with the program on the previous page.

Q26. You need to keep records of all homework done. The last homework done should be the first visited when accessing the container. Accesses will be done linearly. Is a List or Vector preferred?

Q27. You have 1000 customers, with customer numbers from 0 to 999. Customers will be added to the end of the container. You need to access their records in constant

Java question. For each question **Q28 - Q32** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer “Err”. If the statement is legal and nothing is printed, answer “Ok”. If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

```
class B { }

class D1 extends B { }

class D2 extends D1 { }

class Main {

  void foo(int i, long l, double d) {
    System.out.println("ild");
  }

  void foo(int i, int i2, double d) {
    System.out.println("isd");
  }

  void foo(short s, int i, double d) {
    System.out.println("sid");
  }

  void foo(short s, int i) {
    System.out.println("sid");
  }

  void foo(B b, D1 d) {
    System.out.println("bd1");
  }

  void foo(D1 d1, D2 d2) {
    System.out.println("bd");
  }

public static void main(String args[]) {

    B b = new B( );
    D1 d1 = new D1( );
    D2 d2 = new D2( );
    Main m = new Main( );
    int i = 0;
    long l = 0;
    short s  = 0;
    double d  = 0.0;
    float f  = 0.0f;
    char c = '0';

    m.foo(d1, d1); // Q28
    m.foo(b, d2); // Q29
    m.foo(c, i); // Q30
    m.foo(i, s, f); // Q31
    m.foo(s, s, f); // Q32
  }
}
```

Java question. For each question **Q33 - Q45** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer “Err”. If the statement is legal and nothing is printed, answer “Ok”. If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

```
class B {

  public void f1( ) {
    System.out.println("B::f1");
    f4( );
  }

  public void f2(int i2) {System.out.println("B::f2");}

  public void f3(short i3) {System.out.println("B::f3");}

  public void callf4( ) {f4( );}

  private void f4( ) {System.out.println("B::f4");}
}

class D1 extends B {

  public void f2( ) {
    System.out.println("D1::f2");
  }

  public void f3(int f3) {
    System.out.println("D1::f3");
  }

  public void f4( ) {
    System.out.println("D1::f4");
  }
}

class D2 extends D1 {

  public void f3( ) {
    System.out.println("D2::f3");
  }

  public void f4( ) {
    System.out.println("D2::f4");
  }

  public void f5( ) {
    System.out.println("D2::f5");
  }
}
```

```
class Main {

  public static void main(String args[]) {
    B b = new D1( );
    D1 d1 = new D1( );
    D1 d1_2 = new D2( );
    D2 d2 = new D2( );
    short s = 0;
    int i = 0;

    b.f3(i); // Q33
    b.f4( ); // Q34

    d1.f1( ); // Q35
    d1.f2(1); // Q36
    d1.f3(s); // Q37
    d1.f3(i); // Q38
    d1.f4( ); // Q39

    d1_2.f3(i); // Q40
    d1_2.f3( ); // Q41
    d1_2.f4( ); // Q42
    d1_2.f5( ); // Q43

    b = d2;
    d1 = d2;
    b.callf4( ); // Q44
    d1.callf4( ); // Q45

  }
}
```

Java question. For each question **Q46 - Q50** below answer what is printed by the commented line on your answer sheet. If a runtime or compile time error, answer “Err”. If the statement is legal and nothing is printed, answer “Ok”. If the statement is illegal, execute the remainder of the program as if the illegal statement did not exist.

```
class B {

  public void f1(B b, D d) {
    b.f2( );
    d.f2( );
  }

  private void f2( ) {System.out.println("B::f2");}
}

class D extends B {

  public D(int i) {
    val = i;
  }

  public void swap(D d1, D d2) {
    D tmp = d1;
    d1 = d2;
    d2 = tmp;
    System.out.println("d1: "+d1.val+" "+d2.val);
  }

  public void swap(R r1, R r2) {
    D tmp = r1.r;
    r1.r = r2.r;
    r2.r = tmp;
    System.out.println("r1: "+r1.r.val+", r2: "+r2.r.val);
  }

  public void f2( ) {System.out.println("D::f2");}

  public int val;
}

class R {

  public R(D ref) {
    r = ref;
  }

  public D r;
}
```

```
class Main {

  public static void main(String args[]) {
    B b = new B( );
    D d1 = new D(1);
    D d2 = new D(2);

    b.f1(d1, d1);

    d1.swap(d1, d2); // Q46
    System.out.println(d1.val+" "+d2.val); // Q47

    R r1 = new R(d1);
    R r2 = new R(d2);
    d1.swap(r1, r2); // Q48
    System.out.println(d1.val+" "+d2.val); // Q49
    d1 = r1.r;
    d2 = r2.r;
    System.out.println(r1.r.val+" "+r2.r.val); // Q50
  }
}
```

ECE 30862 Fall 2017 First Exam Answer Sheet

Name (Printed):	Name (Signed):
1.	**26.**
2.	**27.**
3.	**28.**
4.	**29.**
5.	**30.**
6.	**31.**
7.	**32.**
8.	**33.**
9.	**34.**
10.	**35.**
11.	**36.**
12.	**37.**
13.	**38.**
14.	**39.**
15.	**40.**
16.	**41.**
17.	**42.**
18.	**43.**
19.	**44.**
20.	**45.**
21.	**46.**
22.	**47.**
23.	**48.**
24.	**49.**
25.	**50.**

ECE 30862 Fall 2017 Second Exam Answer Sheet

Name (Printed): Name (Signed):

1. Ok

2. Err

3. Ok

4. Err

5. Ok

6. Err

7. Ok

8. Ok

9. 1

10. E: Err 1 1

11. Base A

12. Base

13. -1

14. -1

15. 2

16. 1 2 0

17. 1 2 0

18. i1: 1

19. 1 2 0 (this− >val not assigned)

20. 1 2

21. 0 1 (this− >val not assigned)

22. 0

23. A

24. No

25. give credit regardless of the answer

26. list (other answers may be correct, talk to me.)

27. vector (other answers may be correct, talk to me.)

28. bd1

29. bd1

30. Err

31. isd

32. sid

33. Err

34. Err

35. B::f1 B::f4

36. B::f2

37. B::f3

38. D1::f3

39. D1:f4

40. D1::f3

41. Err

42. D2::f4

43. Err

44. B::f4

45. B::f4

46. d1: 2 1

47. 1 2

48. r1: 2, r2: 1

49. 1 2

50. 2 1

ECE 30862 Fall 2016, 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 7:30PM to take this exam. The total number of points should be 100. After taking the test, turn in both the test and the answer sheet.

Your exam should have this sheet, 10 pages with 50 questions, and the answer sheet. As soon as the test begins, check that your exam is complete and *let Prof. Midkiff 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 to answer the question, and answer it under those assumptions. *Check the front board occasionally for corrections.*

Every question is worth 2 points.

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 1 pt):

Last four digits of your ID:

This page intentionally left almost blank

For each statement below which has a question number (e.g., **Q7**), write “Err” if the access is illegal and “OK” if it is legal. There is not need to say what is printed.

```
class Base { // Base.h
public:
   int i, j, l;
protected:
   int k;
public:
   Base( );
   virtual ~Base( );
};

Base::Base( ) {  } // Base.cpp

Base::~Base( ) { }

class D1 : protected Base { // D1.h
public:
   int i, j;
   D1( );
   virtual ~D1( );
};

D1::D1( ) { } // D1.cpp
D1::~D1( ){ }

class D2 : public D1  { // D2.h
public:
   D2( );
   virtual ~D2( );
   void print( );
};

D2::D2( ) { } // D2.cpp
D2::~D2( ){ }
void D2::print( ) {
   cout << i; // Q1
}
```

```
#include "Base.h"
#include "D1.h"
#include "D2.h"
#include <iostream>
#include <string>
using namespace std;

int main(void) {

   Base* b = new Base( );
   Base* d1 = new D1( );
   Base* d2 = new D2( );

   cout << d1->i; // Q2
   cout << d1->k; // Q3
   cout << d1->l; // Q4

   cout << d2->i; // Q5
   cout << d2->k; // Q6
   cout << d2->l; // Q7

   b = d1;
   cout << b->i; // Q8
   cout << b->j; // Q9
   cout << b->l; // Q10
}
```

For each statement below which has a question number (e.g., **Q11**), write the output that results from executing the statement on the answer sheet.   All statements are legal.

```
class B { // B.h
public:
  B( );
  virtual ~B( );
  virtual void f1( );
  void f2( );
  void f3( );
};

B::B( ) {  } // B.cpp

B::~B( ) { }

void B::f1( ) {cout << "B::f1" << endl;}

void B::f2( ) {cout << "B::f2" << endl;}

void B::f3( ) {cout << "B::f3" << endl;}

class C : public B { // C.h
public:
  C( );
  C(int );
  virtual ~C( );
  void f1( );
  virtual void f2( );
  void f3( );
};

C::C( ) {  }
C::C(int i) {  }

C::~C( ) { }

void C::f1( ) {cout << "C::f1" << endl;}
void C::f2( ) {cout << "C::f2" << endl;}
void C::f3( ) {cout << "C::f3" << endl;}
```

```
class D : public C {  // D.h
public:
  D(int);
  D( );
  virtual ~D( );
  void f1( );
  void f2( );
  virtual void f3( );
};

D::D(int i) {  } // D.cpp
D::D( ) {  }

D::~D( ) { }

void D::f1( ) {cout << "D::f1" << endl;}
void D::f2( ) {cout << "D::f2" << endl;}
void D::f3( ) {cout << "D::f3" << endl;}

int main(void) { // main.cpp

  C c1(1);
  D d1(1);

  C c2 = d1;
  c2.f2( ); // Q11
  c2.f3( ); // Q12

  B& bR = (B&) c1;
  bR.f1( ); // Q13
  bR.f3( ); // Q14

  C& cR = (C&) d1;
  cR.f1( ); // Q15
  cR.f2( ); // Q16

  B* bP = &c1;
  bP->f2( ); // Q17
  bP->f3( ); // Q18

  C* cP = &d1;
  cP->f1( ); // Q19
  cP->f3( ); // Q20
}
```

For each statement below which has a question number (e.g., **Q21**), write the output that results from executing the statement on the answer sheet. All statements are legal.

```
class B { // B.h
public:
  B( );
  B(int);
  virtual ~B( );
  virtual void f1(int);
  virtual void f1(double);
};

B::B( ) {  } // B.cpp
B::B(int i) {  }

B::~B( ) { }

void B::f1(int i) {
  cout << "B::int" << endl;
};

void B::f1(double) {
  cout << "B::double" << endl;
};

class C : public B { // C.h
public:
  C( );
  C(int );
  virtual ~C( );
  void f1(int);
};

C::C( ) {  } // C.cpp
C::C(int i) {  }

C::~C( ) { }

void C::f1(int) {
  cout << "C::int" << endl;
};
```

```
int main(void) {

  B b1(1);
  C c1(1);
  int i = 1;
  double d = 1.0;

  b1.f1(i); // Q21
  b1.f1(d); // Q22
  c1.f1(d); // Q23

  B* bP = &c1;
  bP->f1(d); // Q24
}
```

For each statement below which has a question number (e.g., **Q25**), write the output that results from executing the statement on the answer sheet. All statements are legal.

```
class B { // B.h
public:
  int i;
  B(int);
  virtual ~B( );
};

B::B(int j) : i(j) {  } // B.cpp

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

```
void f1(B b) { // main.cpp
  b.i = 0;
};

void f2(B& b) {
  b.i = 0;
};

void f3(B* b) {
  b->i = 0;
};

int main(void) {

  B b(4);
  B& bR = b;

  f1(b);
  cout << b.i << endl; // Q25
  b.i = 4;

  f1(bR);
  cout << bR.i << endl; // Q26
  bR.i = 4;

  f2(b);
  cout << b.i << endl; // Q27
  b.i = 4;

  f2(bR);
  cout << bR.i << endl; // Q28

  bR.i = 5;
  cout << b.i << endl; // Q29
}
```

For each statement that is a question, what is printed by constructors or destructors when the statement executes. For **Q30**, what is printed when *foo* is called? For **Q33**, what is printed by the *cout* statement when *foo* is called. All statements are legal.

```
class B {
public:
  int i;
  B( );
  B(int);
  B(B&);
  virtual ~B( );
};

B::B( ) {cout << "B" << endl; i = 0;}
B::B(int j) : i(j) {cout << "B(int)" << endl;}
B::B(B& b) {
  cout << "B(&B b)" << endl;
  this->i=-b.i;}

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

class C : public B {
public:
  C( );
  C(int );
  virtual ~C( );
};

C::C( ) {cout << "C" << endl;}
C::C(int i) : B(i) {cout << "C(int)" << endl;}

C::~C( ) {cout << "~C" << endl;}
```

```
#include "B.h"
#include "C.h"
#include <iostream>
#include <string>
using namespace std;

void foo(B par) { // Q30
   cout << par.i << endl;
}

int main(void) {

  B b1(1); // Q31
  C c1(1); // Q32
  foo(b1); // Q33
}
```

Q34: what, if anything, is anything printed after the call to *foo* when *b1* and *c1* are popped off the stack?

Answer the questions below using the code below. All statements are legal.

```
class Weird { // Weird.h
private:
  int i;
public:
  Weird( );
  Weird(int);
  virtual ~Weird( );

  Weird operator+(Weird);
  Weird getI( );

  friend Weird operator*(Weird, Weird);
  friend ostream& operator<<(ostream& os, const
Weird&);
};

Weird::Weird( ) {i=0;}; // Weird.cpp
Weird::Weird(int j) : i(j) { }
Weird::~Weird() { }

Weird Weird::operator+(Weird w) {
  int res = this->i * w.i;
  return Weird(res);
}

Weird operator*(Weird w1, Weird w2) {
  int res = w1.i + w2.i;
  return Weird(res);
}

ostream& operator<<(ostream& os, const Weird& w) {
  return os << w.i;
}
```

```
#include "Weird.h"
#include <iostream>
using namespace std;

int main(void) {

  Weird w1(3);
  Weird w2(5);
  Weird w3(7);

  cout << w1+w2 << endl; // LINE A
  cout << w1+w2*w3 << endl; // LINE B
}
```

Q35: If the overloaded * was declared in the Weird class, how many parameters need to be specified by the programmer?

Q36: In LINE A, which of w1 and w2 is passed as the *this* pointer to the function?

Q37: What is printed by **LINE A**?

Q38: What is printed by **LINE B**

Q39: Could the overloaded **<<** operator be declared as a member function of the Weird class? Answer "yes" or "no".

Answer the questions below using the code below. All statements are legal.

```
class Exp { // Exp.h
public:
  Exp( );
  virtual ~Exp( );
  string msg( );
};

Exp::Exp( ) { } // Exp.cpp
Exp::~Exp( ) { }
string Exp::msg( ) {return "E1";}

class Exp2 { // Exp2.h
public:
  Exp2( );
  virtual ~Exp2( );
  string msg( );
};

Exp2::Exp2( ) { } // Exp2.cpp
Exp2::~Exp2( ) { }
```

```
void foo( ) { // main.cpp
  throw 1.0;
}

void heave(int i) {
  if (i == 0) throw Exp( );
  if (i == 1) throw 2;
  if (i == 2) throw Exp2( );
  foo( );
}

int main( ) {
  for (int i = 0; i < 3; i++) {
    try {
      heave(i);
    } catch (int i) {
      cout << "caught it " << i << endl;
    } catch (Exp e) {
      cout << e.msg( ) << endl;
    }
  }
  return 0;
```

Q40: What is printed in the try-catch clause when *i = 0*?
Q41: What is printed in the try-catch clause when *i = 1*?
Q42: Would declaring void *heave(int i)* as *void heave(int i) throw(int, Exp, Exp2)* guarantee that only these exceptions are thrown? Answer "yes" or "no".
Q43: Would it be legal to add a finally clause to the try-catch? Answer "yes" or "no".

Answer the questions using the code, which is a template description, below. The code is legal.

```
template <class T1, class T2> class Tuple {
  private:
    T1 v1;
    T2 v2;
public:
  Tuple(T1, T2);
  virtual ~Tuple( );
  void print( );
};

template <class T1, class T2>
Tuple<T1,T2>::Tuple(T1 a1, T2 a2) : v1(a1), v2(a2) { }

template <class T1, class T2>
Tuple<T1,T2>::~Tuple( ) { }

template <class T1, class T2>
void Tuple<T1,T2>::print( ) {
  cout << v1 << ", " << v2 << endl;
}
```

Q44: If a program uses this template by specifying code like Tuple<int, String> in our program, what is the type of v1?

Q45: If a program uses this template by specifying code like Tuple<int, String> in our program, what is the type of v2?

Q46: Does T2 have to be the name of a class? Answer “yes” or “no”.

Answer the questions using the code below. The code is legal.

```
public class T1 extends Thread {

  public static int count=0;

  public T1( ) { }

  private void update( ) {
    int v = count;
    try {
      sleep(10);
    } catch (Exception e) { }
    v++;
    count = v;
  }

  public void run( ) {
    for (int i = 0; i < 1000; i++) {
      update( );
    }
  }
}

public class T2 extends Thread {

  public static int count=0;

  public T2( ) { }

  private synchronized void update( ) {
    int v = count;
    try {
      sleep(10);
    } catch (Exception e) { }
    v++;
    count = v;
  }

  public void run( ) {
    for (int i = 0; i < 1000; i++) {
      update( );
    }
```

```
class Main {

public static void main(String args[]) throws Exception {

    T1 t1_1 = new T1( );
    T1 t1_2 = new T1( );
    t1_1.start( );  t1_2.start( );

    t1_1.join( ); t1_2.join( );
    System.out.println("T1.start, "+T1.count); // LINE A

    T1.count = 0;
    t1_1.run( ); t1_2.run( );
    System.out.println("T1.run, "+T1.count); // LINE B

    T2 t2_1 = new T2( );
    T2 t2_2 = new T2( );
    t2_1.start( ); t2_2.start( );
    t2_1.join( ); t2_2.join( );
    System.out.println("T2, "+T2.count); // LINE C
  }
}
```

Q47: Answer which is most true of what is printed by LINE A:
(a) Exactly 2000
(b) A value that is greater than or equal to 0 and less than or equal to 2000
(c) Any value is possible to be printed

Q48: Answer which is most true of what is printed by LINE B:
(a) Exactly 2000
(b) A value that is greater than or equal to 0 and less than or equal to 2000
(c) Any value is possible to be printed

Q49: Answer which is most true of what is printed by LINE C:
(a) Exactly 2000
(b) A value that is greater than or equal to 0 and less than or equal to 2000
(c) Any value is possible to be printed.

Answer the questions using the code below. T3 is the same as T2 except for the bold code in T3's update function. The code is legal.

```
public class T3 extends Thread {

  public static int count=0;
  private static Object o = new Object( );

  public T3( ) { }

  private void update( ) {
    synchronized(o) {
      int v = count;
      try {
        sleep(10);
      } catch (java.lang.InterruptedException e) { }
      v++;
      count = v;
    }
  }

  public void run( ) {
    for (int i = 0; i < 1000; i++) {
      update( );
    }
  }
}
class Main {

  public static void main(String args[]) throws
java.lang.InterruptedException {

    T3 t3_1 = new T3( );
    T3 t3_2 = new T3( );
    t3_1.start( );
    t3_2.start( );

    t3_1.join( );
    t3_2.join( );
    System.out.println("T3, "+T3.count); // LINE A

  }
}
```

Q50: Answer which is most true of what is printed by LINE A:
(a) Exactly 2000.
(b) A value that is greater than or equal to 0 and less than or equal to 2000.
(c) Any value is possible to be printed.

All answers should be on this sheet. Put your name on the sheet.

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.

39.

40.

41.

42.

43.

44.

45.

46.

47.

48.

49.

50.

ECE 30862 Fall 2015 Second Exam Key

1. OK

2. OK or ERR

3. OK or ERR

4. OK or ERR

5. OK or ERR

6. OK or ERR

7. OK or Err

8. OK

9. OK

10. OK

11. C::f2

12. C::f3

13. C::f1

14. B::f3

15. D::f1

16. D::f2

17. B::f2

18. B::f3

19. D::f1

20. C::f3

21. B::int

22. B::double

23. C::int

24. B::double

25. 4

26. 4

27. 0

28. 0

29. 5

30. B(&B b)

31. B(int)

32. B(int) C(int)

33. -1 ~B

34. ~C ~B ~B

35. 1

36. w1 is this

37. 15

38. 36

39. NO

40. E1

41. caught it 2

42. NO

43. NO

44. int

45. String

46. NO

47. B

48. A

49. B

50. A

Notes:

Questions 2 - 7 When D2 inherits privately from D1, this also hides the inheritance chain, i.e., that D2 inherits from D1 and Base is private information. When D1 inherits protected from Base, it makes it protected information that D1 extends Base. The good thing about this is that it makes D1 and D2 appear to be a monolithic, stand-alone class outside of these classes. The bad thing is that it makes these test questions poorly formed because it is illegal to say *Base* d1 = new D1( );* and *Base* d2 = new D2( );* because it is private that D2 is a Base and protected information that D1 is a base. Thus the declarations are illegal, and therefore questions 2 through 7 don't make sense.

Question 32 Give 1 point if they answer either C(int) or B(int)

Question 33 Give 1 point if they answer either -1 or ˜B

Question 34 Take away 1 point if the order is wrong

Question 36 The working does not have to be the same as mine, just the idea