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.**	