ECE 30862 Spring 2019 Second Exam Answer Sheet

Put your name above!

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.

This page intentionally left almost blank

ECE 30862 Spring 2019, Test 2

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

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

You have until 9:22AM to take this exam. There are 40 questions, 2.5pts each for 100 points. The exam is 10 pages long, including this page. Let Prof. Midkiff know if it isn't.

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 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 answer and the statement is legal at both compile and runtime answer "Ok", and if nothing is printed by the statement gives either a compile time or run time 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 (must be signed to be graded):

Name

Java question. The code on this page is used for questions 1 - 13. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
class B {
   public void f1(int i) {
      System.out.println("B::f1(i))");
   }
   public void f1(double f) {
      System.out.println("B::f1(f))");
   }
   public void f2( ) {
      System.out.println("B::f2");
   }
   public void f5(B b) {
      b.f3( );
   }
   private void f3( ) {
      System.out.println("B::f3");
   }
}
class D extends B {
   public void f1(long l) {
     System.out.println("D::f1(l))");
   }
   public void f2( ) {
     System.out.println("D::f2");
   }
   public void f4( ) {
     System.out.println("D::f4");
   }
   private void f3( ) {
      System.out.println("B::f3");
   }
}
```

```
class Main {

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

      b.f5(b); // Q1

      d.f1(1); // Q2
      d.f1((long) 1); // Q3
      d.f1(1.0); // Q4
      d.f2( ); // Q5
      d.f4( ); // Q6
      b.f5(b); // Q7

      b = d;
      b.f1(1); // Q8
      b.f1((long) 1); // Q9
      b.f1(1.0); // Q10
      b.f2( ); // Q11
      b.f4( ); // Q12
      b.f5(b); // Q13
   }
}
```

Java question. The code on this page and the facing page are used for questions 14 - 17. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
class A {
   public int x = 5;
   public int y = 6;
   public int z = 7;
}
class T implements Runnable {
   public static Object obj = new Object( );

   public synchronized void f1(A a) {
      a.x++;
   }

   public void f2(A a) {
      synchronized(obj) {
         a.y++;
      }
   }

   public void f3(A a) {
      synchronized(a) {
         a.z++;
      }
   }

   public synchronized void run( ) {
      f1(Main.a);
      f2(Main.a);
      f3(Main.a);
   }
}
```

```
class Main {

   public static A a = new A( );

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

      Thread t1 = new Thread(new T( ));
      Thread t2 = new Thread(new T( ));
      t1.run( ); // S1
      t2.run( ); // S2
      t1.start( ); // S3
      t2.start( ); // S4
   }
}
```

Q14: is there a race on y on the calls in S1 and S2?

Q15: is there a race on x on the calls in S3 and S4?

Q16: is there a race on y on the calls in S3 and S4?

Q17: is there a race on z on the calls in S3 and S4?

Java question. The code on this page is used for questions 18 - 20. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
class A {
   public A( ) {val = 0;}
   public A(int i) {val = i;}

   public int val;
}
class B extends A {
   public B(int i) {
      valB = i;
      super(-10); // Q18
   }

   public int valB;
}
class Main {

   public static A a = new A( );

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

      B b = new B(10);
      System.out.println(b.valB); // Q19
      System.out.println(b.val); // Q20
   }
}
```

Java question. The code on this page is used for questions 21 - 23. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
class A {
   public A(int i) {val = i;}
   public int val;
}
class B implements Cloneable {
   public A myA;
   public B(A a) {
      myA = a;
   }

   public Object clone( )
   throws CloneNotSupportedException {
      return (B) super.clone( );
   }
}
class C implements Cloneable {
   public A myA;
   public C(A a) {
      myA = a;
   }

   public Object clone( )
   throws CloneNotSupportedException {
      C c = (C) super.clone( );
      c.myA = new A(myA.val);
      return c;
   }
}
class D implements Cloneable {
   public A myA;
   public D(A a) {
      myA = a;
   }

   public D(D d) {
      myA = new A(d.myA.val);
   }

}
```

```
class Main {

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

      A origA = new A(10);
      B origB = new B(origA);
      C origC = new C(origA);
      D origD = new D(origA);

      B newB = (B) origB.clone( );
      C newC = (C) origC.clone( );
      D newD = new D(origD);

      origA.val = -100;
      System.out.println(newB.myA.val); // Q21
      System.out.println(newC.myA.val); // Q22
      System.out.println(newD.myA.val); // Q23
   }
}
```

C++ question. The code on this page is used for questions 24 - 33. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
// B.h
class B {
public:

   int v;

   B( );
   virtual ~B( );

   virtual void f1( );
   virtual void f2( );
};

// B.cpp
B::B( ) {
   v = 0;
}

B::~B( ) { }

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

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

// D.h
class D : public B {
public:

   int dv;

   D( );
   virtual ~D( );

   virtual void f2( );
   virtual void f3( );
};

// D.cpp
D::D( ) {dv = 0;}
D::~D( ) { }

void D::f2( ) {std::cout << "D::f2" << std::endl;}
void D::f3( ) {std::cout << "D::f3(f)" << std::endl;}
```

```
// main.cpp

void fx(B b) {
   b.v = -10;
}

void fy(B& b) {
   b.v = 10;
}

int main (int argc, char *argv[]) {
   B bvar;
   B& br1 = bvar;

   D dvar;
   B& br2 = dvar; // Q24

   br1.v = 4;
   br2.v = 5; // Q25
   br2.dv = 6; // Q26

   bvar = dvar;
   std::cout << br1.v << std::endl; // Q27
   std::cout << br2.v << std::endl; // Q28


   br1.v = 4;
   br2.v = 5;
   br1 = br2;
   std::cout << bvar.v << std::endl; // Q29
   std::cout << dvar.v << std::endl; // Q30

   br2.f1( ); // Q31
   br2.f2( ); // Q32
   br2.f3( ); // Q33
}
```

C++ question. The code on this page is used for questions 34 - 35. If something is printed, answer what is printed. If nothing is printed and the statement is legal (i.e, no compile time error and no error at that statement at runtime), answer “ok”. If nothing is printed and the statement is not legal, answer “Err”.

```
template <class myType>
myType GetMax (myType a, myType b) {
   return (a>b?a:b);
}

// C.h
class C {
public:

   C(int);

   virtual ~C( );

   virtual bool operator>(const C&);

   friend std::ostream& operator<<
        (std::ostream&, const C&);

private:
   int val;
};

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

C::~C( ) { }

bool C::operator>(const C& c) {
   return this->val > c.val;
}

std::ostream& operator<<
   (std::ostream& os, const C& c) {
   os << c.val;
   return os;
}
```

```
// N.h
class N {
public:

   N(int);

   virtual ~N( );

   friend std::ostream& operator<<
      (std::ostream&, const N&);

private:
   int val;
};

// N.cpp
N::N(int i) : val(i) { }

N::~N( ) { }

std::ostream& operator<<(std::ostream& os, const N
   os << n.val;
   return os;
}

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

   C ci(2);
   C cj(4);

   N ni(2);
   N nj(4);

   std::cout << GetMax<C>(ci,cj); // Q34
   std::cout << GetMax<N>(ni,nj); // Q35
}
```

C++ questions 36 - 40. Answer on the other pages.

```
// B.h
class B {
public:

   B( );
   B(int);

   virtual ~B( );

   virtual void f1( );
   virtual void f2(int);
   void f3( );

};

// B.cpp
B::B( ) {
   std::cout << "B( )" << std::endl;
}

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

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

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

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

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

// D.h
class D : public B {
public:

   D( );
   D(int);

   virtual ~D( );

   virtual void f1( );
   virtual void f2(float);
   void f3( );
   void f4( );

};
```

```
// D.cpp
D::D( ) {
   std::cout << "D( )" << std::endl; B(1);
}

D::D(int) {
   std::cout << "D(int)" << std::endl;
}

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

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

void D::f2(float) {
   std::cout << "D::f2(float)" << std::endl;
}

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

void D::f4( ) {
   std::cout << "D::f4( )" << std::endl;
}

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

   B* bP;
   D* dP = new D( ); // Q36

   dP->f2(1); // Q37
   delete dP; // Q38

   bP = new D( );
   bP->f3( ); // Q39
   bP->f4( ); // Q40
}
```