ECE 30862 Spring 2019 First 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 1

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 9:00PM to take this exam. The total number of points should be 100, Each of the 50 questions is worth 2 points. After taking the test turn in both the test and the answer sheet.

Your exam should have 8 (seven) pages total (including this cover page, the answer sheet and one almost entire blank page). 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.*

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

The code on this page and the facing page are used for questions 1 - 28. 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". For question 28 say what is printed by either branch of the **if** statement.

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

   int v;

   B( );
   virtual ~B( );

   virtual void f1(B* p); // no override
   virtual void f2( ); // override
   virtual void f3(int); // no over, but hide
   void f4( ); // override  D1

private:
   virtual void f5( );
};

// B.cpp
B::B( ) {
   std::cout << "B( )";
   v = 0;
}

B::~B( ) { }

void B::f1(B* p) {
   std::cout <<  "B::f1(p)";
   p->f5( );
}

void B::f2( ) {std::cout << "B:f2( )";}
void B::f3(int) {std::cout << "B:f3(i)";}
void B::f4( ) {std::cout << "B:f4";}
void B::f5( ) {std::cout << "B:f5";}
```

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

   int v;

   D( );
   virtual ~D( );

   virtual void f2( );
   virtual void f3(float);
   void f4( );
   virtual void f6( );

private:
   virtual void f5( );
};

// D.h
D::D( ) {std::cout << "D( )";}
D::~D( ) { }

void D::f2( ) {std::cout << "D:f2";}
void D::f3(float f) {std::cout << "D:f3(f)";}
void D::f4( ) {std::cout << "D:f4";}
void D::f6( ) {std::cout << "D:f6";}
void D::f5( ) {std::cout << "D:f5";}
```

```
// main.cpp
void funcv(B b) {
   b.v = -99;
}

void funcr(B& b) {
   b.v = -98;
}

void funcp(B* b) {
   b->v = -97;
   b = 0;
}
```

```
// main.cpp
int main (int argc, char *argv[]) {
   B bv;
   B& b1r = bv;

   D dv;
   B* bp = new D( ); // Q1
   B& b2r = dv; // Q2

   bv = dv;
   bv.f1(bp);  // Q3
   bv.f2( ); // Q4
   bv.f3(1.0); // Q5
   bv.f4( ); // Q6
   bv.f5( ); // Q7
   bv.f6( ); // Q8

   b2r.f1(bp);  // Q9
   b2r.f2( ); // Q10
   b2r.f3(1.0); // Q11
   b2r.f4( ); // Q12
   b2r.f6( ); // Q13

   bp->f1(bp);  // Q14
   bp->f2( ); // Q15
   bp->f3(1.0); // Q16
   bp->f4( ); // Q17
   bp->f6( ); // Q18

   dv = bv; // Q19

   bv.v = 0;
   funcv(bv);
   std::cout << bv.v; // Q20

   bv.v = 0;
   funcv(b1r);
   std::cout << b1r.v; // Q21

   bv.v = 0;
   funcr(bv);
   std::cout << bv.v; // Q22

   bv.v = 0;
   funcr(b1r); // Q23
   std::cout << b1r.v; // Q24

   bv.v = 0;
   funcp(b1r); // Q25
   std::cout << b1r.v; // Q26

   funcp(bp);
   std::cout << bp->v; // Q27

   if (bp == 0) std::cout << "0"; // Q28
   else std::cout << "!0";
}
```

The code on this page is used for questions 29 - 36. 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".

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

   int v;
   static int sv;

   C( );
   virtual ~C( );

   static void fs( );

   virtual void setSv(int);
   virtual void setV(int);
};

// C.cpp
int C::sv = 0;

C::C( ) {std::cout << "ctor C";}
C::~C( ) {std::cout << "dtor C";}

void C::fs( ) {
   v = 1.0; // Q29
   sv = -1.0; // Q30
}

void C::setSv(int i) {sv = i;}

void C::setV(int i) {v = i;}

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

   int v;
   static int sv;

   D( );
   virtual ~D( );
};

// D.cpp
D::D( ) {std::cout << "ctor D";}

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

```
// main.cpp
int f(int i) {std::cout << "fi";}

int f(double z) {std::cout << "fz";}

int main (int argc, char *argv[]) {
   C c1; // Q31
   C c2;
   D d; // Q32

   c1.setV(2);
   c1.setSv(4);
   c2.setV(6);
   c2.setSv(8);

   std::cout << c1.v << " " << c1.sv; // Q33
   std::cout << c2.v << " " << c2.sv; // Q34

   f(1); // Q35
   f(1.0); // Q36
}
```

The code on this page is used for questions 37 - 43. 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".

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

   Animal(bool);
   virtual ~Animal( );

   virtual bool getWarmBlooded( );

private:
   bool warmBlooded;
};

// Animal.cpp
Animal::Animal(bool w) {warmBlooded = w;}
Animal::~Animal( ) { };

bool Animal::getWarmBlooded( ) {return warmBlooded;}

// Dog.h
class Dog : public Animal {
public:

   Dog(float, int);
   virtual ~Dog( );

   virtual int getBreedID( );
   virtual float getWeight( );

private:
   int weight;
   float breedID;
};

// Dog.cpp
Dog::Dog(float w, int i) :
   Animal(true), weight(w), breedID(i) { }

Dog::~Dog( ) { }

int Dog::getBreedID( ) {return breedID;}
float Dog::getWeight( ) {return weight;}
```

```
// Fish.h
class Fish : public Animal {
public:

   Fish(int);
   virtual ~Fish( );

   virtual int getBreedID( );

private:
   int breedID;

};

// Fish.cpp
Fish::Fish(int i)
   : Animal(false), breedID(i) { }

Fish::~Fish( ) { }

int Fish::getBreedID( ) {return breedID;}

// main.cpp
int main (int argc, char *argv[]) {
   Fish* fish = new Fish(5);
   Dog* dog = new Dog(47.5, 5);
   Animal* animal = new Animal(true);
   bool b;

   fish = animal; // Q37

   fish = new Fish(5);
   animal = fish; // Q38

   fish = new Fish(5);
   animal = new Animal(true);
   animal = static_cast<Animal*>(fish); // Q39
   dog = static_cast<Dog*>(animal); // Q40
   dog = static_cast<Dog*>(fish); // Q41

   animal = new Animal(true);
   dog = dynamic_cast<Dog*>(animal);
   b = dog->getBreedID( ); // Q42
   dog = dynamic_cast<Dog*>(fish);
   b = dog->getBreedID( ); // Q43
}
```

The code on this page is used for questions 44 - 50. 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".

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

   Animal(bool);
   virtual ~Animal( );

   virtual bool getWarmBlooded( );

   bool warmBlooded;
};

// Animal.cpp
Animal::Animal(bool w) {warmBlooded = w;}
Animal::~Animal( ) { };

bool Animal::getWarmBlooded( ) {return warmBlooded;}

// Dog.h
class Dog : private Animal {
public:

   Dog( );
   Dog(float, int);
   virtual ~Dog( );

   virtual int getBreedID( );
   virtual float getWeight( );

protected:
   int weight;
   float breedID;
};

// Dog.cpp
Dog::Dog( ) : Animal(true) { }

Dog::Dog(float w, int i) : Animal(true) {
   weight = w; // Q44
   breedID = i; // Q 45
}

Dog::~Dog( ) { }

int Dog::getBreedID( ) {return breedID;}
float Dog::getWeight( ) {return weight;}
```

```
// Lab.h
class Lab : public Dog {
public:

   Lab(float, int);
   virtual ~Lab( );

   virtual int getBreedID( );
   virtual float getWeight( );
};

// Lab.cpp
Lab::Lab(float w, int i) : Dog( ) {
   warmBlooded = true; // Q46
   weight = w; // Q47
   breedID = i; // Q48
}

Lab::~Lab( ) { }

int Lab::getBreedID( ) {return breedID;}
float Lab::getWeight( ) {return weight;}

// main.cpp
int main (int argc, char *argv[]) {
   Dog* dog = new Dog(47.5, 5);
   Animal* animal = new Animal(true);

   animal->warmBlooded = true; // Q49
   dog->weight = 70.0; // Q50
}
```