

ECE 30862 Fall 2019 First Exam Answer Sheet**Put your name above!**

- | | |
|-----|-----|
| 1. | 21. |
| 2. | 22. |
| 3. | 23. |
| 4. | 24. |
| 5. | 25. |
| 6. | 26. |
| 7. | 27. |
| 8. | 28. |
| 9. | 29. |
| 10. | 30. |
| 11. | 31. |
| 12. | 32. |
| 13. | 33. |
| 14. | 34. |
| 15. | 35. |
| 16. | 36. |
| 17. | 37. |
| 18. | 38. |
| 19. | 39. |
| 20. | 40. |

This page intentionally left almost blank

ECE 30862 Fall 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 7:30 to take this exam. The total number of points should be 100, Each of the 40 questions is worth 2.5 points. After taking the test turn in both the test and the answer sheet.

Your exam should have 8 (eight) 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 answer sheet whatever assumptions you made to answer the question, and answer it under those assumptions.

Check the front board occasionally for corrections.

Programs may be given without “#include” statements for brevity. Assume all needed includes are present. “std::endl” has been left off for brevity. You may use newlines in your answer, or not, without affecting its correctness.

Each page has the following instructions:

The code on this page and the facing page are used for questions x - y. If something is printed on a line that is a question (has a Qx comment, where “x” is a natural number) say what is printed. If the line has an error at either compile or runtime, answer “Err” and assume the statement doesn’t exist for the rest of the program. If the statement prints nothing but is correct, answer “Ok”. If a value is uninitialized, answer “uninit” or something similar.

where “x” and “y” are question numbers, so you don’t need to read these if you are running short on time.

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 x - y. If something is printed on a line that is a question (has a Qx comment, where “x” is a natural number) say what is printed. If the line has an error at either compile or runtime, answer ”Err” and assume the statement doesn’t exist for the rest of the program. If the statement prints nothing but is correct, answer ”Ok”. If a value is uninitialized, answer ”uninit” or something similar.

```

// A.h
class A {
public:
    int v;

    A( );
    virtual ~A( );

    virtual void a1( );
    virtual void a1(A* a);
    virtual void a2( );
    virtual void a3(int);
    void a4( );

protected:
    virtual void a5( );

private:
    virtual void a6( );
    void a7( );
};

// A.cpp
A::A( ) {std::cout << "A "};
A::~~A( ) {std::cout << "~A "};

void A::a1( ) {
    std::cout << "a1 ";
    a6( );
}
void A::a1(A* a) {
    std::cout << "a1a ";
    a->a7( );
}

void A::a2( ) {std::cout << "a2 "};
void A::a3(int) {std::cout << "a3i "};

void A::a4( ) {std::cout << "a4 "};
void A::a5( ) {std::cout << "a5 "};

void A::a6( ) {std::cout << "a6 "};
void A::a7( ) {std::cout << "a7 "};

// B.h
class B : public A {
public:
    int v;

    B( );
    virtual ~B( );

    virtual void b1( );
    void a2( );
    virtual void a3(double);
    virtual void a4( );

private:
    void a6( );
};

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

void B::b1( ) {
    std::cout << "b1 ";
    a5( );
}

void B::a2( ) {std::cout << "B::a2 "};

void B::a3(double) {std::cout << "B::a3d "};

void B::a4( ) {std::cout << "B::a4 "};

void B::a6( ) {std::cout << "B::a6 "};

```

```

// C.h
class C : public B {
public:
    C( );
    virtual ~C( );
    void a4( );
    virtual void c1( );
};

// C.cpp
C::C( ) {std::cout << "C ";}
C::~~C( ) {std::cout << "~C ";}

void C::a4( ) {std::cout << "Ca4 ";}
void C::c1( ) {std::cout << "c1 ";}

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

    A* aP1 = new A( ); // Q1
    A* aP2 = new B( ); // Q2

    aP1->a1( ); // Q3

    aP1->a1(aP2); // Q4

    aP1->a4( ); // Q5

    aP1->a5( ); // Q6
    aP1->a6( ); // Q7

    aP2->a1( ); // Q8

    aP2->a1(aP2); // Q9

    aP2->a2( ); // Q10

    aP2->a3(2); // Q11

    aP2->a4( ); // Q12

    aP2->b1( ); // Q13

    B* bP = new C( ); // Q14

    bP->a1( ); // Q15

    bP->a1(bP); // Q16

    bP->a2( ); // Q17

    bP->a3(3); // Q18

    bP->a4( ); // Q19

    bP->b1( ); // Q20

    bP->c1( ); // Q21

    bP->a5( ); // Q22
}

```

The code on this page and the facing page are used for questions x - y. If something is printed on a line that is a question (has a Qx comment, where “x” is a natural number) say what is printed. If the line has an error at either compile or runtime, answer ”Err” and assume the statement doesn’t exist for the rest of the program. If the statement prints nothing but is correct, answer ”Ok”. If a value is uninitialized, answer ”uninit” or something similar.

```

// A.h
class A {
public:
    A(int);
    ~A( );
    virtual void print( );
    static void inc( );
    int x;

protected:
    int y;

private:
    int z;
    static int numPrints;
};

// A.cpp
int A::numPrints = 0;

A::A(int p) : z(p), y(1), x(z) {}

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

void A::print( ) {
    inc( );
    std::cout << x << " ";
    std::cout << y << " ";
    std::cout << z << " ";
    std::cout << numPrints << " ";
}

void A::inc( ) {numPrints++;}

// B.h
class B : protected A {
public:
    int v;

    B(int);
    ~B( );

    virtual void print( );
};

// B.cpp
B::B(int p) : A(p) { }
B::~B( ) {std::cout << "~B ";}

// the following line calls A's print
void B::print( ) {A::print( );}

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

// C.cpp
C::C(int p) : B(p) { }
C::~C( ) {std::cout << "~C ";}

// the following line calls B's print
void C::print( ) {B::print( );}

```

```
// main.cpp
void barB(B b) { }

void barC(C c) { }

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

    A* aP = new A(1);
    B* bP = new B(2);
    C c(3);
    C* cP = &c;

    std::cout << aP->x << " "; // Q23

    std::cout << bP->x << " "; // Q24

    std::cout << cP->y << " "; // Q25

    cP->print( ); // Q26

    barB(c); // Q27

    barC(c); // Q28
}
```

The code on this page and the facing page are used for questions x - y. If something is printed on a line that is a question (has a Qx comment, where “x” is a natural number) say what is printed. If the line has an error at either compile or runtime, answer ”Err” and assume the statement doesn’t exist for the rest of the program. If the statement prints nothing but is correct, answer ”Ok”. If a value is uninitialized, answer ”uninit” or something similar.

```

// A.h
class A {
public:
    virtual void f( )=0;
    virtual void g( )=0;
};

// A.cpp is intentionally empty

// B.h
class B : public A {
public:
    void f( );
};

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

// C.h
class C : public B {
public:
    C( );
    void g( );
    int x;
};

// C.cpp
C::C( ) : x(10) { }
void C::g( ) {std::cout << "g " ;}

// main.cpp
void barI(int j) {
    j = 10;
    std::cout << j << " ";
}

void barP(C* cP) {
    cP->x = 4;
    std::cout << cP->x << " ";
}

void barO(C c) {
    c.x = 4;
    std::cout << c.x << " ";
}

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

    A*a1P = new A( ); // Q29
    B* bP = new B( ); // Q30
    int i=0;
    C* cP = new C( ); // Q31
    C c;
    A* aP2 = cP; // Q32
    aP2->f( ); // Q33
    aP2->g( ); // Q34
    cP->f( ); // Q35

    std::cout << i << " "; // Q36
    barO(c); // Q37
    std::cout << c.x << " "; // Q38
    barP(cP); // Q39
    std::cout << cP->x << " "; // Q40
}

```