

ECE 29595C Fall 2021 First Exam Answer Sheet

Write your name in the space above, it is worth 4 points.

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

This page intentionally left almost blank

now is the end

ECE 39595C Fall 2021, Test 1

The first page is the answer sheet. You should sign it and turn it in separate from the rest of the test booklet. Signing the answer sheet is worth 4 points.

The rest of the test booklet should be signed and turned in separately from the answer sheet. Signing your booklet is worth 1 point.

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

There are 38 questions and each question is worth 2.5 points. Unless a page says otherwise, for each line with a “\Qx, where “x” is some integer, answer with “Error” if the statement will give a compile time or runtime error. If a statement is erroneous, assume it is not in the program for answering the remaining questions. If a statement is not an error, answer what is printed if it prints something, and “Ok” otherwise.

Do not open the exam or begin working on the exam until told to do so. The test is open book, open notes, and absolutely no electronics.

By taking and turning in a test answer sheet to be graded, you agree that: I have neither given nor received help during this exam from any other person or electronic source (other than my own notes, viewing the exam and using an electronic device to annotate the answer sheet with my answers), and I understand that if I have I will be guilty of cheating and will fail the exam and possibly the course.

The code on this and the facing page is for questions 1 through 18.

```
class B {
private:
    virtual void f6( );
    virtual void f7( );

public:
    B( );
    virtual ~B( );

    virtual void f1( );
    virtual void f2( );
    virtual void f3( );
    virtual void f4(B* b);
    virtual void f5(B* b);
    void f9( );
    int i = 0;
    int j = 0;
};
```

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

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

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

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

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

void B::f4(B* b) {
    std::cout << "B::f4" << std::endl;
    b->f6( );
}

void B::f5(B* b) {
    std::cout << "B::f5" << std::endl;
    b->f3( );
}

void B::f6( ) {
    std::cout << "B::f6" << std::endl;
}

void B::f7( ) {
    std::cout << "B::f7" << std::endl;
}

void B::f9( ) {
    std::cout << "B::f9" << std::endl;
}
```

```

class K1 : public B {
private:
    void f6( );

public:
    K1( );
    K1(int m);
    virtual ~K1( );

    virtual void f2( );
    virtual void f3( );
    virtual void f8( );
    virtual void f9( );
    int i = 1;
};

K1::K1( ) {
    std::cout << "K1" << std::endl;
}

K1::K1(int m) {
    std::cout << "K1 int" << std::endl;
}

K1::~~K1( ) {
    std::cout << "~K1" << std::endl;
}

void K1::f2( ) {
    std::cout << "K1::f2" << std::endl;
}

void K1::f3( ) {
    std::cout << "K1::f3" << std::endl;
}

void K1::f6( ) {
    std::cout << "K1::f6" << std::endl;
}

void K1::f8( ) {
    std::cout << "K1::f8" << std::endl;
}

void K1::f9( ) {
    std::cout << "K1::f9" << std::endl;
}

```

```

class K2 : public K1 {
public:
    K2( );
    virtual ~K2( );
    virtual void f3( );
    void f9( );
};

K2::K2( ) {
    std::cout << "K2" << std::endl;
}

K2::~~K2( ) {
    std::cout << "~K2" << std::endl;
}

void K2::f3( ) {
    std::cout << "K2::f3" << std::endl;
}

void K2::f9( ) {
    std::cout << "K2::f10" << std::endl;
}

int main(int argc, char* argv[ ]) {
    B* bb = new B( );
    B* bx = new B( );
    B* bk1 = new K1( ); // Q1
    B* bk2 = new K2( ); // Q2
    std::cout << bb->i << std::endl; // Q3
    std::cout << bk1->i << std::endl; // Q4
    bb->f4(bk1); // Q5
    bb->f5(bk1); // Q6
    bb->f6( ); // Q7
    bk1->f1( ); // Q8
    bk1->f2( ); // Q9
    bk1->f8( ); // Q10
    bk1->f9( ); // Q11
    bk2->f3( ); // Q12
    K1* k1 = new K1( );
    k1->f8( ); // Q13
    k1->f9( ); // Q14
    std::cout << k1->j << std::endl; // Q15
    k1 = new K2( );
    k1->f9( ); // Q16
    delete bk2; // Q17
    B b;
    K1 k10bj;
    b = k10bj;
    b.f3( ); // Q18
}

```

The code on this and the facing page is for questions 19 through 30. For the two statements labeled "Q20", answer "Ok" if both compile without an error, and "Error" if either or both give compile time error(s).

```

class B {
public:
    B( );
    B(B&);
    B(int);
    virtual void f( );
    int i;
};

B::B( ) :i(0) { }

B::B(B& orig) {
    i = orig.i+1;
}

B::B(int _i) {
    i = _i;
}

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

class D : public B {
public:
    virtual void f( );
};

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

// main.cpp
void f1(B& bR) {
    std::cout << bR.i << std::endl;
    bR.i=1;
}

void f2(B b) {
    std::cout << b.i << std::endl;
    b.i = 2;
}

int main(int argc, char* argv[ ]) {
    B b1;
    B& b1R = b1; // Q19
    B& b1aR; // Q20
    b1aR = b1; // Q20

    f1(b1R); // Q21
    std::cout << b1.i << std::endl; // Q22

    b1.i = 0;
    f1(b1); // Q23
    std::cout << b1R.i << std::endl; // Q24

    b1.i = 0;
    f2(b1); // Q25
    std::cout << b1R.i << std::endl; // Q26

    b1.i = 0;
    f2(b1R); // Q27
    std::cout << b1R.i << std::endl; // Q28

    B b2;
    b1R = b2;
    b1R.i = 50;
    std::cout << b2.i << std::endl; // Q29

    D d;
    B& b2R = d;
    b2R.f( ); // Q30
}

```

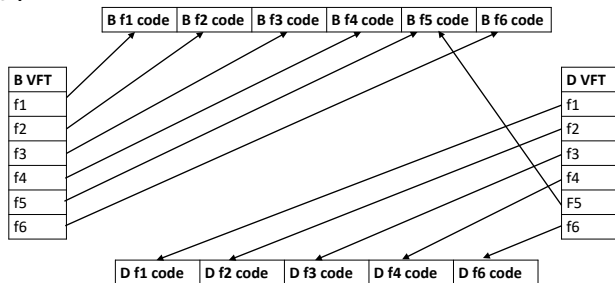
The code and diagrams on this page are for Question 31.

```
class B {
private:
    virtual void f1( );

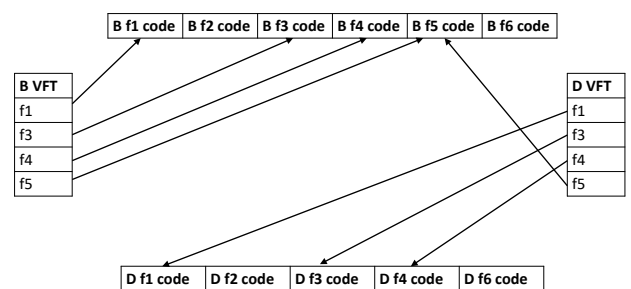
public:
    void f2( );
    virtual void f3( );
    virtual void f4( );
    virtual void f5( );
};
```

```
class D : public B {
private:
    virtual void f1( );

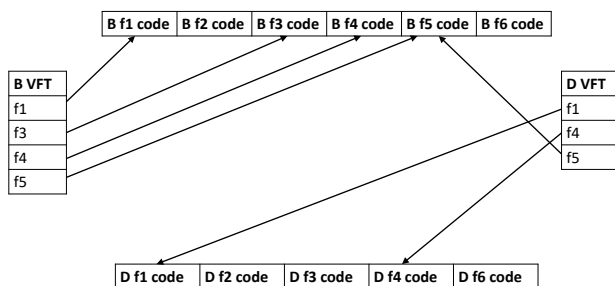
public:
    void f2( );
    void f3( );
    virtual void f4( );
};
```



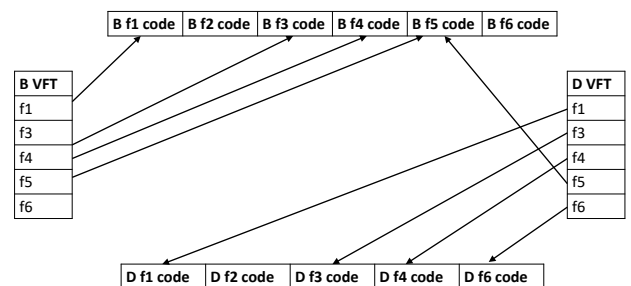
(A)



(B)



(C)



(D)

Q31: What is the correct VFT, given the .h files? Answer a, b, c, d or e.:w

- VFT (A) above
- VFT (B) above
- VFT (C) above
- VFT (D) above
- None of the above.

The code on this page is for questions 32 through 35. For each statement on this page, answer “Ok” if the access does not lead to a compile time error, and “Error” if it does.

```
class B {
private:
    int priv = 1;

protected:
    int prot = 1;

public:
    int publ = 1;
};

class C : private B {
public:
    virtual void f1( );
};

void C::f1( ) {
    std::cout << prot << std::endl; // Q32
    std::cout << priv << std::endl; // Q33
}

class D : public B {
public:
    virtual void f1( );
};

void D::f1( ) {
    std::cout << prot << std::endl; // Q34
}

int main(int argc, char* argv[ ]) {
    C c;
    D d;
    std::cout << c.publ << std::endl; // Q35

    c.f1( );
    d.f1( );
}
```


The code on this page is for questions 36 through 38.

```
class B {
public:
    B( );
    ~B( );
};

B::B( ) { }

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

class C : public B {
public:
    D d;
    C( );
    ~C( );
};

C::C( ) { }

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

class D {
public:
    D( );
    ~D( );
};

D::D( ) { }

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

int main(int argc, char* argv[ ]) {
    B* b = new C( );
    delete b; // Q36
}
```

Q37: Class C has

- a. an ISA relation with class D
- b. a HASA relation with class D
- c. neither.

Q38: Class C has

- a. an ISA relation with class B
- b. a HASA relation with class B
- c. neither.