

# ECE 30862 Fall 2012, Final Exam

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

You have until 10:00 to take this exam.

Your exam should have 20 pages total (including this cover sheet). *Please let Prof. Midkiff know immediately if it does not.* Each problem is worth 5.5 points.

This exam is open book, open notes, but 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.*

All questions are worth 4 points.

**Name:**

# 1 C++ Question

```
void f(int i) {
    if (i > 0) {
        throw 29;
        cout << "> 0, ";
    } else {
        cout << "<= 0, ";
    }
}

void g( ) {
    for (int i = 0; i < 2; i++) {
        try {
            f(i);
            cout << i << " ";
        } catch (int j) {
            cout << "E " << j ;
        }
    }
    cout << " the end." << endl;
}

int main( ) {
    g( );
}
```

What is printed?

- a.** <= 0, 0 E 29 theend.
- b. <= 0, 0 0 > 0, 1 E 29 theend.
- c. <= 0, 0 0 > 0, E 29 the end.
- d. <= 0, 0 E 29

## 2 Java Question

```
class F extends Exception {  
    public F( ) { }  
    public void print( ) {System.out.print("E ");}  
}  
  
class Test {  
    private static void f(int i) throws F {  
        if (i > 0) throw new F( );  
    }  
  
    public static void main(String args[]) {  
        for (int i = -1; i < 1 ; i++) {  
            try {  
                f(i);  
                System.out.print(i+" ");  
            } catch (F f) {  
                f.print( );  
            }  
        }  
        System.out.println(" end");  
    }  
}
```

2012\_final\_key.pdf

What is printed?

- a. -1 0 F 1 end
- b. -1 0 1 F end
- c. -1 F 1 end
- d. -1 F

This question is broken.

For the question as given, the correct answer would be -1 0 end

If the for loop is changed to be "for (int i = -1; i < -1; i++) {" the answer would be "-1 0 E end"

### 3 C++ Question

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

    virtual void print( ) {
        cout << "B::print( ) ";
    }

    virtual void print(int i) {
        cout << "B::print(int)" << endl;
    }
};

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

    virtual void print() {
        cout << "D::print( ) ";
    }

    virtual void print(int i) {
        cout << "D::print(int)" << endl;
    }
};

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

    D *d = new D( );
    B *b = (B*) d;;

    b->print( );
    d->print(4);

    return 0;
}
```

What is printed?

- a. D::print( ) D::print(int)**
- b. D::print( ) B::print(int)
- c. B::print( ) B::print(int)
- d. B::print( ) D::print(int)
- e. none of the above, the program is in error because two functions within a class have the same name.

## 4 C++ Question

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

    virtual void print() {
        cout << "B::print() ";
    }

    virtual void print(int i) {
        cout << "B::print(int)";
    }
};

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

    virtual void print() {
        cout << "D::print() ";
    }
};

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

    D *d = new D();
    B *b = (B*) d;;

    b->print(4);
    d->print();

    return 0;
}
```

What is printed?

- a. D::print() D::print()
- b. B::print(int) D::print()**
- c. D::print() B::print(int)
- d. B::print(int) B::print()
- e. none of the above, the program is in error because two functions within a class have the same name.

## 5 C++ Question

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

    virtual void print( ) {
        cout << "B::print( ) ";
    }

    void print(int i) {
        cout << "B::print(int)";
    }
};

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

    virtual void print() {
        cout << "D::print( ) ";
    }

    void print(int i) {
        cout << "D::print(int)";
    }
};

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

    D *d = new D( );
    B *b = (B*) d;;

    b->print(4);
    d->print(4);

    return 0;
}
```

What is printed?

- a. D::print(int) D::print(int)
- b. D::print(int) B::print(int)
- c. B::print(int)D::print(int)**
- d. B::print(int) B::print(int)
- e. none of the above, the program is in error because two functions within a class have the same name.

## 6 Java Question

```
class B {
    public B( ) { }
    public void print( ) {System.out.print("B::print( ) ");}
    public void print(int i) {System.out.print("B::print(int)");}
}

class D extends B {

    public D( ) { }
    public void print( ) {System.out.print("D::print( ) ");}
    public void print(int i) {System.out.print("D::print(int)");}
}

class Test {

    public static void main(String args[]) {

        D d = new D( );
        B b = d;
        b.print(4);
        d.print(4);
    }
}
```

What is printed?

- a. D::print(int) D::print(int)**
- b. D::print(int) B::print(int)
- c. B::print(int)D::print(int)
- d. B::print(int) B::print(int)
- e. none of the above, the program is in error because two functions within a class have the same name.

## 7 Java Question

```
class B {
    public B( ) { }
    public void print( ) {System.out.print("B::print( ) ");}
    public void print(int i) {System.out.print("B::print(int)");}
}

class D extends B {

    public D( ) { }
    public void print( ) {System.out.print("D::print( ) ");}
}

class Test {

    public static void main(String args[]) {

        D d = new D( );
        B b = d;
        b.print(4);
        d.print(4);
    }
}
```

What is printed?

- a. D::print(int) D::print(int)
- b. D::print(int) B::print(int)
- c. B::print(int)D::print(int)
- d. B::print(int) B::print(int)**
- e. none of the above, the program is in error because two functions within a class have the same name.



## 8 Java Question

```
class B {
    public B( ) { }
    public void print( ) {System.out.print("B::print( ) ");}
    public void print(float f) {System.out.print("B::print(float) ");}
}

class D extends B {

    public D( ) { }
    public void print(int i) {System.out.print("D::print(int) ");}
}

class Test {

    public static void main(String args[]) {

        D d = new D( );
        B b = d;
        b.print(4);
        d.print(4);
    }
}
```

What is printed?

- a. B::print(float) D::print(int)
- b. D::print(int) D::print(int)
- c. none of the above, the program is in error because two functions within a class have the same name.

## 9 C++ Question

```
class Circle {
public:
    float radius;
    Circle(float r) {radius = r;}
    double area( ) {return 3.14*radius;}
    virtual double circumference( ) = 0;
};

class Sphere : public Circle {
public:
    Sphere(float r) : Circle(r) { };
    double area( ) {return 4 * 3.14 * radius*radius;}
    double circumference( ) {return 2 * 3.14 * radius;}
    double volume( ) {return 4.0/3.0 * 3.14 * radius*radius*radius;}
};

int main( ) {
    Sphere * s = new Sphere(2.0);
    cout << s->volume( );
}
```

What is the best answer?

- a. Because **circumference** returns a double, its declaration should be **virtual double circumference( ) = 0.0;**
- b. **Circle** is an abstract class because it contains an abstract function.
- c. **circumference** is an abstract function.
- d. **circumference** only needs to be declared in **Sphere** if it is called.
- e. b and c above.

## 10 Java Question

```
abstract class Circle {
    public double radius;
    public Circle(double r) {radius = r;}
    public double area( ) {return 3.14*radius;}
    public abstract double circumference( );
};

class Sphere extends Circle {
    public Sphere(double r) {super(r);}
    public double area( ) {return 4 * 3.14 * radius*radius;}
    public double circumference( ) {return 2 * 3.14 * radius;}
    public double volume( ) {return 4.0/3.0 * 3.14 * radius*radius*radius;}

    static void main( ) {
        Sphere s = new Sphere(2.0);
        System.out.println(s.volume( ));
    }
}
```

What is the best answer?

- a. **Sphere** has to define all methods in **Circle**, not just **circumference**, because **Circle** is abstract.
- b. Sphere has to define circumference because it is an abstract function that has not been defined.**
- c. **Sphere** has to define **circumference** only if it is called.

## 11 C++ Question

```
class B {
public:
    int age;
    B() {age = 20;}
};

class B1 : virtual public B {
public:
    B1() { age=1; }
    ~B1() { };
};

class B2 : virtual public B {
public:
    B2() { age=2; }
    ~B2() { };
};

class D : public B1, public B2 {
public:
    D() : B(), B1(), B2() {age=2; }
    ~D();
};

int main(int argc, char * argv[]) {
    D* d = new D();
    cout << d->age << endl;
}
```

What is the best answer?

- a. The “virtual” keyword is not needed when class B1 and B2 inherit B.
- b. The “virtual” keyword would not be needed if only B1, and not B2, inherited from B in this program.
- c. The “virtual” keywords are necessary because there would be two copies of a B object accessible from D without it.
- d. b and c.**

## 12 C++ Question

```
class B1 {
public:
    int i;
    B1( ) { i=0; }
    ~B1( ) { };
};

class B2 {
public:
    int i;
    B2( ) { i=0; }
    ~B2( ) { };
};

class D : public B1, public B2 {
public:
    D( ) {i=2; }
    ~D();
};

int main(int argc, char * argv[]) {
    D* d = new D( );
}
```

What is the best answer?

- a. This program is illegal because multiple inheritance is illegal in C++.
- b. This program will give an error because it is ambiguous whether the i in B1 or B2 is referenced in the constructor "D( ) i=2; "**
- c. This program is legal, and will execute.

## 13 C++ Question

```
class MyComplex {
private:
    double re, im;
public:
    MyComplex(double r, double i) : re(r), im(i) { }
    // MyComplex(const MyComplex& orig) { re = orig.re; im = orig.im; }

    MyComplex operator-( const MyComplex& arg) {
        return MyComplex(re-arg.re, im-arg.im);
    }

    MyComplex operator-( ) {
        return MyComplex(-re, -im);
    }

    friend ostream& operator<< (ostream& os, const MyComplex& arg);
};

ostream& operator<< (ostream& os, const MyComplex& arg) {
    os << "(" << arg.re << ", " << arg.im << ")" << endl;
    return os;
}

int main( ) {
    MyComplex first(3,4);
    cout << first - -first << endl;
    return 0;
}
```

What is the best answer?

- a. Making “operator<< ” a friend of the MyComplex class allows it to accept an argument of type MyComplex.
- b. Making “operator<<” a friend of the MyComplex class allows it to access private fields in the class MyComplex.**
- c. “friend” has no meaning in C++, and this is an illegal program.

## 14 C++ Question

```
class MyComplex {
public:
    double re, im;
    MyComplex(double r, double i) : re(r), im(i) { }
    friend MyComplex operator-(const MyComplex& arg1, const MyComplex& arg2);
};

MyComplex operator-( const MyComplex& arg1, const MyComplex& arg2) {
    double d1 = arg1.re - arg2.re;
    double d2 = arg1.im - arg2.im;
    return MyComplex(d1, d2);
}

ostream& operator<< (ostream& os, const MyComplex& arg) {
    os << "(" << arg.re << ", " << arg.im << ")" << endl;
    return os;
}

int main( ) {
    MyComplex first(3,4);
    MyComplex second(2,9);

    cout << first - second << endl;
    return 0;
}
```

What is the best answer?

- a. Making “operator- a friend of the MyComplex class allows it to access private fields in the class MyComplex.
- b. “operator-” is a binary operator, and in the expression “first - second”, “arg1” is “first” and “arg2” is “second”.
- c. both a and b.**

## 15 Java Question

```
class B {
    public int i;
    public B(int i) {this.i = i;}
}

class Test {

    public static void foo(B bb) {
        bb.i = 58;
        bb = new B(97);
    }

    public static void main(String args[]) {

        B b = new B(29);
        System.out.print(b.i+" ");
        foo(b);
        System.out.print(b.i+" ");
    }
}
```

What is printed?

- a. 29 58
- b. 29 29
- c. 29 97



## 16 C++ Question

```
void f(int* p) {
    int j = 10;
    p = &j;
}

int main( ) {
    int i = 9;
    int* p = &i;
    cout << *p << " ";
    f(p);
    cout << *p << endl;
    return 0;
}
```

What is printed?

- a. 9 10
- b. 10 9
- c. 9 9**

## 17 Java Question

```
interface Shape {
    int i = 0;
    public double area( );
}

interface Color {
    int i = 0;
    public int red( );
    public int green( );
    public int blue( );
}

class Test implements Shape, Color {
    int myI = i;
    public Test( ) { }
    public double area( ) {return -1;}
    public int red( ) {return 0;}
    public int blue( ) {return 1;}
    public int green( ) {return 2;}
}
```

What is printed?

- a. Only methods used by a class that implements an interface must actually be defined.
- b. This program is illegal because Java prohibits multiple inheritance, including implementing multiple interfaces.
- c. All of “area”, “red”, “green” and “blue” must be defined in Test because Test implements interfaces that declare them.

## 18 C++ Question

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

private:
    void print( ) {
        cout << "B::print( ) ";
    }

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

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

    void print() {
        D::print( );
    }
};

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

    D *d = new D( );
    B *b = (B*) d;;

    b->print(4);
    d->print(4);

    return 0;
}
```

What is the most correct answer?

- a. B::print(int)B::print( ) is printed.
- b. The program is illegal because both print functions in B are private and "print( )" in B cannot be accessed in the call "d->print(4);"**
- c. The program is legal because although "print( )" in B is private, the class D can access it since it inherits from B.

## 19 Threads Question:

Assume that variables X and Y are initialized to 0 when the program begins. Consider the following code executing in *Thread 1* and *Thread 2*. There is no synchronization in the program but you may assume all statements in the thread execute in the order written, i.e. the execution is sequentially consistent.

Thread 1	Thread 2
X = 1;	X = 2;
Y = 2;	Y = 1;

Which answer is most correct about what the values of X and Y can be after the code in Thread 1 and Thread 2 executes?

- a. "X=1, Y=2"
- b. "X=2, Y=1"
- c. "X=1, Y=1"
- d. "X=2, Y=2"

e. All of the above.