CS 535 Object-Oriented Programming & Design
Fall Semester, 2000
Doc 14 CS 535 Past Exams
Contents

Here are the past exams given in this course. There have been a number of major changes in this course. Some of the offerings have covered C++, some have covered Java, in 1999 no language was covered, this semester Java was sort-of covered. This means that this semester's course differs from past offerings. If you wish to see how this semester course differs from past semesters the lecture notes for the last four years are on the web. Given that the current course differs from past courses exam on November 15 will be somewhat different from past exams. The exams below should help give you some insight to my exam style.

Some of the questions below are on material not yet covered this semester. Since my lecture notes are publicly available (and you have been attending lectures) you should be able to determine which questions are relevant.

I do not provide answers to these questions. The value of the questions is determining the answer, not in memorizing my answers.

In making up an exam I first list all the topics covered. I try to make the number of questions on a topic to be proportional to amount of time spent on the topic and the importance of the topic. Given that the exam is only 75 minutes long, one can not cover all material in 12 weeks of classes.

Spring 1999
Midterm

Answer 8 of the following 9 questions. Indicate which 8 you wish to be graded. If you do not indicate the 8 to be graded, the 8 that give you the lowest score will be selected for you.

1. We read two texts that covered coupling: the one chapter in Berard's Essays on Object-oriented Software Engineering, and Riel's Heuristics text. The Heuristics text defines export coupling and overt coupling. Define one of those types of coupling. What term does did Berard use for that type of coupling?

2. We have two classes A and B. What relationship between A and B do the following phrases indicate. Please indicate the role of A and B in the relationship. For example: A is the parent of B.

a. A is-kind-of B

b. A is-analogous-to B

c. A has-a B

d. A depends-upon B

3a.What is a god class?

3b. Give two heuristics for detecting god classes. Briefly explain how the heuristics help detect god classes.

4. What are the major steps in the Wirfs-Brock design process?

5. Explain one of the following types of cohesion: Logical, Temporal, Functional, and Sequential. Give an example.

6. Define the following terms:

a. Subsystem

b. Scenario

c. Primitive method

d. Abstract class

7. Briefly describe the observer pattern. Explain how it reduces coupling.

8. What problems can arise from successful cutting and pasting of code segments in a project? By successful cutting and pasting means the original code works, and the code also works correctly in its new location. Briefly explain your answer.

9. What is a contract in the Wirfs-Brock design process? Give an example.

Fall 1997
Midterm

1a. Explain the difference between static and dynamic access of methods.

1b. Give one situation where a reference to a non-static method will be resolved statically in Java.

2. Define overloading a method.

3a. List the four types of access levels that a field of a class can have in Java.

3b. Explain two of the access levels. That is given a field "foo" in class Bar with access level of type X, where can field "foo" be accessed? Where is field "foo" hidden from view?

4. Explain the differences between super and this in Java.

5. Show how to declare and create an triangular array in Java. That is first row of the array will be one column long, the second row will be two columns long, the k'th row will be k columns long.

6. Class Child is a subclass of class Parent. Class Parent has a method called foo. If class Child is to override the method foo, what restrictions does the Parent's foo method place on the Child's foo method in Java?

7. Explain how the finally block operates in a try-catch.

8. A blank final field is a field that is declared final, but not assigned a value in its declaration. Where can the field be assigned a value in Java?

9. What will be the output of the following Java program? Be careful!

```
class Parent
    {
    static
        {
        System.out.println( "Parent static" );
        }
    }

class Child extends Parent
    {
        {
        System.out.println( "Child non-static" );
        }

    public Child()
        {
        System.out.println( "In child constructor" );
        }

    static
        {
        System.out.println( "Child static" );
        }

    public static void main( String[] arguments )
        {
        new Child();
        new Child(); //Yes I mean to repeat it here
        }
    }
```

Final

1a. List the different access levels for fields and methods in Java.

b. Explain the differences between the access levels in Java? That is given a class A with a method B where can the method B be accessed and where can it not be accessed if it has access level X. (Answer for X equal to each access level listed in a.)

2a. How is an exception raised in Java?
b. What are the different types of exceptions in Java?
b. How is the exception handler found in Java. Give a separate answer for each type of exception.

3a. What is polymorphism?
b. Explain how to achieve polymorphism in Java.

4a. What is an overloaded method in Java?
b. What is an overridden method in Java?
c. Give an example when you can not overload a method in Java.
d Give an example when you can not override a method in Java.

5a. What is a NaN?
b. If A and B are fields that have been assigned a NaN (A = NaN, B = NaN) what are the results of the following comparisons?

A < B
A == B
A > B

6. We have two classes A and B. What relationship between A and B do the following phrases indicate. Please indicate the role of A and B in the relationship. For example: A is the parent of B.

a. A is-kind-of B
b. A is-analogous-to B
c. A has-a B
d. A depends-upon B

7. True or False. A Java class A has a final field F. All objects created from class A must have the same value for F.

True or False. An access of a non static method is always resolved dynamically.

True or False. If a child class constructor does not explicitly call it's parent's constructor, then the parents constructor is not called when creating a child object.

8. a. What is a scenario?
b. What do we use scenarios for in object-oriented design?

9. Identify the classes in the following description of a refrigerator. List at least one responsibility for each class.

> A refrigerator has a motor, a temperature sensor, a light and a door. The motor turns on and off primarily as prescribed by the temperature sensor. However, the motor stops when the door is opened. The motor restarts when the door is closed if the temperature is too high. The light is turned on when the door opens and is turned off when the door is closed.

Fall 1996
Midterm

1) a) We have a class A which has a protected field P. Explain where the protected field P is accessible and where it is not accessible.

b) We have a class A which has a field N, with no explicit access level given. Explain where the field N is accessible and where it is not accessible.

2) a) What is a NaN?

b) Under what conditions will a NaN be generated in Java?

3) What is a constructor? When are they called?

b) What is the role of a finalize method? When is it called?

4) What are the differences between an abstract class and an interface in Java?

5) a) What makes an exception check or unchecked?

b) How can a Java program treat an unchecked exception differently than a checked exception?

6) Let a child and parent class both have a method, public void foo(), with the same signature. The rules for determining which foo() is executed at run time can be summarized as: "overridden methods are resolved dynamically". Explain why hidden fields can not be resolved dynamically.

7) What is super? Give an example of when you would use “super”.

8) We have two classes A and B. What relationship between A and B do the following phrases indicate. Please indicate the role of A and B in the relationship. For example: A is the parent of B.

a) Is-kind-of

b) is-analogous-to

c) has-a

9) True or False. A Java class A has a final field F. All objects created from class A must have the same value for F.

True or False. An access of a non static method is always resolved dynamically.

True or False. The average number of fields per class should be less than 6.

True or False. If a child class constructor does not explicitly call it's parent's constructor, then the parents constructor is not called when creating a child object.

10) What is the result of compiling and if they compile executing the following programs?
a)

```
class  Test
    {
    public static void A()
        {
        int  array[] = new int[5];
        try
            {
            System.out.println( "Start Try"  );
            array[  10  ]  =  1;
            }
        catch (  Exception error  )
            {
            System.out.println( "Exception " );
            throw error;
            }
        finally
            {
            System.out.println(  "Final Block"  );
            return;
            }
        }

    public  static  void  main( String  args[]  )
        {
        try {  A();  }
        catch  (ArrayIndexOutOfBoundsException error )
            {
            System.out.println(  "In Catch"  );
            }
        System.out.println(  "After try in main"  );
        }
    }
```

b)

```
class  Test
    {
    String message = "Global";

    public static void A( String message)
        {
        message = message + "End";
        }

    public  static  void  main( String  args[]  )
        {
        String message = "Main";
        A( message );
        System.out.println( message );
        }
    }
```

c)

```
class Parent {
    public Parent() {
        System.out.println( "Parent" );
    }

    public Parent( double value ) {
        System.out.println( "Parent value "  + value );
    }
}

class  Child extends Parent {
    public Child( String message ) {
        System.out.println( "Child " + message );
    }

    public Child( double value ) {
        System.out.println( "Child value " + value );
    }

    public  static  void  main( String  args[]  ) {
        Child A = new Child( 5.5 );
        Parent B = new Child();
    }
}
```

Fall 1995
MidtermExam

1) List the different ways to implement implicit type conversion in a class.

2) If a class has a pointer as a data member, what items should be part of the class to prevent various pointer problems.

3) a) True or False. A copy constructor does not have initialization phase.

b) Give two different types of items that must be initialized in the initialization phase of a constructor.

4) Explain the object-oriented meaning of the following terms.
a) client-server

b) responsibilities

c) scenario

5) a) How do the variables A and B differ?

```
char  *const  A  =  "Hi";
const  char*  B  =  "Hi";
```

b) Explain the problems with the following uses of C and D

```
const  char*  C  =  "hi mom";
C[3] = 'a';

char  *const  D  =  "hi mom";
D = "hi dad";
```

6) We have two entities A and B. What relationship between A and B do the following phrases indicate. Please indicate the role of A and B in the relationship. For example: A is the parent of B.

a) A is a B

b) A is part of B

c) A uses B

d) A is kind of B

What is the output of the following programs?

7)#include <iostream.h>

```
void functionA( int X, double Y) { cout << "First function\n";};
void functionA( char X, float Y ) { cout << "Second function\n";};

main() {
    char Me = 'a';
    int You = 5;
    functionA( Me, You );
}
```

8)

```
#include <iostream.h>

class Container {
    public:
        int value;

        Container( int amount ) { value = amount;
                    cout << "Value " << value << endl; };

        ~Container() { cout << " You just killed: " << value << endl; };
};

class ExamQuestion {
    public :
        Container data;

        ExamQuestion(int A) : data(A) { cout << "New Object\n";};

        ExamQuestion( const ExamQuestion& X ) : data(X.data.value + 10)
                        { cout << "Special\n"; };
};

void TrickyPart(ExamQuestion why) {
    ExamQuestion PartB = why;
    cout << "After PartB\n";
}

void main() {
    ExamQuestion Answer(1);
    cout << "Call TrickyPart\n";

    TrickyPart(Answer);
    cout << "end" << endl;
}
```

What is the result of compiling and running (if they compile) the following programs.

9)

```
class Test {
    private :
        float data;

    public :
        void setData( int& value) { data = value;};

        static float getData()   { return data;};

        Test( int value ) : data( value ) {};
};

#include <iostream.h>

void main()
{
    Test me = 10.5;
    cout << me.getData() << endl;
}
```

10)

```
int A = 10;

float functionB( int A, char B = 5, float C ) {
    return ::A + B + C;
}

#include <iostream.h>

main() {
    int A = 2;
    float X = 11.1;
    cout << functionB( A, X );
}
```

Final Exam

1) A data member can be declared public, private, or protected. A base class can be inherited via public, private, or protected inheritance. Explain the effects of private and protected inheritance on the access derived class has of the public and protected data members of the bass class.

2) A base class has a data member called "Foo". The derived class also implements a data member of the same name "Foo". Explain how a function member of the derived class can access the base's data member "Foo".

3) What problem does multiple inheritance cause? How does C++ solve the problem?

4) What are the major steps in the Wirfs-Brock design process?

5) What is the difference between a superclass and a subsystem?

6) a) What is an abstract class?

b) How do you indicate an abstract class in C++?

7) Given classes A and B, what is the most likely relationship between A and B indicated by the phrases:

   a) A is a kind of B

   b) A uses B

   c) A is similar to B

   d) A depends on B

   e) A has knowledge of B

8) What is the output of the following program?

```
#include <iostream.h>
class ClassA {
    public:
        virtual void Print() {   cout << "Class A \n"; }
};

class ClassZ :  public ClassA  {
    public:
        void Print() { cout << "Class Z \n";}
};

main()
{
    ClassZ* Zpointer = new ClassZ;
    ClassA* Apointer  = new ClassA;
    ClassA  Avariable;

    Zpointer->Print();
    Apointer->Print();

    Apointer = Zpointer;
    Avariable = *Zpointer;

    Apointer->Print();
    Avariable.Print();
}
```

9) What is the output of the following program?

```
#include <iostream.h>
class A  {

    public :
        A(int a = 0)    {cout << a << "A\n" ;}
};

class B :  public A    {

    public :

        B(int b = 0) : A(b - 1)   {cout << b <<"B\n" ;}
};

class C :  public A {

    public :
        B DataMember;

        C(int c = 0) : A(c-1)    {cout << c << "C\n" ;}
};


class Test :  public C, public B, public A  {

    public:
        C State;
        Test() : State(10),  C(50),  B(40),  A(30) { cout << "Test\n"; };
};


main () {
    Test me;
    cout << "Done" << endl;
};
```

Fall 1994
MidtermExam

1. What is the difference between private and protected class members.

2. What is “this”. Give an example of when you would use “this”.

3. What are the initialization and the assignment phases of a constructor? What is each used for?

4. Having a pointer as a class data member can lead to some undesirable side effects. What should you add to a class to avoid these side effects?

Each of the following programs illustrates one or more concepts or issues in C++. What is/are those issues? What is the result of compiling and running each program? (Some programs may not compile.) Explain your answer. Assume iostream.h is included in each program. Any minor syntax errors are typos.

5a)

```
void you( long,  long ) {cout << "long"};
void you( double, double ) {cout << "double"};

main()
{
    int got, me;
    dont( got, me );
}
```

5b)

```
main()
{
const  char*  what  =  "Is This";
pc = "Interesting"                cout << *what;
what[3] = 'a';                  cout << *what
}
```

6a)

```
int& Now()  {
    int Where = 1;
    return Where ;
}

main() {
    int Where;
    Where= Now();
    cout << Where;
}
```

6b)

```
class Where {
    public:
        int here;

        int me() const;
        int you() const;
};

int Where ::me() const {
    this->here = here + 5;                    return this->here;
}

int Where ::you() const {
    return here + 5;
}

int OhNo(const Where* Now)
{
    return Where->you();
}


main()
{
    Where* IsIt = new Where;
    cout << OhNo(IsIt);
};
```

7a)

```
class Top
{   public:
        int a;
        Top(int x) { a = x;}

};

class Bottom : public Top
{   public:
        int b;
        Bottom() { b = 0; a = 0;}          };


void main()
{
    Bottom barrel;
    cout << barrel.a << barrel.b;
}
```

7b)

```
class Top
{
    public:
        Hat(char *string) {    cout << "Top";}
        Hat(float a) {   cout << "Top Too";}
};

class Bottom :  public Top
{
    public:
        Hat(const int a) { cout << "Bottom";}
        Hat(float a) {   cout << "Bottom Too";}
};

main()
{
    Bottom* Rung;
    Rung->Hat(5.5);
    Rung->Hat("cat");

}
```

8.

```
class Trouble
{   public:
    Trouble() { cout << "Start Trouble";}
    ~Trouble() { cout << "End Trouble";}
};

class Big : public Trouble
{   public:
    Big() { cout << "Start Big";}
    ~Big() { cout << "End Big";}
};

class Dont
{   public:
    Big deal;
    Dont() { cout << "Start Dont";}
};

main()
{
    Dont DoThisToMe;
}
```

Final Exam

1. Give the output of the following program.

```
#include <iostream.h>

class Top {
public:
    virtual void MyMemory() { cout << "I forget" << endl;};
    void Disk()     { cout << "Space" << endl;};
    void Erased()  { cout << "For good" << endl;};
    void ThisExam() { Erased(); MyMemory(); };
};

class IsOver : public Top {
public:
    void MyMemory()   { cout << "Gone" << endl;};
    void Disk()         { cout << "Slipped" << endl;};
    void virtual Erased()     { cout << "Rubbed out" << endl;};
};
void main()
{
    Top* Hat = new IsOver;
    Hat->MyMemory();
    Hat->Disk();
    Hat->ThisExam();

    Top Dog = *(new IsOver);
    Dog.MyMemory();
    Dog.Disk();
    Dog.ThisExam();
}
```

Mini-Adventure (MA). The game MA consists of a maze of rooms. A room consists of four walls and one or more doors. Doors can be either open, close, or closed and locked. Some locked doors are opened by a key, others are opened by a magic phrase. Each key opens only one door, except for the master key. Doors lead into other rooms. Rooms are rectangular shaped. Long skinny rooms are called hallways. Rooms may contain gold coins, keys, and/or scrolls with magic phrases. When picked up, scrolls self destruct in a fixed amount of time. This time differs for each scroll. The goal of the game is to explore the rooms and collect as much gold as possible. A player can move around using commands as: goUp, goDown, goLeft, goRight, gotoX where X can be Door, key, scroll, and pickUpX where X can be key, scroll, and gold. Each time a game is played a new maze is created and the player starts in a randomly selected room. A player can see on the screen only the contents of the current room and the outline of rooms they have already been in.

2 a. Give a list of potential classes for the MA game.

b. Pick a class from part a. and give a list of responsibilities for the class,

3. Give a detailed scenario for the game MA involving the class in problem 2b.

4. a. What is an abstract class?
b. What is the purpose of abstract classes in object-oriented programming?

5. Define the following terms in object-oriented programming.
a. client-server

b. contracts

c. protocol

6. What is permitted and what is not permitted by the law of Demeter?

7. a. Why do some object-oriented experts recommend programmers avoid case (and if) statements?

b. Give two alternative mechanisms to case (and if) statements.

Fall 1993
Final Exam

1) Explain the object-oriented meaning of the following terms.

a) abstraction
b) encapsulation
c) information hiding

2) What problem(s) does multiple inheritance have that single inheritance does not?

3) a) What is polymorphism?
b) Give an example of polymorphism in C++.

4) Why is polymorphism important in object-oriented programing?

5) We have two classes A and B. What do the following relationships indicate about the two classes:

a) A is-a B
b) A has-a B
c) A is-kind-of B
d) A is-analogous-to B
e) A depends-upon B

6) a) What is a scenario?
b) What role(s) do scenarios play in object-oriented design.

7) Identify the classes in the following description of a refrigerator. List at least one responsibility for each class.

> A refrigerator has a motor, a temperature sensor, a light and a door. The motor turns on and off primarily as prescribed by the temperature sensor. However, the motor stops when the door is opened. The motor restarts when the door is closed if the temperature is too high. The light is turned on when the door opens and is turned off when the door is closed.

8) a) What is a subsystem?
b) What is the purpose of a subsystem in the Wirfs-Brock object-oriented design process?
c) Why are subsystems found toward the end of the process, rather than at the beginning?

9) What is a contract? Give an example.

10) Briefly describe the steps in the Wirfs-Brock object-oriented design process.

11) What is a virtual function in C++? How does a virtual function operate?
How is inheriting a virtual function different than inheriting a normal functions members?