

You May Use Your C++ Book For This Exam

1. Create a class to hold a variable-sized array of integers. This object will require a size component and a data component. The data component should be a pointer to an integer. Show the constructor, and destructor for this object, in addition to the components.
2. Create a Polymorphic type for geometric objects. There are three types of objects, Rectangle, Circle, and Triangle. For Rectangle and Triangle, we keep width and height. For Circle we keep radius. We want to be able to create a linked list of these objects, and add up the total area using the following sort of loop.

```
Total = 0;
for (CGeom * Temp = HeadOfList ; Temp != NULL ; Temp=Temp->Next)
{
    Total += Temp->Area( );
}
```

3. Create a Vector class that allows for the operations Add, Subtract, and Dot Product. The size of each vector can be arbitrary. When adding, subtracting or dot-producting vectors of different sizes, the missing elements in the shortest vector should be zero.
4. Show the object diagram of a family. Include parents, children, and grandparents.
5. Show the object diagram of the UNIX/DOS/Windows file system (which ever you prefer.)
6. Give an object diagram of a binary search tree, and show the implementation of the classes.

Application Development Pretest 2

1. You are building a simulation of a car. Show the class diagram and the state diagram.
2. For the car simulation, break everything into modules and give the module diagram.
3. You are going to design a replacement for Microsoft Office. Describe the staffing requirements. Make a list of the different kinds of people you need, and describe the duties of each.
4. Describe the SortAnimation Object and how it works. How does the Sort Animation Template speed up development?
5. Describe the components of FHDL, and what they do.

1. Create a template class for a generic dynamically sized array.
2. Create a template for a minimum function.
3. Create a foundation class element for storing an arbitrary amount of text.
4. Critique the class structure of the data acquisition project. Suggest something better
5. Discuss the class structure of the DGL project.

1. Create a class to hold a variable-sized array of integers. This object will require a size component and a data component. The data component should be a pointer to an integer. Show the constructor, and destructor for this object, in addition to the components.

2. Create a Polymorphic type for geometric objects. There are three types of objects, Rectangle, Circle, and Triangle. For Rectangle and Triangle, we keep width and height. For Circle we keep radius. We want to be able to create a linked list of these objects, and add up the total area using the following sort of loop.

```
Total = 0;
for (CGeom * Temp = HeadOfList ; Temp != NULL ; Temp=Temp->Next)
{
    Total += Temp->Area( );
}
```

3. Create an animal class (or set of classes) that contains information for squirrels, rattle snakes, and owls. Include a function called “CanFly” that returns true if the animal can fly. I should be able to call this function for any object in any of the classes you create. (There are several correct ways to do this.)

4. Suppose you want to create a computer simulation of a CD jewel case, possibly with a CD inside. Show the class diagram for the required objects. You should have *at least* three objects.

5. Suppose you want to create a computer simulation of a coke machine, and of the drinks inside the machine. Show the class diagram for the required objects. You should have *at least* three objects.

Application Development Pretest 2

1. You are building a simulation of a car. Show the class diagram and the state diagram.
2. For the car simulation, break everything into modules and give the module diagram.
3. You are going to design a replacement for Microsoft Office. Describe the staffing requirements. Make a list of the different kinds of people you need, and describe the duties of each.
4. Describe the SortAnimation Object and how it works. How does the Sort Animation Template speed up development?
5. Describe the components of FHDL, and what they do.

1. You want to create a class called `OrderedPair` that has two elements of the same type. You don't know, beforehand, what the type is. Create a template for this class. Add two functions to the class called `GetFirst` and `GetSecond`. These two functions will return the first and second elements of the class, respectively.

2. You want to create a function that computes the square of a number (x^2), but you don't know what kind of numbers you want to use it for. Create a template for this function, and show how it would be used to find the square of a long and the square of a float. Your function should be named "sq".

3. Create a foundation class element for storing the title of a document. The title of a document will be a C string (char *). There should be two functions, one that converts the C string to all upper case, and one that converts the string to all lower case. Use the following functions to convert characters from one case to another. If these functions are applied to a character, such as '5', that has no upper case, then the character is returned unchanged.

char *toupper*(char x);

char *tolower*(char x);

4. Discuss the merits of a chain of derived objects such as that exhibited by the data acquisition project. Contrast this approach with that of multiple inheritance.

5. Answer the following questions about the DGL project.
 - a. What is the objective of the DGL system?
 - b. How are classes and objects used to distinguish productions?
 - c. How is polymorphism used?

1. Create a class that will be used to create binary trees. Your class will describe one element of the tree. Each element of the tree has a right child, a left child and an integer value. Create the constructor and the destructor for this class. When the destructor is called, it should delete the entire tree, not just the current element.

2. Create a Polymorphic type for batches of foreign and domestic currency. Each batch of currency will contain the total amount of currency in the batch. There will be four different types of currency, Dollars, Pounds, Marks, and Francs. A Pound is worth 1.5 Dollars, a Franc is worth .2 Dollars, and a Mark is worth .75 Dollars. The dollar value *must be calculated*, it is not stored in the type. (The type for Pounds will contain NumberOfPounds, the type for Francs will contain NumberOfFrancs, and so forth.) There will be many batches of currency of each type. The batches will be contained in a linked list. Construct your classes so the following loop will correctly add up the total Dollar value of all currency.

```
Total = 0.0;
for (Currency * Temp = HeadOfList ; Temp != NULL ; Temp=Temp->Next)
{
    Total += Temp->DollarValue( );
}
```

3. Create a class for two-element vectors. Create two overrides for the operator $+=$, one that allows one vector to be added to another, and one that allows an integer to be added to all elements of the vector. Adding (1,2) and (20,40) should yield (21,42), adding (1,2) and 6 should yield (7,8).

4. Suppose you want to create a computer simulation of a stapler, possibly with staples inside. Show the class diagram for the required objects. You should have *at least* three objects.

5. Suppose you want to create a computer simulation of a chessboard and the pieces. Give a class diagram for this. You should have at least three classes.

1. Create a class to represent polygons. Each polygon will consist of an array of points. Each point will have an x and a y coordinate. The CPolygon class must have an array of points and a counter indicating the number of points in the array. Create a destructor and two constructors. The default constructor will create a polygon with four points. The other constructor will have one parameter, an integer giving the number of points in the polygon. Don't worry about accessor and mutator functions, but make things private if they need to be private.

2. Create a Polymorphic type for Poker Chips. You must have separate classes for white red and blue chips. A white chip is worth “one chip,” a red chip is worth “five chips,” and a blue chip is worth “ten chips.” You must have one class for each type of chip. Each object represents a stack of chips of one color. For each type of chip, you must keep a count of *the number of actual chips in the stack*. For example, if you have two red chips, the counter will contain 2 not 10. We want to be able to create a linked list of chip objects, and add up the total number of white-equivalent chips using the following loop.

```
WhiteTotal = 0;
for (CChip * Temp = HeadOfList ; Temp != NULL ; Temp=Temp->Next)
{
    WhiteTotal += Temp->WhiteCount( );
}
```

3. Create a set of classes to represent vehicles. Each vehicle will have at least one variable representing the top speed of the vehicle in miles per hour. Your vehicle classes should include, Ferrari's (a car), Sailboats, and a single-engine Cessnas (an airplane). Give the class diagram of your set of classes. You will need to use inheritance.

4. Suppose you want to create a computer simulation of a desk-top telephone. Give the class diagram using UML notation. (Model the device itself, not phone-calls.)

5. Create an object-oriented model of a bicycle. You want to model the parts and their relationships. Your model should have at least five classes. Give the UML diagram for your model.

Name: _____

1. Create a polymorphic type to represent a set of books that a book publisher wishes to publish and sell. There are three different types of books, Text-Books, Foreign Novels, and Picture Books (more commonly known as coffee-table books). The cost of publishing a text book is \$.04 per word, plus \$.15 per figure. The cost of publishing a picture book is \$2.00 per photograph. The cost of publishing a Foreign Novel is \$.07 per word plus \$10.00 per page for the translation. **Make the following loop work!**

```
TotalPublishingCost = 0.0;
for (CBook * Temp = BookHead ; Temp != NULL ; Temp=Temp->Next)
{
    TotalPublishingCost += Temp->BookCost();
}
```

2. Suppose you are building a simulation of a roller coaster. Create a reasonable set of classes for this simulation.

3. The following is a set of classes that will be used in the simulation of a radio station. Create a class diagram for each type of relationship.

Janitor	Compact Disk	Salesman
Microphone	Parking Lot	Reporter
Station Manager	Personnel	Radio Station
News Department	Transmitter	Studio
Office	Electronic Equipment	Building
News Program	News Wire	Special Bulletin
News Announcer	Advertising Department	Recording Tape
Tower	The Don Bailes Show	Disk Jockey

4. The following is a set of classes describing a VCR. You may assume that the VCR is plugged in, and that the power in the house is on. (The VCR itself is not necessarily on.) Identify at least eight cases of static and dynamic behavior. You must have at least one of each. Identify the class that will have this behavior. If a behavior is dynamic, indicate the conditions under which it will happen. Don't use If-Then-Else conditions. "Power turns on" and "Power turns off" are two different behaviors.

Record Button	Program Button	Button
Play Button	Rewind Button	Tape Door
Pause Button	Fast-Forward Button	Read/Write Head
Time Display	Mark-Tape Button	Tape-Feed Mechanism
Tape-In-VCR Indicator	Tracking button	Stop Button

5. The following classes are used in the simulation of a train station. The idea is to model track usage under various schedules of train arrivals. We want to model different customer loads, and we also want to determine the optimum length of each train. All cars are the same size, and there is no locomotive. (It's an electric train.) Identify several responsibilities (at least 6), and assign them to the appropriate class. (Where do the customers come from?)

Ticket Window	Train	Car
Ticket Agent	Customer	Engineer
Schedule	Street Entrance	Turnstile
Timer	Track	Platform
Ticket	Conductor	Vending Machine

Name: _____

1. You have an application that needs to record a list of numbers. These numbers are all in long-integer format. The numbers will be created and added to the list from many different parts of the program. Near the end of the program it will be necessary to retrieve the numbers, one at a time, in the order in which they were recorded. Using the knowledge you have acquired in this course, create a set of classes to implement these functions. Make sure that you use the private/public/protected attributes properly.

2. When creating an object-oriented design, there are several steps that one must go through. The first of these is identifying the objects. List the others, and give examples. Use the “Towers of Hanoi” case study for your examples.

3. You have just landed the job of your dreams. Your boss asks you to create an object-oriented model of a car. You realize that this could mean several different things, but unfortunately, you don't have any additional details. Does this mean driving a car? Manufacturing a car? What? You decide to create three different object oriented models, breaking the car down into objects in three different ways. Create three different object diagrams representing three different types of relationships. Each relationship should be a reasonable way of looking at a car, but different from the other two. Each relationship should have from 3 to 5 classes.

4. Discuss the case study for evaluating expressions. How are the expressions represented? How are they evaluated? It is not necessary to show the implementation of every class, but show me enough of them to prove that you understood the example.

5. You are now a professional programmer, and you have just come back from your first “thinking outside the box” seminar. Your boss hands you a set of requirements for a new program. The requirements read as follows:

It has to run in 2.4 milliseconds on either UNIX or MS Windows, preferably both. The red button should open the valve, but only after proper confirmation. When the valve is closed improperly, an alarm should sound. Timing is important, and a 2.7 microsecond clock with a wireless reset should be used for that purpose. However timing is not so important that it needs to be considered continuously by all parts of the program, since some of these are of a priority nature. When the buffer becomes full, it is ok to ignore further input until such time as the buffer becomes empty. Totals should be presented in a two-column format.

What is the first step in creating an object-oriented design for this program?