

Rotate screen



Fit screen



Enter Browser



CS304:Grand Quiz (Mid Term Exam)

Question # 20 of 30 (Start time: 12:25:56 AM, 27 June 2020)

Through _____, Objects communicate with outer world.

Select the correct option

- Encapsulation
- Interface
- Inheritance
- Polymorphism

CS304: Grand Quiz (Mid Term Exam)

Question # 13 of 30 (Start time: 02:52:07 AM, 2

Which of the following is a tangible entity?

Select the correct option


- Database
- Time
- Car
- Date of birth

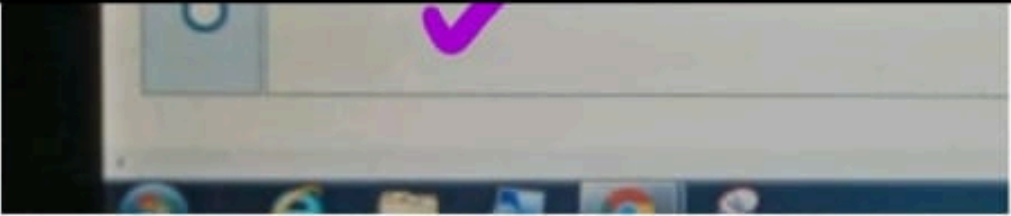
CS304: Grand Quiz (Mid Term Exam)

Question # 6 of 30 (Start time: 12:08:30 AM, 27 June 2020)

Entities in OO model are represented by:

Select the correct option

- Circle
- Square
- Lines
- Rectangle 



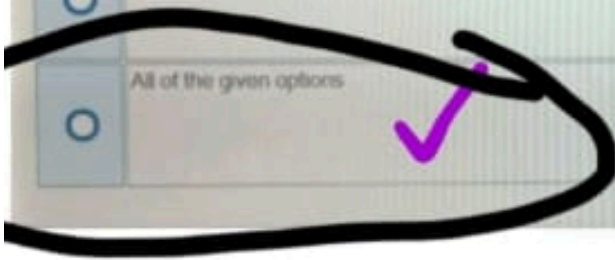
CS304: Grand Quiz (Mid Term Exam)

Question # 3 of 30 (Start time: 12:05:04 AM, 27 June 2020)

Which of the following is true about inline functions?

Select the correct option

- It is used by compilers to improve efficiency of the program.
- Small size functions that need to be called many times during program execution are declared inline.
- Inline functions decrease the code execution time.
- All of the given options




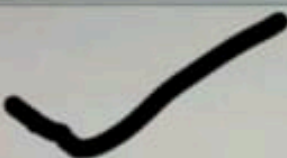


CS304:Grand Quiz (Mid Term Exam)

Question # 30 of 30 (Start time: 12:36:22 AM, 27 June 2020)

A child inherits characteristics from its _____.

Select the correct option

- Neighbor
- Parent  
- Sibling
- itself

CS304:Grand Quiz (Mid Term Exam)

Question # 27 of 30 (Start time: 12:34:07 AM, 27

Polymorphism makes the system:

Select the correct option

- reusable
- flexible
- faster
- All of the given options

Question # 25 of 30 (Start time: 12:32:11 AM, 27 June 2020)

which of the following is part of the class diagram?

Select the correct option



- Class Name
- Attributes
- Behaviors
- All of the given options



Question # 12 of 30 (Start time: 12:52:30 AM, 27 June 2020)

Subtyping means that derived class is behaviorally _____ with the base class.

Select the correct option

- Same
- Compatible  
- Different
- Incompatible

CS304:Grand Quiz (Mid Term Exam)

Question # 9 of 30 (Start time: 12:12:18 AM, 27 June 2020)

A post-fix unary operator is implemented in C++ using member function with:

Select the correct option

- 1 dummy char argument
- 1 dummy int argument
- 1 dummy float argument
- None of the given options

CS304:Grand Quiz (Mid Term Exam)

Question # 7 of 30 (Start time: 12:09:53 AM, 27 June 2020)

_____ is the reverse process of Generalization.

Select the correct option

- Aggregation
- Inheritance
- Dependence
- Specialization

CS304:Grand Quiz (Mid Term Exam)

Question # 4 of 30 (Start time: 12:06:14 AM, 27 June 2020)

Inheritance is shown by

Select the correct option

- Open head arrow
- Filled arrow
- Single Line
- Bi-directional

CS304:Grand Quiz (Mid Term Exam)

Question 7 29 of 30 (Start time: 01:11:05 AM, 27 June 2020)

A class defined within another class is:

Select the correct option

Nested Class

Inheritance

Containership

Encapsulation



CS304:Grand Quiz (Mid Term Exam)

Question # 19 of 30 (Start time: 03:00:04 AM, 27 June 2)

The other name of specialization is _____

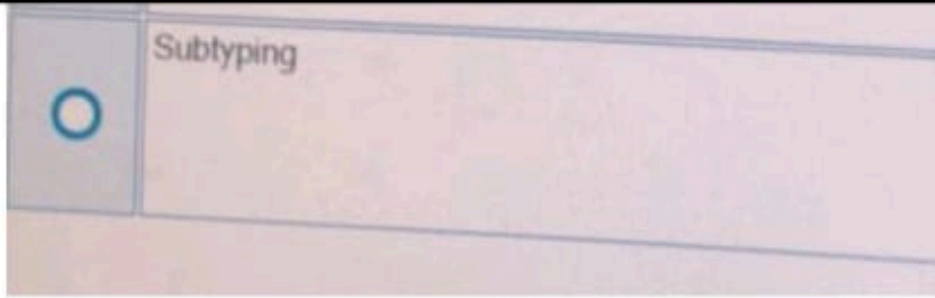
Select and correct option

Restriction



Dependency

Subtyping



Smart time: 01:17:27 AM, 27 June 2020

means that different objects can behave in different ways for the same message.

Select the correct option

<input type="radio"/>	Specialization
<input type="radio"/>	Generalization
<input checked="" type="radio"/>	Polymorphism
<input type="radio"/>	Abstraction

A screenshot of a quiz question. At the top, there is a timestamp: "Smart time: 01:17:27 AM, 27 June 2020". Below it, a text box contains the question: "means that different objects can behave in different ways for the same message." Underneath the text box is the instruction "Select the correct option". There are four radio button options: "Specialization", "Generalization", "Polymorphism", and "Abstraction". The "Polymorphism" option is selected, indicated by a purple checkmark and a black checkmark drawn over the radio button.


CS304:Grand Quiz (Mid Term Exam)

Question # 8 of 30 (Start time: 12:11:16 AM, 27 June 2020)

Compiler generated default constructor is called _____ and user written default constructor is called _____

Select the correct option

- Compiler-defined, Implicit
- Explicit, User-defined
- Explicit, Implicit
- Implicit, Explicit



SS304: Grand Quiz (Mid Term Exam)

Question # 29 of 30 (Start time: 12:35:48 AM, 27 June 2020)

In _____, a pointer or reference to an object is created inside a class

Select the correct option

<input type="radio"/>	Aggregation	✓
<input type="radio"/>	Composition	
<input type="radio"/>	Separation	
<input type="radio"/>	None of the given	✗



CS304-Grand Quiz (Mid Term Exam)

Quiz Start Time: 12

Question # 20 of 30 (Start time: 01:01:09 AM, 27 June 2020)

Suppressed form of class representation contains:

Select the correct option

- Operations
- Class Name  
- Attributes
- Interface Name



CS304 Grand Quiz (Mid Term Exam)

Question # 24 of 30 (Start time: 01:33:47 AM, 27 June 2020)

_____ is represented by a line with an unfilled diamond head towards the contained

Select the correct option

<input type="radio"/>	Inheritance
<input type="radio"/>	Association X
<input type="radio"/>	Aggregation ✓
<input type="radio"/>	Composition

CS304: Grand Quiz (Mid Term Exam)

Question # 11 of 30 (Start time: 04:32:32 AM, 27 June 2020)

_____ is creating objects of one class inside another class.

Select the correct option

- Association
- Composition ✓
- Aggregation
- Inheritance



CS304: Grand Quiz (Mid Term Exam)

Question # 25 of 30 (Start time: 04:50:24 AM, 27 June 2020)

_____ is set of functions of an object exposed to other objects

Select the correct option

- Interface
- Encapsulation
- Object Model
- Abstraction

A purple checkmark is drawn next to the "Interface" option, and a larger black checkmark is drawn to the right of the entire list.

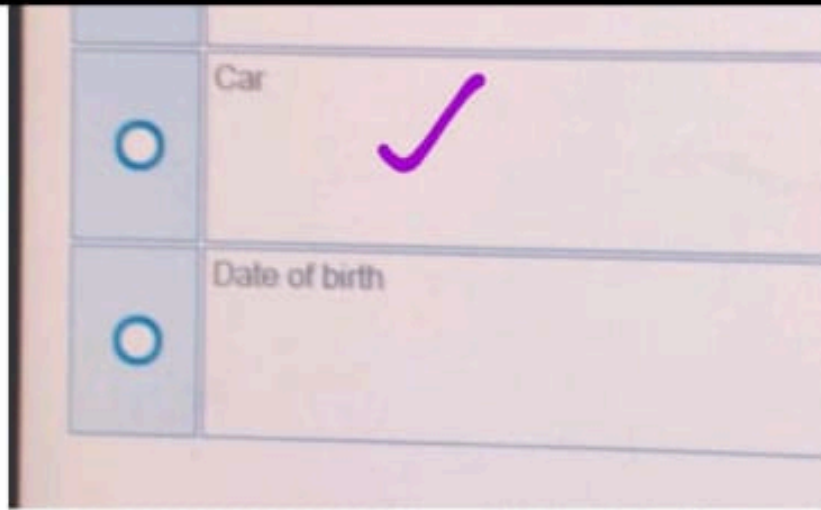
CS304:Grand Quiz (Mid Term Exam)

Question # 12 of 30 (Start time: 12:15:38 AM, 27 June 2020)

Member functions defined inside a class declaration are _____ by default.

Select the correct option

- Visible
- Public
- Private
- Protected



Question # 5 of 30 (Start time: 01:15:54 AM, 27 June 2020)

Which of the following will happen when a constant function tries to change the value of data members of the class?

Select the correct option

- Compiler will allow changing data members.
- Program will be executed successfully
- Compile time error will occur
- Run time error will occur ✓

CS304:Grand Quiz (Mid Term Exam)

Question # 12 of 30 (Start time: 12:15:38 AM, 27 June 2020)

Member functions defined inside a class declaration are _____ by default.

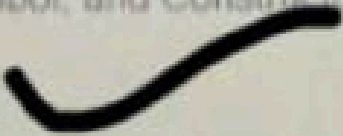

Select the correct option

- Visible
- Public
- Private
- Protected

Question # 1 of 30 (Start time: 01:11:46 AM, 27 June 2020)

How can we differentiate between constructors and destructors?

Select the correct option

- | | |
|-----------------------|---|
| <input type="radio"/> | Destructors have a return type but Constructors do not have return type |
| <input type="radio"/> | Destructors can't be defined by the programmer, but Constructors can be defined by the programmer |
| <input type="radio"/> | Destructors are preceded with a tilde (~) symbol, and Constructors are not preceded with any symbol |
| <input type="radio"/> | Destructors are exactly same as Constructors in syntax |
-  

Extension or subtyping is _____ approach.

Select the correct option



Bottom-up



Top-bottom



Right-Left



Left-Right

Question 2 of 30 (Start time: 12:38:02 AM, 27 June 2020)

In OOP we can achieve reusability through?

Select the correct option

- Inheritance
- Encapsulation
- Aggregation
- Association

Give an Answer

CS304:Grand Quiz (Mid Term Exam)

Question # 22 of 30 (Start time: 12:28:17 AM, 27 June 2020)



What is meant by multiple inheritance?

Select the correct option

- | | |
|-----------------------|--|
| <input type="radio"/> | Deriving a base class from derived class |
| <input type="radio"/> | Deriving a derived class from base class |
| <input type="radio"/> | Deriving a derived class from more than one base class |
| <input type="radio"/> | None of the given option |
- ✓
- ✓

_____ Operator is used to create objects at run time.

Select the correct option

- create
- new 
- dynamic 
- subscript

Hiding the implementation details makes program:

Select the correct option

<input type="radio"/>	Easy to understand
<input type="radio"/>	More complex
<input type="radio"/>	Provide more features
<input type="radio"/>	Logical



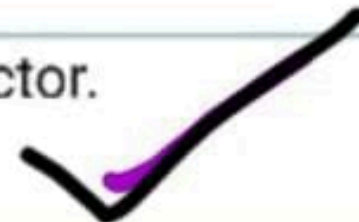
Question 7 of 30 (Start time: 12:46:06 AM, 27 June 2020)

Total Mar

Suppose a class does not have any constructor. What will happen when an object of this class is created?

Select the correct option

- Compiler will generate syntax error.
- Run time error will occur.
- Compiler will call implicit default constructor.
- None of the given options



Question 4 of 30 (Start time: 12:43:20 AM, 27 June 2020)

Aggregation between objects shows that _____

Select the correct option



Objects have strong relationship



Objects have weak relationship



Objects have no relationship



Objects have hierarchical relationship

CS304 Grand Quiz (Mid Term Exam)

Quiz Start Time: 12:35 AM,

Question 7 30 of 30 (Start time: 01:12:09 AM, 27 June 2020)

Suppose you have following C++ statements:

```
int oldValue=10;
```

```
int newValue = oldValue ++;
```

What will be the value of oldValue and newValue after executing above statements?

Select the correct option

 oldValue= 11 , newValue=11 oldValue= 10 , newValue=10 oldValue= 11 , newValue=10 oldValue= 10 , newValue=11

CS304: Grand Quiz (Mid Term Exam)

Question 7 of 30 (Start time: 12:50:35 AM, 27 June 2020)

Inheritance is shown by:

Select the correct option

- Open head arrow
- Filled arrow
- Single Line
- Bi-directional



Question # 17 of 30 (Start time: 12:21:50 AM, 27 June 2020)

_____ is represented by a line with a filled diamond head towards the composer object.

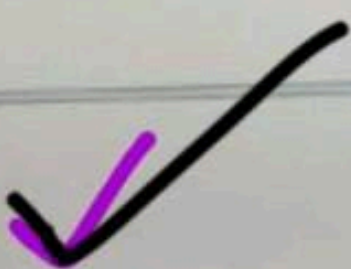
Select the correct option

- | | |
|----------------------------------|--------------|
| <input type="radio"/> | Inheritance |
| <input checked="" type="radio"/> | Composition |
| <input type="radio"/> | Association |
| <input type="radio"/> | Polymorphism |

Question # 28 of 30 (Start time: 12:34:53 AM, 27 June 2020)

Which of the following concept is more close to encapsulation?

Select the correct option

- | | |
|-----------------------|---|
| <input type="radio"/> | Exception Handling |
| <input type="radio"/> | Inheritance |
| <input type="radio"/> | Polymorphism |
| <input type="radio"/> | Information Hiding  |

Question # 24 of 30 (Start time: 01:05:11 AM, 27 June 2020)

Which of the following operator(s) take(s) one or no argument if overloaded?



Select the correct option

- ++
- *
- %
- All of the given choices

Question # 2 of 30 (Start time: 12:03:44 AM, 27 June 2020)

_____ represents "IS A" relationship.

Select the correct option



- | | |
|-----------------------|--|
| <input type="radio"/> | Simple Association |
| <input type="radio"/> | Inheritance   |
| <input type="radio"/> | Composition |
| <input type="radio"/> | Aggregation |

Question 7 of 30 (Start time: 12:41:46 AM, 27 June 2020)

Total Marks: 1

How can we identify classes from a given problem statement?

Select the correct option

- By extracting Nouns from problem statement 
- By extracting Verbs from problem statement
- By extracting Adjectives from problem statement
- All of the given options 

CS304: Grand Quiz (Mid Term Exam)

Question # 22 of 30 (Start time: 01:31:40 AM, 27 June 2020)

_____ is the relationship between the "part" object and the "whole" object.



Select the correct option

- | | |
|-----------------------|---------------|
| <input type="radio"/> | Inheritance |
| <input type="radio"/> | Association |
| <input type="radio"/> | Polymorphism |
| <input type="radio"/> | Composition ✓ |

Question # 17 of 30 (Start time: 01:26:54 AM, 27 June 2020)

_____ remain in memory even when all objects of a class have been destroyed.

Select the correct option

<input type="radio"/>	Static Variables	
<input type="radio"/>	Instance Variables	
<input type="radio"/>	Primitive Variables	
<input type="radio"/>	None of the given	

Question # 3 of 30 (Start time: 01:13:55 AM, 27 June 2020)

In order to make any function constant, keyword const is placed at the _____ of the parameter list.

Select the correct option

- | | |
|----------------------------------|-------------------|
| <input type="radio"/> | Beginning |
| <input type="radio"/> | Middle |
| <input checked="" type="radio"/> | End |
| <input type="radio"/> | None of the given |

Question 7 of 30 (Start time: 01:06:50 AM, 27 June 2020)

_____ is the weakest link between objects.

Select the correct option



Inheritance



Polymorphism



Composition



Simple Association



Question 7 of 30 (Start time: 12:57:51 AM, 27 June 2020)

Total Marks: 1

An object has attributes, operations, and _____.

Pg 12

Select the correct option

Unique Identity ✓

Behavior

Characteristics

All of these

Abstraction provides information according to _____.

Select the correct option

User perspective



Owner perspective

Random information

All information

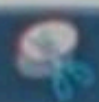
CS304:Grand Quiz (Mid Term Exam)

Question # 5 of 30 (Start time: 12:07:23 AM, 27 June 2020)

Composition is _____ relationship.

Select the correct option


- | | |
|-----------------------|------------|
| <input type="radio"/> | Is a |
| <input type="radio"/> | Is kind a |
| <input type="radio"/> | Has |
| <input type="radio"/> | Part-Whole |



Question # 16 of 30 (Start time: 12:55:59 AM, 27 June 2020)

Hiding the implementation details makes program:

Select the correct option

- Easy to understand 
- More complex
- Provide more features
- Logical

Question 7 of 30 (Start time: 01:08:04 AM, 27 June 2020)

Total Marks: 1

In which of the following way(s), we can inherit a class from another class in C++?

Select the correct option



Private



Public



Protected



All of the given options




Question # 23 of 30 (Start time: 12:29:26 AM, 27 June 20

Through interface we access object's _____

Select the correct option

<input type="radio"/>	States
<input type="radio"/>	Data members
<input type="radio"/>	Behaviour
<input type="radio"/>	Class



Question # 14 of 30 (Start time: 12:18:13 AM, 27 June 2020)

Student *ptr = new Student("Ali"); is an example of _____

Select the correct option

- Static memory allocation
- Dynamic memory allocation
- Destructor
- Constructor

Screenshot has been saved to/Pictures/Screenshot

83 / 98

PHASE 3

9AM-11PM Lahore

Question # 24 of 30 (Start time: 12:30:53 AM, 27 June 2020)

The relationship between container and contained object is known as _____.

Select the correct option

- | | |
|-----------------------|-------------|
| <input type="radio"/> | Inheritance |
| <input type="radio"/> | Association |
| <input type="radio"/> | Composition |
| <input type="radio"/> | Aggregation |

Question # 12 of 30 (Start time: 02:50:45 AM, 27 June 2020)

In an OOP model private information should only be accessible to its _____

Select the correct option

- | | |
|----------------------------------|---------------|
| <input checked="" type="radio"/> | Owner |
| <input type="radio"/> | Outside World |
| <input type="radio"/> | Objects |
| <input type="radio"/> | Class |

Question 7 of 30 (Start time: 12:53:13 AM, 27 June 2020)

Identify the common attributes of the given scenario:

Circle has color, vertices, and radius.

Triangle has color, vertices, and angle.

Select the correct option

Color, Vertices



Vertices, Radius

Color, Angle

Vertices, Radius

In _____, base class can be replaced by the derived class.

Select the correct option



Aggregation



Inheritance



Dependency



Extension

CS304: Grand Quiz (Mid Term Exam)

Question # 2 of 30 (Start time: 01:13:20 AM, 27 June 2021)

How many objects are involved in the Binary association?


Select the correct option

<input type="radio"/>	1
<input checked="" type="radio"/>	2
<input type="radio"/>	3
<input type="radio"/>	4

Question # 16 of 30 (Start time: 12:20:49 AM, 27 June 2020)

Consider the statement "room has chair" Which of the following type of association exists between room and chair?

Select the correct option


- Inheritance
 - Composition
 - There is no association
 - Aggregation
- 

Question 21 of 30 (Start time: 01:02:03 AM, 27 June 2020)

Total Marks: 1

"A fan has wings". Which type of relation exists between fan and wings in this sentence?

Select the correct option



 Aggregation Association Generalization Composition 

CS304: Grand Quiz (Mid Term Exam)

Question # 24 of 30 (Start time: 01:33:47 AM, 27 June 2020)

_____ is represented by a line with an unfilled diamond head towards the container


Select the correct option

- | | |
|-----------------------|---|
| <input type="radio"/> | Inheritance |
| <input type="radio"/> | Association  |
| <input type="radio"/> | Aggregation  |
| <input type="radio"/> | Composition |

Question # 4 of 30 (Start time: 04:24:29 AM, 27 June 2020)

Static Data Member is declared _____

Select the correct option

- | | |
|-----------------------|----------------------|
| <input type="radio"/> | Inside the class |
| <input type="radio"/> | Outside the class |
| <input type="radio"/> | Inside main() method |
| <input type="radio"/> | None of the given |
- 

Which of the following can be used as an alternative to static members?

Select the correct option

Local Variables

Global Variables




Constant Variables

Array of Objects

Question # 26 of 30 (Start time: 12:33:02 AM, 27 June 2020)

Which of the following can only be overloaded as member function of the class?

Select the correct option

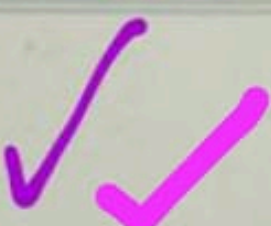
- | | |
|-----------------------|--------------------------------|
| <input type="radio"/> | Equality operator: == |
| <input type="radio"/> | Inequality Operator: != |
| <input type="radio"/> | Function Operator: () |
| <input type="radio"/> | Stream Extraction Operator: >> |
- 

CS304: Grand Quiz (Mid Term Exam)

Question # 13 of 30 (Start time: 01:23:13 AM, 27 June 202

Through interface we access object's _____

Select the correct option

- | | |
|-----------------------|--------------|
| <input type="radio"/> | States |
| <input type="radio"/> | Data members |
| <input type="radio"/> | Behaviour |
| <input type="radio"/> | Class |
- 

CS304: Grand Quiz (Mid Term Exam)

Question # 14 of 30 (Start time: 01:23:50 AM, 27 June 2020)

Which of the following statement(s) describes constant member functions?

Select the correct option

- | | |
|-----------------------|--|
| <input type="radio"/> | Constant member functions cannot modify the state of any object. |
| <input type="radio"/> | They are just "read-only". |
| <input type="radio"/> | Errors due to typing are also caught at compile time. |
| <input type="radio"/> | All of the given options |
- ✓ ✓

Question 7 of 30 (Start time: 12:59:29 AM, 27 June 2020)

Total Marks: 1

Which relationship exists between Cat and Animal class?

Select the correct option

Inheritance



Aggregation

Abstraction


Association

CS304: Grand Quiz (Mid Term Exam)

Question # 20 of 30 (Start time: 01:29:42 AM, 27 June 2020)

What is the general syntax of overloading Unary Operator as member function of the class?

Select the correct option

- | | |
|-----------------------|--------------------------------------|
| <input type="radio"/> | TYPE & operator OP (int); |
| <input type="radio"/> | TYPE & operator OP (); |
| <input type="radio"/> | TYPE & operator OP (TYPE &); |
| <input type="radio"/> | TYPE & operator OP (TYPE &, TYPE &); |
- 

Question # 28 of 30 (Start time: 04:53:40 AM, 27 June

Inheritance is the relationship between _____

Select the correct option



Functions



Interfaces



Messages



Classes




CS304 Grand Quiz (Mid Term Exam)

Question # 12 of 30 (Start time: 01:22:15 AM, 27 June 2020)

Compiler generated default constructor is called _____ and user written default constructor is call


Select the correct option

- | | |
|-----------------------|----------------------------|
| <input type="radio"/> | Compiler-defined, Implicit |
| <input type="radio"/> | Explicit, User-defined |
| <input type="radio"/> | Explicit, Implicit |
| <input type="radio"/> | Implicit, Explicit |
- 

Question # 29 of 30 (Start time: 01:37:59 AM, 27 June 2020)

Information hiding can be achieved through _____

Select the correct option

- | | |
|-----------------------|-----------------------------|
| <input type="radio"/> | Encapsulation, Inheritance |
| <input type="radio"/> | Encapsulation, Polymorphism |
| <input type="radio"/> | Encapsulation, Abstraction |
| <input type="radio"/> | Encapsulation, Overloading |
- 

CS304:Grand Quiz (Mid Term Exam)

Question # 4 of 30 (Start time: 01:41:26 AM, 27 June 2020)

In programming, where the actual actions are coded is called:

Select the correct option

<input type="radio"/>	Function declaration
<input checked="" type="radio"/>	Function calling
<input type="radio"/>	Function definition
<input type="radio"/>	None of the given

Question # 26 of 30 (Start time: 01:35:36 AM, 27 June

An object has _____ interface(s)

Select the correct option



No



Only one



Only two



One or more than one

CS304: Grand Quiz (Mid Term Exam)

Question # 16 of 30 (Start time: 01:25:46 AM, 27 June 2020)

If our class involves dynamic memory and we don't write Deep copy constructor for this class, _____

Select the correct option

- | | |
|-----------------------|------------------|
| <input type="radio"/> | System may crash |
| <input type="radio"/> | Memory leakage |
| <input type="radio"/> | Dangling pointer |
| <input type="radio"/> | All of the given |



Question # 30 of 30 (Start time: 04:55:19 AM, 27 June 2020)

In C++, which of the following is defined as stream insertion operator?

Select the correct option

- <<
- >>
- *
- <



Question # 30 of 30 (Start time: 04:55:19 AM, 27 June 2020)

In C++, which of the following is defined as stream insertion operator?

Select the correct option

- | | |
|----------------------------------|----|
| <input checked="" type="radio"/> | << |
| <input type="radio"/> | >> |
| <input type="radio"/> | *> |
| <input type="radio"/> | *< |

Question 7 of 30 (Start time: 12:48:25 AM, 27 June 2020)

Total Marks:

Choose the correct statement.

Select the correct option

- Constructors cannot be const.
- Destructors cannot be const.
- Constructors and Destructors are used to modify the object to a well-defined state.
- All of the given options




Question 7 11 of 30 (Start time: 12:51:32 AM, 27 June 2020)

Total Marks: 1

In Object Oriented programming, objects communicate with each other through _____

Select the correct option

- Messages 
- Data members
- Voice chat
- Ports

Question # 7 of 30 (Start time: 01:18:11 AM, 27 June 2020)

In which of the following OOP paradigm objects can exist independently?

Select the correct option


<input type="radio"/>	Polymorphism
<input type="radio"/>	Inheritance
<input type="radio"/>	Composition
<input type="radio"/>	Aggregation ✓ ✓

Question # 15 of 30 (Start time: 01:24:40 AM, 27 Jul

Which of the following is not a type of association?

Select the correct option

<input type="radio"/>	Polymorphism
<input type="radio"/>	Aggregation
<input type="radio"/>	Composition
<input type="radio"/>	Simple Association



Question # 22 of 30 (Start time: 01:02:54 AM, 27 June 2020)

Total Marks: 1

How many objects are involved in the N-ary association?

Select the correct option

More than 1

More than 2

More than 3

More than 4





a class have only ----destruct



People also ask

Does every class need a destructor?



What is a class destructor?




How many destructors May a class contain?



Destructor rules

*2) There cannot be more than one destructor in a **class**. 3) Unlike constructors that **can have** parameters, **destructors** do not allow **any** parameter.*

 <https://beginnersbook.com> › 2017/08

[Destructors in C++ -
BeginnersBook.com](https://beginnersbook.com)

More results

Can a class have virtual destructor yes or no?



Discover



Snapshot



Search



Collections



More





Advantages of Encapsulation

The following are the main advantages of Encapsulation,

a. Simplicity and clarity

As all data and functions are stored in the objects so there is no data or function around in program that is not part of any object and in this way it becomes very easy to understand the purpose of each data member and function in an object.

b. Low complexity

As data members and functions are hidden in objects and each object has a specific behavior so there is less complexity in code there will be no such situations that a functions is using some other function and that functions is using some other function.

c. Better understanding

Everyone will be able to understand whole scenario by simple looking into object diagrams without any issue as each object has specific role and specific relation with other objects.

02.3.Interface

Interface is a set of functions of an object that he wants to expose to other objects.

As we discussed previously that data and behavior of each object is hidden in that object itself so we have to use the concept of interface of the object to expose its behavior to outer world objects.

- Different objects may need different functions of an object so interface of an object may be different for different objects.
- Interfaces are necessary for object communication. Each object provides interface/s (operations) to other objects through these interfaces other objects communicate with this object.

Example - Interface of a Car

- Steer Wheels
- Accelerate
- Change Gear
- Apply Brakes
- Turn Lights On/Off

Example - Interface of a Phone

- Input Number
- Place Call
- Disconnect Call
- Add number to address book
- Remove number
- Update number

02.4.Implementation

It is actual implementation of the behavior of the object in any Object Oriented language.

It has two parts.

- Internal data structures to hold an object state that will be hidden from us it will store values for an object data members.
- Functionality in the form of member functions to provide required behavior.





```
ReturnType ClassName::FunctionName()
```

```
|
| ...
|
```

Example

```
class Student{
    ...
    int rollNo;
public:
    void setRollNo(int aRollNo);
};
void Student::setRollNo(int aRollNo){
    ...
    rollNo = aRollNo;
}
```

Scope resolution operator

08.3. Inline Functions

- **Inline functions** is a way used by compilers to improve efficiency of the program, when functions are declared inline normal process of function calling (using stack) is not followed instead function code is added by compiler at all points where these functions have been called. Basic concept behind **inline functions** is that they are functions in our code but in compiler generated files these functions code is added by compiler at all places where they were called in the code.
- Normally small size functions that need to be called many times during program execution are declared inline. **Inline functions** decrease the code execution time because program in their case doesn't involve function call overhead.
- Keyword 'inline' is used to request compiler to make a function inline.
- However using inline keyword with function doesn't guarantee that function will definitely in inlined, it depends on the compiler if it finds it can make function inline it does so otherwise it ignores the keyword inline and treat the function as normal function.

Example

```
inline int Area(int len, int hi)
|
|   return len * hi;
|
int main()
|
|   cout << Area(10, 20);
```

```
return 0;
```



Previous

2/6



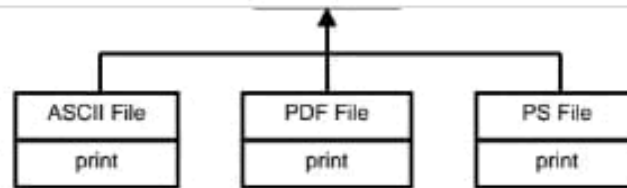
Next



flexible



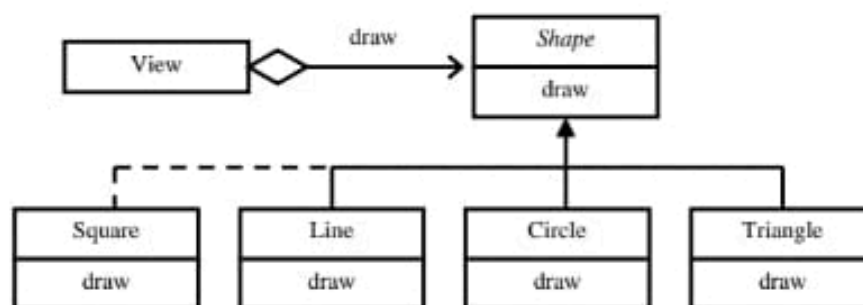
Close



Editor sends message print to file class and print is called based on the actual child object of file class message is same and appropriate execution will be done.

06.4. Polymorphism - Advantages

Messages can be interpreted in different ways depending upon the receiver class
New classes can be added without changing the existing model



In general, polymorphism is a powerful tool to develop **flexible** and reusable systems

06.5. Object-Oriented Modeling an Example

Problem Statement

Develop a graphic editor that can draw different geometric shapes such as line, circle and triangle. User can select, move or rotate a shape. To do so, editor provides user with a menu listing different commands. Individual shapes can be grouped together and can behave as a single shape.

Identify Classes

Extract nouns in the problem statement

Develop a graphic **editor** that can draw different geometric **shapes** such as **line**, **circle** and **triangle**. **User** can select, move or rotate a shape. To do so, **editor** provides **user** with a **menu** listing different **commands**. Individual **shapes** can be grouped together and can behave as a single **shape**.

57 / 355



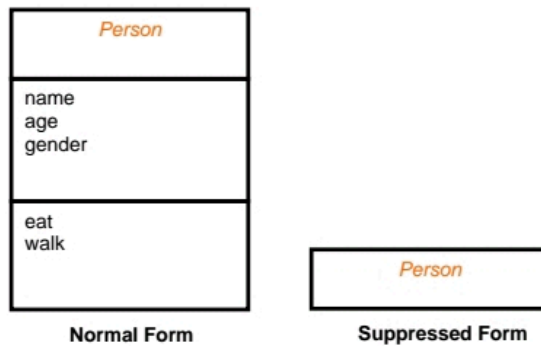
Previous

1/1



Next



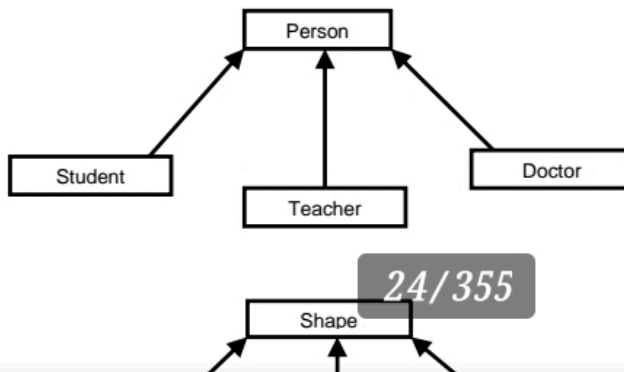


03.3. Inheritance

A child inherits characteristics of its parents, besides inherited characteristics, a child may have its own unique characteristics

Inheritance in Classes

If a class B inherits from class A then it contains all the characteristics (information structure and behaviour) of class A
 The parent class is called *base* class and the child class is called *derived* class
 Besides inherited characteristics, derived class may have its own unique characteristics



attributes



Close

Each one is a Student so we say these objects are *instances* of the Student class.

Class -Example 2

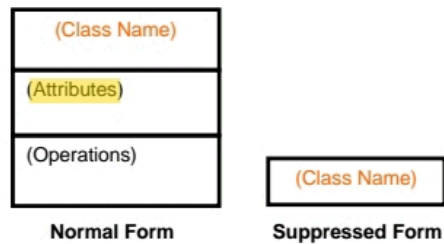
Consider the objects given below,

- Ahsan teaches mathematics
- Aamir teaches computer science
- Atif teaches physics

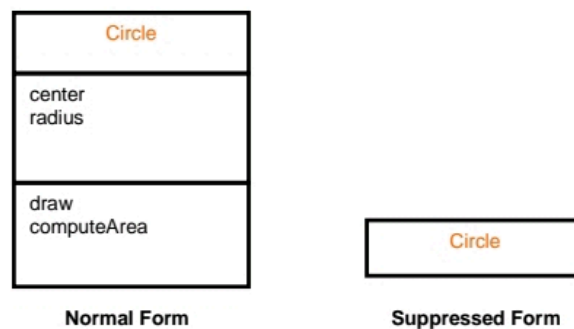
Each one is a teacher so we say these objects are *instances* of the Teacher class

Class Representation:

we can represent a class using a rectangle as follows,



Class Example: Circle



Class Example: Person

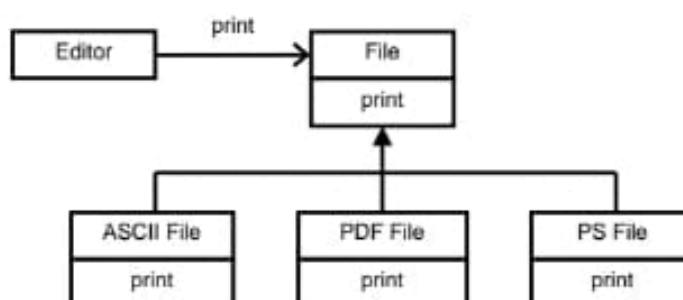


Previous

Next



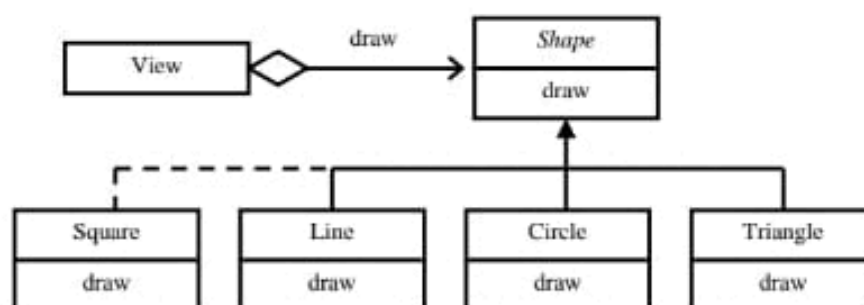
Shape class hierarchy shape is base class and there are three child classes line circle , triangle. View send draw method to shape class and draw is called according to the nature of actual object present.



Editor sends message print to file class and print is called based on the actual child object of file class message is same and appropriate execution will be done.

06.4. Polymorphism - Advantages

Messages can be **interpreted in** different ways depending upon the receiver class
New classes can be added without changing the existing model



In general, polymorphism is a powerful tool to develop flexible and reusable systems

06.5. Object-Oriented Modeling an Example

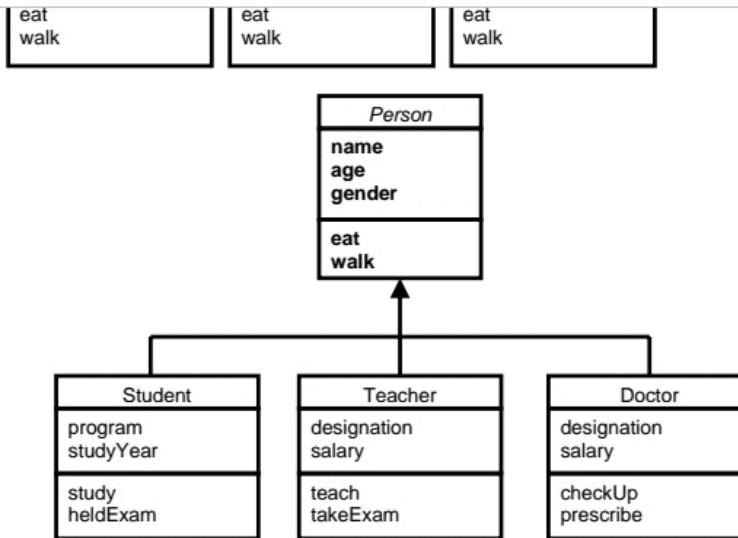
Problem Statement

Develop a graphic editor that can draw different geometric shapes such as line, circle and triangle. User can select, move or rotate a shape. To do so, editor provides user with a menu listing different commands. Individual shapes can be grouped together



that derived class

Close



Common attributes,
Name, age, gender

Common behaviour
Eat, Walk

Sub-typing & Specialization

We want to add a new class to an existing model
 We have developed an existing class hierarchy
 Find an existing class that already implements some of the desired state and behaviour
 Inherit the new class from this class and add unique behaviour to the new class

04.3.Sub-typing (Extension)

Sub-typing means **that derived class** is behaviourally compatible with the base class
 Derived class has all the characteristics of base class plus some extra characteristics
 Behaviourally compatible means that base class can be replaced by the derived class

Sub-typing (Extension) - Example



1/4



Previous

Next



Q *fix unary operator*



Close

Example (Pre-increment):

```
class Complex{
    double real, img;
public:
    ...
    Complex & operator ++ ();
    // friend Complex & operator ++(Complex &);
};

Complex & Complex::operator++(){ // member function
    real = real + 1;
    return * this;
}

Complex & operator ++ (Complex & h){ // non member function
    h.real += 1;
    return h;
}
```

Example:

```
Complex h1, h2, h3;
++h1;
```

Function `operator++()` returns a reference so that the object can be used as an *lvalue*

```
++h1 = h2 + ++h3;
```

How does a compiler know whether it is a pre-increment or a post-increment?

A post-**fix unary operator** is implemented using:
Member function with 1 dummy int argument

OR

Non-member function with two arguments

In post increment, current value of the object is stored in a temporary variable

Current object is incremented

Value of the temporary variable is returned

21.2.Post-increment operator:

```
class Complex{
    ...
    Complex operator ++ (int);
    // friend Complex operator ++(const Complex &, int);
};

Complex Complex::operator ++ (int){
    complex t = *this;
```



Previous

1/1



Next





Lecture No.04

Lecture Contents

- Generalization
- Sub typing (extension)
- Specialization (restriction)
- Overriding
- Abstract classes
- Concrete classes

Recap - Inheritance

- Derived class inherits all the characteristics of the base class
- Besides inherited characteristics, derived class may have its own unique characteristics
- Major benefit of inheritance is reuse

04.1. Concepts Related with Inheritance

- o Generalization
- o Subtyping (extension)
- o Specialization (restriction)

04.2. Generalization

In OO models, some classes may have common characteristics.

We extract these features into a new class and inherit original classes from this new class. There are many objects with common characteristics in object model. The common characteristics (attributes and behaviour) of all these objects are combined in a single general class. Base class encapsulates the idea of commonality of derived classes. Base class is general class representing common behaviour of all derived classes.

This concept is known as Generalization.

It reduces the redundancy and gives us reusability, using generalization our solution becomes less complex.

In generalization there should be "Generalization Relationship" (also called "Is A relationship") between base and child classes.



Previous

2/13



Next



Google



a class defined within another

All

News

Images

Videos

Maps

Books


The Java programming language allows you to **define a class within another class**. Such **a class** is called a **nested class** and is illustrated here: ... Non-static nested **classes** (inner classes) have access to other members of the enclosing **class**, even if they are declared **private**.

 <https://docs.oracle.com/tutorial/n...>

[Nested Classes \(The Java™ Tutorials > Learning the Java ...](#)

 [About featured snippets](#)

 [Feedback](#)

 <https://compsci.edu.com/discussion>

[A class defined within another class is:](#)

A class defined within another class is: Nested class



Discover



Snapshot



Search

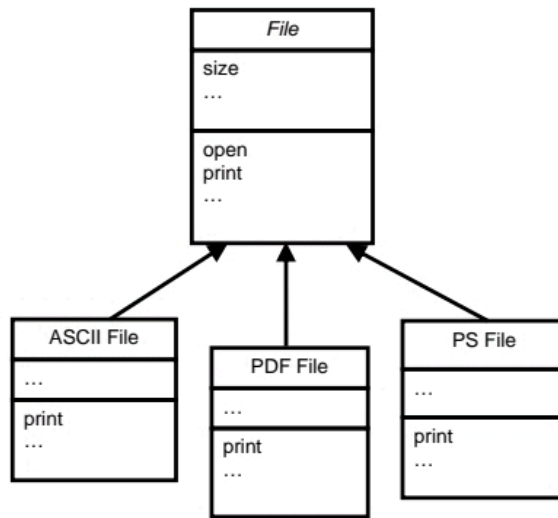


Collections



More





Wherever the file class is it can be replaced by any of its child classes.

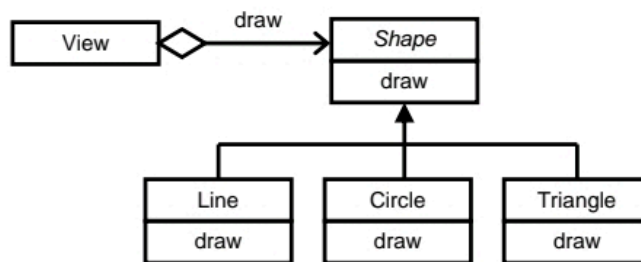
06.2. Polymorphism

It is also essential component of object oriented modeling (paradigm). In general, polymorphism refers to existence of *different forms* of a single entity. For example, both Diamond and Coal are different forms of Carbon.

06.3. Polymorphism in OO Model

In OO model, polymorphism means that different objects can behave in different ways for the same message (stimulus). Consequently, sender of a message does not need to know exact class of the receiver. Sender sends message to receiver and appropriate method is called on receiver side.

Example - Polymorphism



**Lecture No.04****Lecture Contents**

- Generalization
- Sub typing (extension)
- Specialization (restriction)
- Overriding
- Abstract classes
- Concrete classes

Recap - Inheritance

- Derived class inherits all the characteristics of the base class
- Besides inherited characteristics, derived class may have its own unique characteristics
- Major benefit of inheritance is reuse

04.1. Concepts Related with Inheritance

- o Generalization
- o Subtyping (extension)
- o Specialization (restriction)

04.2. Generalization

In OO models, some classes may have common characteristics.

We extract these features into a new class and inherit original classes from this new class. There are many objects with common characteristics in object model. The common characteristics (attributes and behaviour) of all these objects are combined in a single general class. Base class encapsulates the idea of commonality of derived classes. Base class is general class representing common behaviour of all derived classes.

This concept is known as Generalization.

It reduces the redundancy and gives us reusability, using generalization our solution becomes less complex.

In generalization there should be "Is a Kind of Relationship" (also called "Is A relationship") between base and child classes.

*Previous*

8/116

*Next*

constructor

- Compiler generated default constructor is called implicit and user written default constructor is called explicit
- This compiler generated default constructor initialize the data members to their default values
- If we have given any constructor for a class whether it is
 - *our own explicit default constructor (i.e parameterless or with*

Q *suppressed*



Close

Each one is a Student so we say these objects are *instances* of the Student class.

Class -Example 2

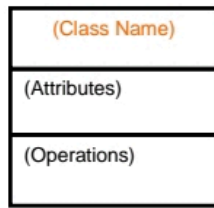
Consider the objects given below,

- Ahsan teaches mathematics
- Aamir teaches computer science
- Atif teaches physics

Each one is a teacher so we say these objects are *instances* of the Teacher class

Class Representation:

we can represent a class using a rectangle as follows,

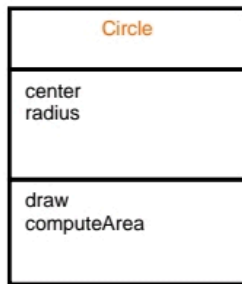


Normal Form



Suppressed Form

Class Example: Circle



Normal Form



Suppressed Form

Class Example: Person

Already the last one. Go to the first one now.



Previous

Next



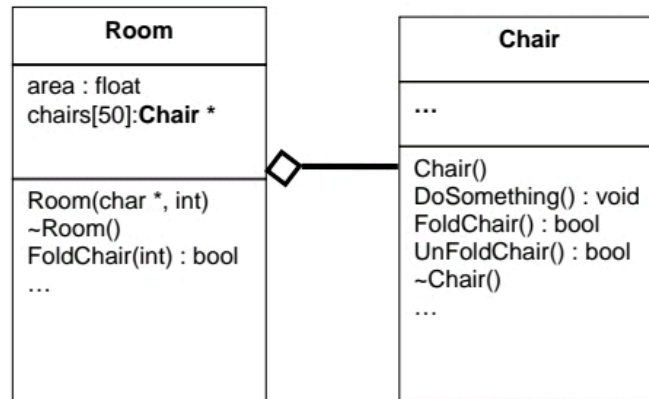
Q *created inside*



Close

Example:

Take the example of Room and Chair as given below,



Aggregation C++ implementation:

In aggregation, a pointer or reference to an object is **created inside** a class. The sub-object has a life that is **NOT** dependant on the life of its master class.
e.g

- Chairs can be moved inside or outside at anytime
- When Room is destroyed, the chairs may or **may not** be destroyed

Aggregation:

```

class Room{
private:
    float area;
    Chair * chairs[50];
Public:
    Room();
    void AddChair(Chair *, int chairNo);
    Chair * GetChair(int chairNo);
    bool FoldChair(int chairNo);
    ...
};

Room::Room(){
for(int i = 0; i < 50; i++)
chairs[i] = NULL;
  
```



Previous

1/1



Next



Q *creating objects*



Close

```

if (!_name){
    name = new char[strlen(_name)+1];
    strcpy(name,_name);
}
else name = NULL;
rollNumber = roll;
gpa = g;
}

```

```

Student::Student(const Student & st){
if(str.name != NULL){
    name = new char[strlen(st.name) + 1];
    strcpy(name, st.name);
}
else name = NULL;
rollNumber = st.roll;
gpa = st.g;
}

```

```

const char * Student::GetName(){
return name;
}

```

// never return handle of private data members or private member functions const ensures that private data members will not be changed

```

Student::~Student(){
delete [] name; // deleting name array
}

```

In C++ "it is all about code reuse"

Composition is **Creating objects** of one class inside another class

"Has a" relationship:

Bird has a beak

Student has a name

Composition

Now we change code slightly replacing name char * by String so that it is whole object of class String as it qualifies to be an object because we have to apply many operations on it like string dynamic creation and deletion, string copy using deep copy, searching a substring and so on....

Conceptual notation:



Previous

2/3



Next



Q *exposed*



Close

- o There are situations where designer wants to return reference to current object from a function
- o In such cases reference is taken from this pointer like (*this)

Example

```
Student Student::setRollNo(int aNo)
{
    ...
    return *this;
}
Student Student::setName(char *aName)
{
    ...
    return *this;
}
```

Usage:

```
int main()
{
    Student aStudent;
    Student bStudent;

    bStudent = aStudent.setName("Ahmad");
    ...
    bStudent = aStudent.setName("Ali").setRollNo(2);

    return 0;
}
```

10.2. Separation of interface and implementation

- o Public member functions **exposed** by a class are called interface.
- o Separation of implementation from the interface is good software engineering.

Benefits of separating interface and implementation:

Consider the example of following complex no. class, this complex no. class two forms of implementations one is new and one is old implementation you can observe that if you have separated interface and implementation then we can easily change implementation without changing interface,

10.3. Complex Number



1/1



Previous

Next



Q *towards the container*



Close

Composition is stronger relationship:

Composition is a stronger relationship, because
Composed object becomes a part of the composer
Composed object can't exist independently

Example I

Ali is made up of different body parts

They can't exist independent of Ali

Example II

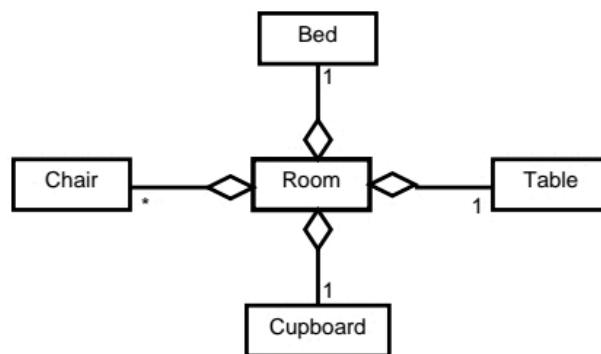
Chair's body is made up of different parts

They can't exist independently

05.5. Aggregation

An object may contain a collection (aggregate) of other objects, the relationship between the container and the contained object is called aggregation, Aggregation is represented by a line with unfilled-diamond head **towards the container**

Example - Aggregation



Example - Aggregation



Previous

1/1



Next



09.4.Destructor

1. Destructor is used to free memory that is allocated through dynamic allocation. We have to free memory allocated using new operator by over self in destructor otherwise it remain occupied even after our program ends.
2. Destructor is used to perform house keeping operations.
3. Destructor is a function with the same name as that of class, but preceded with a tilde '~'

STRUCTURE AND COMPUTE AREA.

Subtyping and generalization are related concepts, Subtyping (**extension**) and generalization is a way to look same thing in two ways.

Sub typing is looking at things from Top to bottom whereas in generalization we look at things from bottom to top.



in oop we can achieve reusal



"reusability", i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class. 26-Aug-2020



<https://eng.libretexts.org> › Courses

1.7: OOP Inheritance -
Engineering LibreTexts

More results

Which OOP concept for reusability of existing code?



*Explanation: Inheritance indicates the **code reusability**. Encapsulation and abstraction are meant to hide/group data into one element. Polymorphism is to indicate different tasks performed by a single entity.*



<https://www.sanfoundry.com> › obje...

OOP Features - Object Oriented
Programming Questions and ...



Discover



Snapshot



Search



Collections



More



Advantage of Multiple Inheritance:

As was the case with simple (single) inheritance multiple inheritance also decreases redundant code as we can inherit a class from many classes and can use their functions without the need to write them again.

However, there are more disadvantages of multiple inheritance, than its advantages.

Problems with Multiple Inheritance

Increased complexity

Amphibious vehicle hierarchy is a complicated as this class is derived from two classes that will make code more complex and less understandable however this is obvious as amphibious vehicle is a complicated vehicle. It is generic problem.

Reduced understanding

Due to increased complexity of class hierarchy the object model becomes difficult to understand especially for someone who is looking at it first time.

Duplicate features

As we are **deriving a** single class from more than one class so there is a chance of duplication of features (same methods in both parents), following problems may arise due to duplicate features,

Problem 1: Ambiguity

Following are two major advantages of information hiding,

It simplifies our Object Oriented Model:

As we saw earlier that our object oriented model only had objects and their interactions hiding implementation details so it **makes it easier** for everyone to understand our object oriented model.

It is a barrier against change propagation

Allocation with new Operator

- new operator can be used to **create objects** at runtime

Example

```
int main(){
```

argument.

Implicit Default constructor:

Compiler generates implicit default constructor for any class in case we have not given any constructor for the class.

Explicit Default constructor:

If user has given constructor for any class without any arguments or with all

Composition vs. Aggregation

Aggregation is a *weak relationship* than composition because in this relationship two classes get services of each other but can exist independently as well, main difference is memory organization of two objects as shown below,

05.4. Composition

An object may be composed of other smaller objects, the relationship between the “part” objects and the “whole” object is known as Composition, Composition is represented by a line with a filled-diamond head towards the **composer object**

itself'

Encapsulation and information hiding are much related concepts (information hiding is achieved using Encapsulation)

We have seen in previous lecture that object characteristics include data members

Inheritance - "IS A" or "IS A KIND OF" Relationship

Identify Classes

Extract **nouns** in the problem statement

Develop a graphic **editor** that can draw different geometric **shapes** such as **line**, **circle** and **triangle**. **User** can select, move or rotate a **shape**. To do so, **editor** provides **user** with a **menu** listing different **commands**. Individual **shapes** can be grouped together and can behave as a single **shape**.

05.4. Composition

An object may be composed of other smaller objects, the relationship between the “part” objects and the “whole” object is known as Composition, Composition is represented by a line with a filled-diamond head towards the composer object

Example - Composition of Ali

12.2. Life of Static Data Member

- They are created even when there is no object of a class
- They **remain in memory** even when all Objects of a class are destroyed

Question # 27 of 30 (Start time: 01:09:05 AM, 27 June 2020)

Mermaid is an example of:

Select the correct option



Single inheritance



Polymorphism



Specialization



Multiple Inheritance



05.5. Aggregation

An object may contain a collection (aggregate) of other objects, the relationship between the container and the contained object is called aggregation, Aggregation is represented by a line with **unfilled**-diamond head towards the container

Static data member is declared inside the class
But they are defined outside the class

33.2. User-Defined Types

Besides primitive types, user-defined types can also be passed as type arguments to templates, compiler performs static type checking to diagnose type errors.

Consider the String class without **overloaded** operator "=="

12.5. Global Variable vs. Static Members

- **Alternative** to static member is to use global variable
- Global variables are accessible to all entities of the program
- Use of Global variables is against the principle of information hiding

c. **N-ary Association**

An association between 3 or more classes its practical examples are very rare.

05.4. Composition

Aggregation is weaker relationship

Aggregation is weaker relationship, because

- Aggregate object is not a part of the container
- Aggregate object can **exist independently**

We can achieve **information hiding** using **Encapsulation** and **Abstraction**, so we see these two concepts in detail now,

02.6.Messages

Objects communicate through messages they send messages (stimuli) by invoking appropriate operations on the target object. The number and kind of messages that can be sent to an object depends upon its interface

Examples - Messages

```
class Parent {
public:
    void Func1();
    void Func1(int);
};

class Child: public
    Parent {
public:
    void Func1();
};
```

Function Overloading:
Two functions in same class with same name but different parameters and return type.

Function Overriding:
Two functions in two derived classes

```
class Parent {
public:
    void Func1();
    void Func1(int);
};

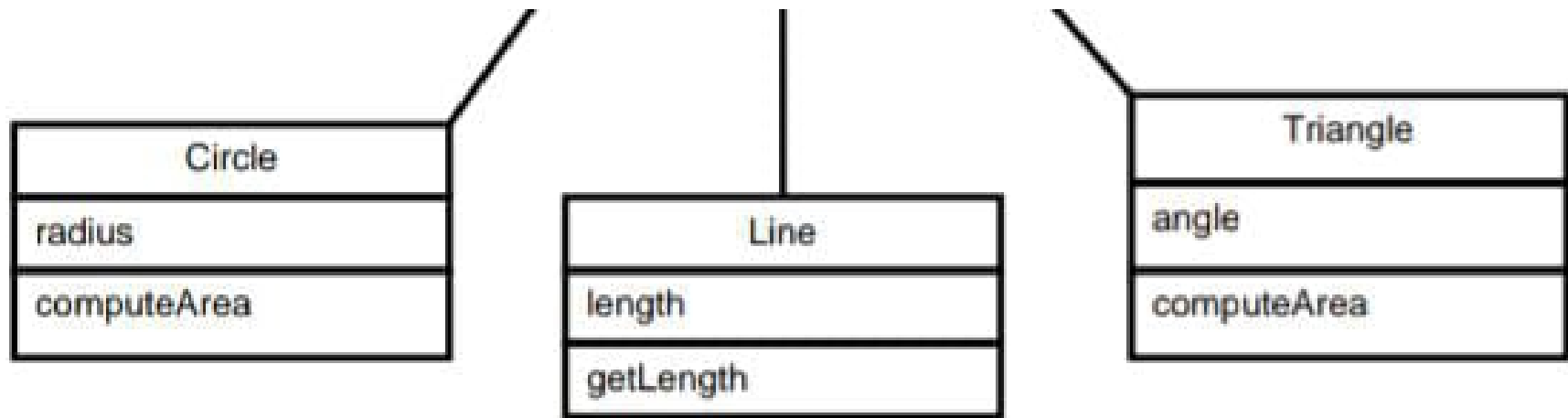
class Child: public
    Parent {
public:
    void Func1();
};
```

```
class Parent {
public:
    void Func1();
    void Func1(int);
};

class Child: public
    Parent {
public:
    void Func1();
};
```

208 / 355





Common attributes
Color vertices
Common behaviour
Set **Color**, Move

15.1. Aggregation

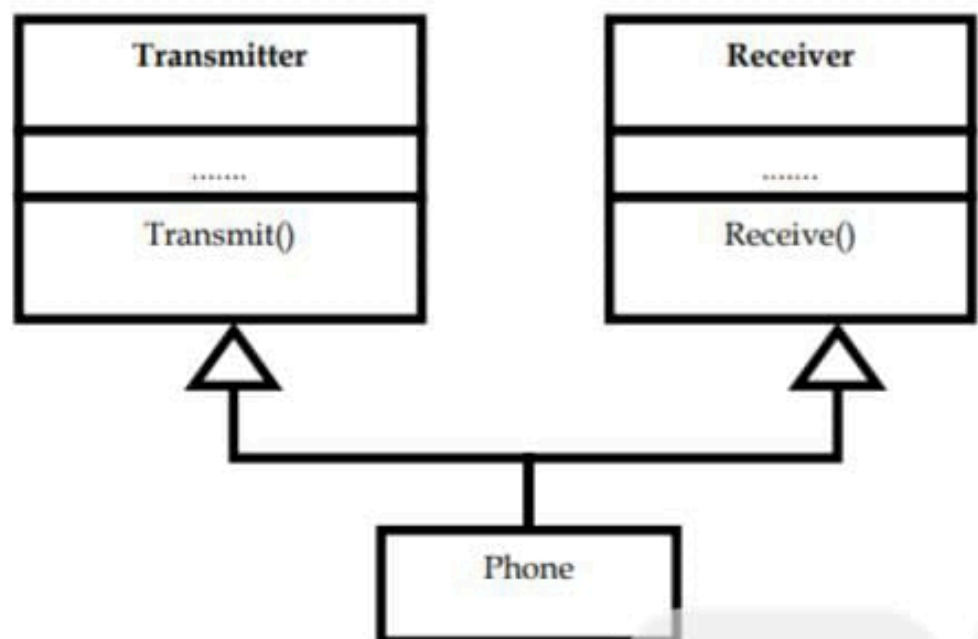
In composition we made separate object of those concepts that we think were worthy to be implemented as an object within other object to make our code simpler and to make functionality modular (divided in parts) and understandable like we made String class in Student class, but in real life most situations are such that two distinct (different) objects and one object is using services of the other one like student and teacher, student and librarian, **room** and chair, passenger and bus, book and

Lecture No.31

31.1. Multiple Inheritance

We have seen multiple inheritance in start of OOP now we see its implementation in C++, A class in C++ can inherit from more than one classes like Phone class can inherit from Transmitter or Receiver.

Examples:



250 / 355

```
class Phone: public Transmitter, public Receiver
// As phone class is publicly inherited so any class derived from Phone class will
also have access to public and protected members of Transmitter and Receiver
class
{
...
};
```

Derived class can inherit from public base class as well as private and protected base classes

```
class Mermaid: private Woman, private Fish
// As Mermaid class is privately inherited from Woman and Fish so any class
derived from Mermaid class will Not have access to public and protected members
```

05.5. Aggregation

An object may contain a collection (aggregate) of other objects, the relationship between the **container and** the contained object is called aggregation, Aggregation is represented by a line with unfilled-diamond head towards the container

Example - Aggregation