

FAST School of Computing

Object Oriented Programming – Spring 2023

Cyber Security Department

LAB 11

Association(Aggregation & Composition) in C++

Learning Outcomes

In this lab you are expected to learn the following:

- Basic Concept and Implementation of Aggregation and Composition

Note: Plagiarism(from some else or internet) in any 1 question will lead to zero marks in the whole lab task.

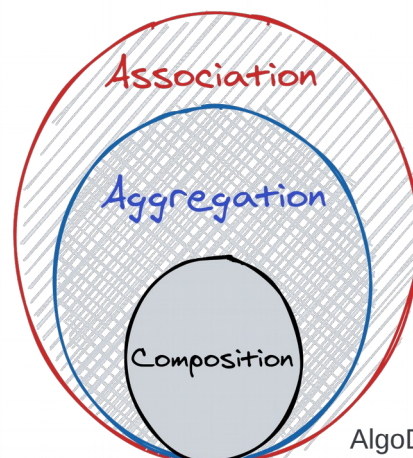
An Overview of these concepts is given in the picture below:

For detailed explanation refer to this link:

<https://www.urbanpro.com/btech-tuition/in-this-article-we-will-try-to-understand>

Association	Aggregation	Composition
Class A uses Class B.	Class A owns Class B.	Class A contains Class B.
Example: <ul style="list-style-type: none"> Employee uses BusService for transportation. Client-Server model. Computer uses keyboard as input device. 	Example: <ul style="list-style-type: none"> Manager has N Employees for a project. Team has Players. 	Example: <ul style="list-style-type: none"> Order consists of LineItems. Body consists of Arm, Head, Legs. BankAccount consists of Balance and TransactionHistory.
An association is used when one object wants another object to perform a service for it. <u>Eg. Computer uses keyboard as input device.</u>	An aggregation is used when life of object is independent of container object But still container object owns the aggregated object. <u>Eg. Team has players, If team dissolve, Player will still exists.</u>	A composition is used where each part may belong to only one whole at a time. <u>Eg. A line item is part of an order so A line item cannot exist without an order.</u>
Association UML Notation: Associations are represented by just the line (no diamond).	Aggregation UML Notation: Aggregations are represented by the line with diamond.	Composition UML Notation: Compositions are represented by the line with filled diamond.

Association, Aggregation, Composition



AlgoDaily

Problem 1:

Write a class **Point** that has the following data members.

X_Coordinate: x coordinate of type integer

Y_Coordinate: y coordinate of type integer

The Point class has following **member functions**.

1. A **default constructor** that initializes the data members to zero.

Point()

2. A **parameterized constructor** that accepts the parameters for each member variable.

Point(int , int)

3. A **copy constructor** that takes a previously constructed object as an argument.

Point(const Point &p)

4. Write **accessors** for each data member.

int getX_Coordinate() const

int getY_Coordinate() const

Write a class **Line** that represents a line segment between two Points hence it composes Point class.

The Line class has the following data members.

Point_1: a point P1 of type Point

Point_2: a point P2 of type Point

The Line class has the following **member functions**.

Note: Use member initializer list for all constructors.

1. A **default constructor** that initializes the **coordinates of two points to 4,6 and 2, 4.**

Line()

2. A **parameterized constructor**.

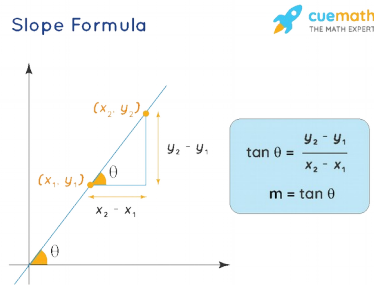
Line(int x1, int y1, int x2, int y2)

3. A **copy constructor** that takes two previously constructed Point objects as argument.

Line(const Point &p1, const Point &p2)

4. A member function **findSlope** that returns the slope of the length.

float findSlope()



5. A member function **findLength** that returns the length of the line segment using distance formula.

float findLength()

If A (x_1, y_1) and B (x_2, y_2) , then

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

6. A member function **findMidPoint** that returns the midpoint of the line segment.

In this function you have to create a pointer of type Point, dynamically allocate memory to the pointer, set midpoints and return it.

Point* findMidPoint()

The Midpoint Formula

$$\left(\frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Problem 2:

Every **course** has an **instructor**, who will teach the students and a **textbook** which the instructor is going to follow for that course. Your task is to:

Create the **Instructor class which will have following attributes:**

- First name
- Last name
- Office number
- Designation

Note: All of them will be of string data type.

And will have only the following **functions**:

- A default constructor that assigns empty strings to the instructor's last name, first name, office number and designation.
- A constructor that accepts arguments for the instructor's last name, first name, office number and designation.
- A set function that can be used to set all of the class's attributes. • A print function that displays the class's attribute values.

Now create the **TextBook class which will have following attributes:**

- string title
- string author
- int book_id

And will have only the following **functions**:

- a default constructor,
- a constructor that accepts arguments,
- A set function that can be used to set all of the class's attributes.
- A print function that displays the class's attribute values.

Now finally, Create the **Course** class which has an **Instructor object** and a **TextBook object** as member variables.

Test the functionality of all classes using main by Creating an array of 3 which contains the objects of course class and pass the instructor information and textbook information and finally prints the course information.

Submission Details:

1. Save single .cpp file with your roll no and lab number e.g. i22-XXXX_Lab#.cpp
2. Take screen shot of running test cases of tasks.
3. Zip the .cpp file and screen shots (Do not create .rar file) with roll no and lab no.
e.g. i22-XXXX_Lab#.zip.
4. Submit the zip file on google class room.