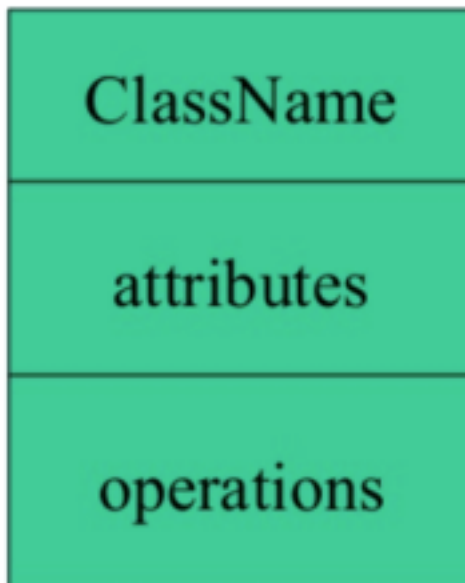


Topics in this lab:

1. Procedural and Object-Oriented Programming
2. Introduction to classes
3. Defining an instance of a class
4. Why have private members and public members
5. Separating class specification from Implementation
6. Default constructor
7. Getters and Setters (accessors and mutators)
8. Introduction to UML

UML diagram for Classes and Objects



Following rules must be taken care of while representing a class:

1. A class name should always start with a capital letter.
2. A class name should always be in the center of the first compartment.
3. A class name should always be written in **bold** format.
4. UML abstract class name should be written in *italics* format.

Attributes

An attribute is named property of a class which describes the object being modeled. In the class diagram, this component is placed just below the name-compartment.

Attributes characteristics

- The attributes are generally written along with the visibility factor.
 - Public, private, protected and package are the four visibilities which are denoted by +, -, #, or ~ signs respectively.
 - Visibility describes the accessibility of an attribute of a class.
- Attributes must have a meaningful name that describes the use of it in a class.

Examples of Classes

Classname (Identifier)	Student	Circle
Data Member (Static attributes)	name grade	radius color
Member Functions (Dynamic Operations)	getName() printGrade()	getRadius() getArea()

SoccerPlayer	Car
name number xLocation yLocation	plateNumber xLocation yLocation speed
run() jump() kickBall()	move() park() accelerate()

Objects

Classname	<u>paul:Student</u>	<u>peter:Student</u>
Data Members	name="Paul Lee" grade=3.5	name="Peter Tan" grade=3.9
Member Functions	getName() printGrade()	getName() printGrade()

Section A

- Objects are created from abstract data types that encapsulate _____ and _____ together.
 - numbers, characters
 - data and functions
 - addresses and pointers
 - integers and floats
 - None of the above
- In OOP terminology, an object's member variables are often called its _____ and its member functions are sometimes referred to as its behaviors, or _____.
 - values, morals
 - data, activities
 - attributes, activities
 - attributes, methods
 - None of these
- A C++ class is similar to one of these.
 - inline function
 - header file
 - library function
 - structure
 - None of these
- Examples of access specifiers are the keywords:
 - near and far
 - opened and closed
 - private and public
 - table and row

5. This is used to protect important data.
 - a) public access specifier
 - b) private access specifier
 - c) protect() member function
 - d) class protection operator, @

6. Class declarations are usually stored here.
 - a) On separate disk volumes
 - b) In their own header files
 - c) In .cpp files, along with function definitions
 - d) Under pseudonyms
 - e) None of these

7. This type of member function may be called from a statement outside the class.
 - a) public
 - b) private
 - c) undeclared
 - d) global
 - e) None of these

8. If you do not declare an access specification, the default for members of a class is
 - a) Inline
 - b) Private
 - c) Public
 - d) Global
 - e) None of these

9. In a procedural program, you typically have _____ stored in a collection of variables, and a set of _____ that perform operations on the data.
 - a) Numbers, arguments
 - b) Parameters, arguments
 - c) Strings, operators
 - d) Data, functions
 - e) None of these

Section B

Problem 1:

Write and represent by UML diagram a program that demonstrates the Circle class

by asking the user for the circle's radius, creating a Circle object and then reporting the circle's area, diameter and circumference.

The **Circle** class should have the following and should be written as (Circle.h and Circle.cpp):

1. Member variables:
 - a) Radius (double variable)
 - b) Pi (double variable, initlized with the value 3.14159)
2. Member functions:
 - a) Default constructor: sets the radius to 0.0
 - b) setRadius: A mutator function that sets the radius variable
 - c) getRadius: An accessor function that returns the radius variable
 - d) getArea: Returns the area of the circle which is calculated as :
$$area = pi * radius^2$$
 - e) getDiameter: Returns the diameter of the circle which is calculated as :
$$diameter = radius * 2$$
 - f) getCircumference: Returns the circumference of the circle, which is calculated as :
$$circumference = 2 * pi * radius$$

Solving steps (Answer the following questions):

1. What is the input and what are we expecting as an output?
2. What will be set as private and what will be set as public?
3. Where is the object created?

Extra request:

Design a function called “display” and add it to the “circle” class, this function is to display the results as follows and modify the main program to use the “display” function:

The circle radius = 5
The circle area = 35
The circle circumference = 24

Problem 2:**Write and represent by UML diagram a program that demonstrates**

Design an **Inventory class** that holds information and performs some calculations about items in a retail store.

The Inventory class should have the following and should be written as (Inventory.h and Inventory.cpp):

1. Member variables:

- a) itemNumber: (int)
- b) quantity: (int)
- c) cost: (double for holding price per item)
- d) totalCost: (double for holding total inventory cost calculated as quantity*cost)

2. Member functions:

- a) Default constructor sets all member variables to 0
 - b) Mutator functions namely, setItemNumber, setQuantity, setCost
 - c) Accessor functions namely, getItemNumber, getQuantity, getCost, getTotalCost
 - d) calTotalCost: calculates the total inventory cost for the item (quantity*cost)
- Demonstrate the class in a program (main).

Input validation: Do not accept negative numbers for the itemNumber, quantity and cost.

Extra Request:

1. Design a function called “getAllValues” and add it to the **Inventory** class, this function is to ask the user about each input value apart and to store them in their appropriate places in the inventory class and modify the main program in order to use the “getAllValues” function.
2. Design a function called “display” and add it to the “inventory” class, this function is to display the results as follows and modify the main program in order to use this function.

The values of the created object are as follows:

Item number: 3542617
Quantity: 50
Cost: 100 \$
Total cost: 5000 \$

NOTE solution of this lab will be uploaded by the end of the week