

Week 4: Constructor, Destructor, Default Parameters, const and static Members, Pass object and return object

Learning Materials: Chapter 6

Home Practice Task

For all lab tasks in Lab 3 (copy the code of Lab 3 to Lab 4), modify the class definition to add **constructors** that assign the member variables to an appropriate initial value. Write a zero-argumented constructor as well as an argumented constructor that initializes the member variable to the passed arguments' value.

Add a destructor function that displays a message: **"The destructor for the object(<name of the class>) is called."** This demonstrates that for each object, when the lifetime ends, the destructor is called.

Task 1

Design a **Trigonometry Calculator** class where the object of this class will have a field to keep track of the current angle in degrees (float type). The class should perform trigonometric calculations (like sine, cosine, and tangent) based on this angle and allow updating the angle as needed. Encapsulation must be enforced with private fields and controlled access. In case of an invalid operation (e.g., calculating the tangent of 90 degrees), the object will retain the previous angle and display a message: **"Invalid operation: undefined result."**

List of methods for the Calculator class:

Constructor methods:

Zero-argumented constructor initializes the object with 0 value to the current status. One-argument constructor initializes the current status with the passed argument.

Get methods:

Write appropriate get and set methods for the **current** status. But make them **private**.

Trigonometric operations:

float calculateSine(); - Returns the sine of the current angle.
float calculateCosine(); - Returns the cosine of the current angle.
float calculateTangent(); - Returns the tangent of the current angle. If the tangent is undefined (e.g., 90 degrees), the previous angle is retained, and a warning message is displayed.
float calculateArcSine(); - Prints the radian equivalent of the angle and returns the inverse sine (arcsin).
float calculateArcCosine(); - Returns the inverse cosine
float calculateArcTangent(); - Returns the inverse tangent
void calculateForAngle(float paramAngle); - Accepts an angle in degrees and displays the sine, cosine, and tangent of that angle without modifying the internal angle of the calculator.

Utility methods:

void clear(): Resets the current value to 0.
void display(): Displays the current value. e.g. "Calculator display: <current status>"

Destructor:

~Calculator(): Displays the current value and message that **"Destructor of the Calculator object is called."**

Sample Output:

Call Each Method and Display the Results

Task 2

Define a class "**BankAccount**" with the following description.
Each account will have the following information:

Private members

- The account number.
- The account holder name.
- The account type (current/savings) (assume the data type)
- The current balance.
- The minimum balance (An account has to maintain the minimum amount; it cannot withdraw. It can only be initialized at the time of creating object. **USE const modifier**)

The class will have the following criteria:.

Public members

- The member variable value of the object can be assigned **during** object **creation** or **after** the object has been created.
- A function to show all the information of a **BankAccount** object.
- Function **showBalance()** (for displaying current balance),
- Functions **deposit()** and **withdrawal()** of money from an account. Show appropriate messages for invalid amount.
- Function **giveInterest()** will deposit net interest to the account. Default interest is **3 percent** of current balance but it might be different. A fixed **10%** Source Tax will be deducted from the incurred interest.
- When the **BankAccount** object is destroyed display a message like :
Account of Mr. X with account no 1234 is destroyed with a balance BDT 5000

Non-member functions

Define a non-member function `display_stat()` (this is not a member function of the `BankAccount` class) that displays the following information:
Total number of `BankAccount` objects created and total number of `BankAccount` objects currently present in the program. It also shows the total amount of source tax collected from all the bank accounts.

Define a non-members function with the prototype

BankAccount Larger(const `BankAccount` A, const `BankAccount` B): it returns the `BankAccount` Object that has a higher current balance.

Task 3

Define a class in C++ with following description:

Private Members:

- A data member EmpName of type String
- A data member ID of type Integer
- A data member Age of type integer
- A data member Salary of type float
- A member function getStatus() to **return** the status of an employee on the basis of the following criteria

Age	Salary	Status
<=25	<=20000	Low
	>20000	Moderate
>25	<=21000	Low
	>21000 and <=60000	Moderate
	>60000	High

Public Members:

A function **FeedInfo()**, to allow the user to enter information of an employee like **EmpName, ID, Salary**.

A function **ShowInfo()**, to allow the user to view the content of all the data members. At the end it should display the status of an employee (Low Salaried Person, Moderate Salaried Person, High Salaried Person).