I've organized it from Beginner to Advance. I hope you find it helpful!
Follow Me—> YouTube || Instagram

C++ Problems

1. Print "Hello World"
2. User input a value and print the value
3. Sum two numbers using user input
4. Sum two numbers using function
5. Calculator using function
6. Define a class and create a object

⏩01-Class & Object

1.Create a class called person. It has two attributes called Name, contact. Initialize the attributes with two objects.

2. A class named Teacher has three attributes (Name, short code, Dept). Initialize the attributes using a member function called set_value() and display them using get_value(). Create one object of the Teacher class.

3. Create a class called Employee which has two public attributes Name and Address. A private attribute salary to store the salary. Now create an object of the class to initialize the attributes and display the public attributes.

4. Create a class called Square with two attributes length and width. Now calculate the area using a function. Find which object is smaller.

⏩02-Constructor & Destructor

1. Create a class “Person”. Which has two public attributes (Name, Address) and a private attribute Address. Initialize the attributes using constructor and print the values using a function display().

2. Create a class “Distance” which has two attributes (length,width). Initialize the attributes using a parameterized constructor and display the attributes.

3. Create a class called “Person” with two attributes: name, address. Initialize the attributes with a parameterized constructor. Print the values with a display function. Define all the methods outside the class.

4. Define a class Circle with attributes radius and area. Now, with a parameterized constructor, initialize the radius. Calculate the area using a calculate() function and print the area using another function called display(). And print “Circle destroyed” with a distractor. Now access file members of the class using an object.

5, Define a class “Box” with three attributes length, width and area. Initialize them with a parameterized constructor and print “Box created”. Calculate the area of the Box and print. Lastly, use a destructor to print “Box destroyed”.

6. Design a class Temperature with attributes celsius and fahrenheit.
Create a constructor that initializes the temperature in Celsius and an inline function to convert it to Fahrenheit. Display the temperature in both Celsius and Fahrenheit using an object.

⏩ 03-Object Pointer

1. Design a class named Product with attributes product_Name(Public), product_Code(Public), and product_price(private). Use a parameterized constructor to initialize these attributes. Implement a method called apply_discount() that reduces the price by a given percentage. Call the function using an object pointer.

⏩ 04-Inline Function

1, Create a class circle with an attribute Radius. Set the value to radius using constructor. Create a non-member function called “area()” to calculate area. Define the area() function inline.

⏩05-Object pass and return

1. Design a class named Time with attributes for hour and minutes. Create two Time objects representing different times. Implement a non-member function called add_times() which takes two Time objects as parameters and returns a new time object representing the sum of these two times.

2. Design a class named Time with attributes for hour and minutes. Create two Time objects representing different times. Calculate the difference in the time between the two objects using a non-member function called diff() which takes two Time objects as parameters and returns a new time object representing the sum of these two times.

3, Implement a class Box with attributes length, width, and height.
Write a function combineBoxes() that accepts two Box objects, combines their dimensions by adding corresponding sides, and returns the resulting Box object.
Demonstrate the combination of two boxes using object passing and returning.

⏩ 06-Array of Object

1, Create an Employee class with name and salary as attributes. Initialize the attributes and find the average salary of 10 Employees

2, Create a student class with name and English_mark, Math_mark as attributes. Initialize the attributes and find the total mark of 15 students.

3, There is a class Student that illustrates some students where each student has a name, an id number, and marks.Implement a function to set students name, id number, and another function to show the students name, id number, and marks. Now consider the above scenario and demonstrate a C++ program to set and show the average mark of 20 students using array of objects.

⏩ Mid CT Question:

1. a) Define the class Book for the given code:

```
int main() {
   Book b1;
   b1.readBook();
   b1.displayBook();
   return 0;
}
```

1. b) How can you access the private attribute `balance` of class `Account` outside the class? Write the code:

```
class Account { private: float balance; };
```

1. c) Define the functions of the `Distance` class outside the class:

```
class Distance {
public:
   int feet;
   float inches;
   void addData(int f, float in);
   void showData();
};
```

2. a) Design a `Product` class. The product class should have attributes for its name, price, and quantity. Implement a function called setProduct() within the class to input the product details from the user. Additionally, create a function named printProduct() to display the product details. Ensure that setProduct() is not directly called in the main() function. Add a destructor to print "Product object destroyed" when an object is destroyed.

b) Implement a class called `Account` with attributes like account holder name, account number, account type, and balance. Use a parameterized constructor to set these attributes when a new account is created. If attributes are not provided, use the default constructor to set the values:

- account holder name = "Not Assigned"
- account number = 0
- account type = "Savings"
- balance = 0.0

Final Problems

⏩07-Object Pointer

1. Create a Teacher class with public attributes Name, Short_code, Address, and a private attribute salary. Initialize the attributes for 10 objects of the Teacher class. Now, find a specific teacher based on the short code (given by the user) and print the details of that teacher using a pointer. Also, find the average salary of the teachers.

2. Design a class named Product with attributes: product_Name (public), product_Code (public), and product_Price (private). Use a parameterized constructor to initialize these attributes. Implement a method called apply_discount() that reduces the price by a given percentage. Call the function using an object pointer.

⏩08-Friend Function

1. Create a class called square1 and square2. Now calculate their are (private attribute). And find which class's area is greater.

2. Design a class named Real with Private attribute number. Now initialize the attribute and print it using a display() function. Design another class named Imaginary with private attribute number, initialize and display number using the display() function.

3. You have two classes, Employee and Manager. The Employee class has attributes for name and salary that are private. The Manager class has the same attributes as the Employee class and an additional attribute called department. Implement a mechanism to pass an Employee object to the Manager class to assign the same attribute values.

4. Create a class `Employee` with private members `Employee_id` and `salary`. Write a friend function named `displayEmployeeInfo` (outside the class) that can access these private members and print out the employee's information.

Instructions:

- Implement the `Employee` class with a constructor to initialize the `Employee_id` and `salary`.
- Create the friend function `displayEmployeeInfo`.

- Show how the friend function can be called in the `main` function to display employee details.

5. Create a class named `Number` that has private members `value1` and `value2`. Write a friend function `addValues` that takes two `Number` objects and returns the sum of their values in a third object.
Instructions:

- Implement the `Number` class with a constructor to initialize the values.
- Create the friend function `addValues`.
- Demonstrate the function in the `main` function by creating two `Number` objects and printing their sum.

6. Define a class named `Student` that has private members `name` (a string) and `grade` (a float). Write a friend function named `calculateAverageGrade` that takes an array of `Student` objects and returns the average grade of all students in the array.

Instructions:

- Implement the `Student` class with a constructor to initialize `name` and `grade`.
- Write the friend function `calculateAverageGrade`.
- In the `main` function, create an array of `Student` objects, populate it with data, and call the friend function to compute and display the average grade.

7. Create an Employee class with public attributes: `Name`, `address`, and private attributes: `salary` and `working_hours`. Another class called `HR` with public attributes `name`, `address`, and `salary`. Initialize the attributes with parameterized constructors. Now, a function called `calculateBonus()` in `HR` class will calculate the bonus of employees based on their salary and working hours. 5% bonus if working hours > 40 hr, 20% bonus if working hours > 30 hr, 10% bonus if working hours > 20 hr, otherwise no bonus is allocated. Implement the program to show the bonus amount to employees.

⏩09-Array

1. Create a `Student` class with public attributes `name`, `id`, and `cgpa`. Initialize the values with a `set_data()` method and print the attributes with a `get_data()` method. Write a program that

creates an array of students, sets their data, and prints the name of the student with the highest `CGPA`.

⏩10-Friend Class

⏩11-Function Overloading ✅

1. Write a function named addvalues() ; it will add two members, three members and four members. Now overload addvalues() function to achieve this.

2. Develop a class TimeManipulator with overloaded functions to add and subtract time intervals from a given time. Provide versions that work with hours, minutes, and seconds.

3. Overload the functions to calculate the area of a circle, rectangle, and triangle. When you pass the radius, the area of the circle will be calculated. When you pass the length and width, the area of the rectangle will be calculated. When you pass the base and height, the area of the triangle will be calculated. (Area of triangle: base×height/2)

4. Write a program in C++ that converts temperature values between Celsius, Fahrenheit, and Kelvin using function overloading. Implement an overloaded function `convertTemperature()` with the following variations:

- `convertTemperature(double celsius)`: Accepts a temperature in Celsius and returns its equivalent in Fahrenheit.
- `convertTemperature(double fahrenheit, char fToC)`: Accepts a temperature in Fahrenheit and a character to specify that it should be converted to Celsius, returning the result.
- `convertTemperature(double kelvin, bool isKelvin)`: Accepts a temperature in Kelvin and a boolean `isKelvin` to indicate conversion to Celsius, returning the result.

Conversions:

- Celsius to Fahrenheit: $F=C\times9/5+32$
- Fahrenheit to Celsius: $C=(F-32)\times5/9$
- Kelvin to Celsius: $C=K-273.15$

⏩12-Copy Constructor

1. Design a class called square with width as attribute. Now define a copy constructor to decrease the width of an object with another object by 50. Create three objects to invoke the constructors.
2. Create a class called Manager with the following attributes, name,dept,unique_ID.Overload the constructors(default, parameterized and copy constructor). Create three objects of the manager class to invoke three constructors.

⏩13_Operator Overloading

1. Overload the + operator to increase the attribute length of square class by 10.

```
Class square
{
Public:
float length;
};
int main()
 {
   square s1(5.3), s2(2.4), s3;
   s3 = s1 + 10;
   s3 = 10 + s1;
   cout<< s3.length << endl;
   return 0;
}
```

2. Overload the > operator to compare the attribute area of square class for two objects.

```
Class square
{
Public:
float length;
                -    - -
};
int main()
 {
   square L1(5.3), L2(2.4), L3;
if( L1>l2)
{ cout<<"area of L1 is greater"; }
```

```
    return 0;
}
```

3.

⏩14_Inheritance

1. Design a class School with attribute Department (Public). Create another class Teacher with attribute name, short_code, Teacher class inherits school in private mode. Now initialize all the attributes using teacher class object.
    - Design another class Student that inherits teacher class in public mode. Student class has attributes name, ID, course_name (private). Initialize all the attributes with student object.

2. Create a class called `Parent` with a private attribute `p` and another class called `Child` with a private attribute called `c`. Now write a function in the `Child` class to multiply the attributes and print them.

3. Create a class `University` with attributes (`public: name, Id`, `private: Address, Dept`). Create another class called `Student` with a private attribute `cgpa` that inherits `University` in protected mode. Create another class called `Teacher` with private attributes called `short_code` and `salary`. Now initialize the attributes for a teacher and a student.

4. Create a class called `BUBT` with a default constructor that prints "Welcome to BUBT". Now inherit the `BUBT` class with `Student` class and `Staff` class. Both classes have `id`, `address`, and `name` as attributes. Initialize them using their constructors.

5. Design a class called `ABC_company` with attributes `emp_name`, `emp_address`, and `emp_ID`. Create a class `IT_dept` that inherits `ABC_company` in private mode with additional protected attributes `working_hour` and `position`. Another class named `HR` inherits `ABC_company` and `IT_dept` with an additional member function `salary_cal()` to calculate the salary based on the working hours and position of the employees. Implement the code.

6. Create a class called Vehicle with three public members color, model and functions setValue(), and printDetails(). Create a car class to inherit the vehicle class and initialize the attributes of the parent class with a child object. And print the car details with a child object.

7. Create a parent class named A and a child class B. Parent class has an int type attribute called a. Class B has an int type attribute called b. A child class's member function is used to perform multiplication of parent and child attributes.

8. Create a class called Bank with protected members: ID, Name, contact_info. And two functions set_Value(), displayValue() to set and display the values. Now create a child class Employee with an attribute called department to inherit the Bank class privately and another class called Customer with an attribute called amount which will also inherit the Bank class as protected. Now initialize the members using the child classes and display their details.

⏩15_Function Overriding

⏩16_Virtual Function

⏩17_Template

1. Swap two numbers without a third variable.

⏩18_Exception Handling

A program may face some unexpected error during run-time. C++ can handle these errors with exception handling.

Syntax:

```
try
{
}
catch ()
{
}
```

Problems

1. Write program to handle an exception when age<age;

⏩Lab Evolution

1. Implement a `BankAccount` class where each account has a balance. Overload `+` and `-` operators to deposit and withdraw money.

2. Implement code for the statement: `ob1 = (10 + ob3) * ob2` to compile correctly.

3. Write a `Statistics` class with overloaded mean functions:

- Mean function that calculates the mean of an array of integers.
- Mean function that calculates the mean of an array of floating-point numbers.
- Mean function that calculates the mean of two integers.
- Mean function that calculates the mean of two floating-point numbers.

⏩Final Assignment

1. Difference between pass value and pass by reference with coding example.

2. Design a class called Distance with an attribute length. Now overload + operator to add two Distance objects. If the distance objects aren't equal, make the lesser object equal to the greater.

⏩Final Lab report

1. Create a Company class with attributes called no_of_employee, no_of_dept, employee_id, employee_name and a function set_value() to initialize the attributes. Create another class HR with attributes, address, salary, contact and a function set_value() to set the values. Now create an Employee class which inherits Company class and HR class. Now call the functions of the parent class with the child's object.

2. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle". Now, try calling the function by the object of each of these classes.

3. We want to calculate the total marks of each student of a class in Physics,Chemistry and Mathematics and the average marks of the class. The number of students in the class are entered by the user. Create a class named Marks with data members for roll number, name and marks. Create three other classes inheriting the Marks class, namely Physics, Chemistry and Mathematics, which are used to define marks in individual subjects of each student. Roll number of each student will be generated automatically. Create some more questions like this.

4. Create a class named Library with attributes such as lib_name and location. Add a function set_library() to initialize the attributes. Derive a class Section that inherits the Library class, with attributes section_name and section_code. Add a function set_section() to initialize its attributes. Further, derive another class Book from the Section class, with attributes book_name, author, and isbn. Add a function display_info() to display all details about a book. Implement the concept of multilevel inheritance and display the attributes using the most derived class's object.

5. Create a class Department with attributes dept_name and dept_code. Add a function set_dept() to initialize the attributes.
   Create a separate class Professor with attributes prof_name and prof_id.
   Add a function assign_department() to assign a department to a professor. Derive a class HOD (Head of Department) from Professor, which includes additional attributes hod_start_date and hod_end_date. Implement the functionality to display all details.

⏩Final Lab 1.1

1. **Create a class `Time` that stores hours and minutes. Overload the prefix `--` operator to decrement the time by 1 minute.**
   Example Code:
   ```
   int main() {
      Time T1(2, 59);
      T1--;
      T1.display();
      return 0;
   }
   ```

2. **Create a base class `Member` with attributes `name` and `membershipType`. Add a virtual method `calculateFee()` to calculate the annual membership fee.**
   Derive classes `GoldMember`, `SilverMember`, and `BronzeMember` with overridden `calculateFee()` methods:

- **GoldMember**: Fee = 10,000.
- **SilverMember**: Fee = 7,000.
- **BronzeMember**: Fee = 5,000.

Write a program to display the membership fees for each type.

3. **Write a recursive function template `gcd()` to calculate the greatest common divisor of two numbers.**
   Example Code:
   ```
   int main() {
      cout << "GCD of 24 and 36: " << gcd<int>(24, 36) << endl;
      cout << "GCD of 45.0 and 60.0: " << gcd<double>(45.0, 60.0) << endl;
      return 0;
   }
   ```

4. **Create a base class `LibraryItem` with attributes `title` and `daysOverdue`. Add an overloaded method `calculateFine()` to calculate fines for different library items based on the type of item.**
   Define the following fine rules:
- **Book**: Fine = $0.50 per day.
- **Magazine**: Fine = $0.30 per day.
- **DVD**: Fine = $1.00 per day.

The overloaded `calculateFine()` method should:

1. Calculate the fine for a **Book** when called with the number of days overdue.
2. Calculate the fine for a **Magazine** when called with the number of days overdue and a discount rate (e.g., 10% discount on the fine).
3. Calculate the fine for a **DVD** when called with the number of days overdue and a late fee multiplier (e.g., if the multiplier is 2, the fine is doubled).
5.

⏩Final Lab 1.2

1. Overload the relational operator < for the following code
   ```
   Int main()
   {  Distance D1(11.10), D2(5.11) };

   if( D1<D2)
   ```

```
{
cout<< “Height of D1 is less than D2” <<endl;
}
Else
{
cout<< “Height of D2 is less than D1” <<endl;
}
Return 0;
}
```

2. **Create a class hierarchy where `Employee` is the base class, `Manager` and `Developer` are derived classes.**

- The base class has a public attribute called `Working_hour`.
- Add a virtual method `getSalary()` in the base class and override it in the derived classes to calculate and return the salary of each employee.
  (Salary of Manager per hour: 300 Tk, Developer: 400 Tk)

3. Create a base class Vehicle with attributes name and ratePerDay (protected). Derive classes Car and Bike with methods calculateRentCost(days) to calculate the rental cost based on the number of days. Car rate: 2000 Tk/day . Bike rate: 500 Tk/day

4. Create a class BankAccount with overloaded functions withdraw() to: Withdraw a specific amount from the account. Withdraw a specific amount from the account after confirming a PIN. Withdraw a specific amount from the account while checking overdraft limits.