OOP's LAB-3,Lab-4, Lab-5 Experiment (Class, Object and OOP's Concepts)

Experiment 1:

An employer plane to pay a bonus to employees. A bonus of 10% is to be paid to employees with earning at least Rs. 2000/- of the earning to others. The input contains name and earning of an employee and the desired output is name and bonus to be paid to the employee.

Create a class to represent an employee. It should incluse the following:

Data Members:

- Name
- Earning
- Bonus

Member Functions:

- To input data
- To compute bonus
- To output the desired information

Using this class, write a program to accomplish the intended task.

Experiment 2. The monthly telephone bill is to be computed as follows:

Minimum Rs. 200 for upto 100 calls
Plus Rs. 0.60 per call for next 50 calls
Plus Rs. 0.50 per call for next 50 calls
Plus Rs. 0.40 per call for any call beyond 200 calls.

The input contains name of the customer and number of calls made and the desired output is the name and telephone bill to be paid by the customer.

Create a class to represent an employee. It should include the following:

Data Members:

- Name
- Number of calls
- Bill amount

Member Functions:

- To input data
- To compute bill
- To output the desired information

Using this class, write a program to accomplish the intended task.

Experiment 3. Income tax for individuals is computed on slab rates as follows:

Income	Tax Payable
Upto Rs. 1,00,000/-	Nill
From Rs. 1,00,001/- to Rs. 1,50,000/-	10% of the excess over Rs. 1,00,000/-
From Rs. 1,51,001/- to Rs. 200,000/-	20% of the excess over Rs. 1,50,000/-
Above Rs. 2,00,000/-	30% of the excess over Rs. 2,00,000/-

The input contains name of the individual and income and the desired output is the name of the individual and amount of tax to be paid by the individual.

Create a class to represent an employee. It include the following:

Data Members:

- Name
- Income
- Tax due

Member Functions:

- To input data
- To compute tax
- To output the desired information

OOP's LAB-3,Lab-4, Lab-5 Experiment (Class, Object and OOP's Concepts)

Using this class, write a program to accomplish the intended task.

Experiment 4. Create a class to represent a bank account. It should include the following:

Data Members:

- Name of account holder
- Account number
- Type of account
- Balance amount

Member Function:

- To initialize the data members with appropriate data
- To deposit an amount
- To withdraw an amount after checking the balance
- To display details of account holder

Write a program to use this class.

Experiment 5: Assume that a publishing company markets prints books and digital books (electronic form – CD'). Create a class named Publication with data members named title, price, and author's name. From Publication class, derive two classes named Book and Ebook. The Book class adds a page count data member named pcount while Ebook adds data member playing time name ptime. Each of these classes must have member function getDatd() to read class specific data from keyboard and displayData() to output the class specific data to the computer screen. Write a program to test these classes.

Experiment 6: Write a program to find mean of two numbers belonging to two different classes using friend function.

Experiment 7: Write C++ program to calculate area of a cuboid. Convert the cuboid into a cube using a friend class.

Experiment 6: Write a C++ Program to implement the arithmetic operations multiplication on complex numbers using the concept of this pointer, inline functions and scope resolution operator.

Experiment 7: Create a record of 10 employee which contains following information: emp_id., emp_name, and age. Create a C++ (OOPs concept) program to search an employee with its emp_id.

Experiment 8: (friend function with inline function, friend function with another class)
Implement Two Class name is class_2,class_1 define one function name is exchange this function work in both class properties. friend function can be called by reference inthis case copies of the object are not made. Instead a pointer to the address of the object is passed

Experiment 9: (Dynamic memory allocation with new, delete, and destructor)
A Program to allocate contiguous memory for an array using the new operator and the object of the array is destroyed by the delete operator.

Experiment 10: (Operator Overloading)
Design a class named vector to model a one-dimensional array with integer elements.
Overload the followimg operators to perform the intended task:
(i) Operator '<<' to output a vector object.

(ii) Operator '>>' to input a vector object in the form [10, 20, 30, . . .].
(iii) Operator '+' to add two vector objects.
(iv) Operator '-' to subtract a given vector object from another vector object.

Experiment 11 :
Create class name is Integer in this class implement the overloading >> and<< opertor using friend function.

Experiment 12:
Create class name is Integer in this class implement the operator over loading using Compare to String.

Experiment 13:
Create class name is Integer in this class implement the operator over loading using friend function Binary(+) operator.
