Tutorial-8
EEN-103

1. Write a program using a pointer to function that adds, subtracts and multiplies two numbers. You can call these functions only using pointers. Is it possible to write the same code with array of function pointers?

2. The prototype of a function, called sum, which has a pointer to function as pf. This function sums the square and cube of even integers up to an integer n. You can call sum by passing the pointer to other function as

```
int sum (int (*) (int), int);
```

3. Write a program where the user is first provided with the following three choices:

   - To compute the area of a circle.
   - To compute the area of a triangle using Heron's formula.
   - To compute the area of a rectangle.

   Based on the user's choice the program then asks him/her to enter the dimension(s) of the selected 2-D figure. Finally, these dimensions are then used by a class called `Area` to compute the area. Use overloaded constructors to compute this area.

4. Create a C++ class for player object with the following attributes player no., name, number of matches and number of goals done in each match. The number of matches varies for each player. Write parameterized constructor which initializes player no., name, number of matches and creates array for number of goals and number of matches dynamically.

5. Write a C++ program to display Pascal's triangle using the friend function.

6. Write a program to display Fibonacci series (upto a n places, n is entered by the user) using:

   - Operator Overloading.
   - Friend functions.

7. Write a program where you need to define a class called `strings` which performs the following functions:

   - Takes a string of characters as an input from the user and displays it on the screen (the string should not contain space).
   - Compares two strings and gives true when they are equal and gives false otherwise.
   - Concatenates two strings.

   Use operator overloading to design such a class.

8. Enter a set of numbers to a linked list. Then sort the linked list into both ascending and descending order.

9. Convert a decimal number to a binary number using multiple inheritance.

10. Write a program to display first n odd numbers and their sum using multiple inheritance.

11. Consider a Base class as defined below:

```
class fixDeposit
{
    protected:
    int accno; // account number
    protected:
    double amount; // principal amount
    public:

    fixDeposit(int a, double p)
    {
        accno = a;
        amount = p;
    }
    public:
    double interest()
    {
        cout << "The member function in fixDeposit");
        return 0;
    }
    public:
    void update(double d)
    {
        amount += d;
    }
    public:
    void display()
```

```
    {
    cout<<"The member function in fixDeposit");
    }
}
```

Drive classes (from base class fixDeposit)

(a) SIdeposit, and

(b) CIdeposit for accounts earning Simple Interest (SI), and Compound Interest (CI) respectively on the principal amount.

Besides inherited members, each derived should have additional data members for:

(a) yearly rate of interest, and

(b) time period of deposit in number of years.

Define following methods for each child/derived class:

- Constructors for the derived classes, explicitly calling base class constructors.
- The method 'interest' which overrides the method 'interest' of the base class, calculates and returns simple/compound interest.
- The function 'display' which overrides the method 'display' of the base class, updates the principal amount by adding interest and then displays the final amount. Write a test driver class and explain how the polymorphic feature in C++ is implemented in the above.

12. Modify the program in Q11. for auto-increment of account number of a customer (hint: use of static member, which will be shared by all the objects of the class) i.e. if 1 is account no. for 1st customer then 2 should be the account number of the second customer.

13. Modify the program in Q11 to make use of 'abstract' class and virtual methods for the same application and explain if any difference in implementation of Q12 and Q13. Are the results same?