

Assignment-29: Basics of C++

1. Write a C++ program to print **Hello MySirG** on the screen.
2. Write a C++ program to print **Hello** on the first line and **MySirG** on the second line using endl.
3. Write a C++ program to calculate the sum of two numbers.
4. Write a C++ program to calculate the area of a circle
5. Write a C++ program to calculate the volume of a cuboid.
6. Write a C++ program to calculate an average of 3 numbers.
7. Write a C++ program to calculate the square of a number
8. Write a C++ program to swap values of two int variables without using third variable
9. Write a C++ program to find the maximum of two numbers.
10. Write a C++ program to add all the numbers of an array of size 10.

Assignment-30: Functions in C++

1. Define a function to check whether a given number is a Prime number or not.
2. Define a function to find the highest value digit in a given number.
3. Define a function to calculate x raised to the power y.
4. Define a function to print Pascal Triangle up to N lines.
5. Define a function to check whether a given number is a term in a Fibonacci series or not.
6. Define a function to swap data of two int variables using call by reference
7. Write a function using the default argument that is able to add 2 or 3 numbers.
8. Define overloaded functions to calculate area of circle, area of rectangle and area of triangle
9. Write functions using function overloading to find a maximum of two numbers and both the numbers can be integer or real.
10. Write functions using function overloading to add two numbers having different data types.

Assignment-31: More on Functions in C++

1. Define a function to sort an array of integers in ascending or descending order depending on bool type argument (true for ascending and false for descending). Use default argument to implement it.
2. Define a function to sort an array of strings in ascending or descending order depending on bool type argument (true for ascending and false for descending). Use default argument to implement it.
3. Define a function to rotate an array by n positions in the d direction. Implement position and direction using default arguments. Argument d can be -1 or 1 denoting left or right direction. Argument n should be the last argument with default value 1.
4. Define a function to calculate LCM of three numbers.
5. Define a function to print all the prime factors of a given number. [for example num=36, prime factors are 2, 3]
6. Define a function to calculate HCF of two numbers.
7. Define overloaded functions to calculate volume of a cuboid, cone and sphere
8. Define a function to print a substring from startIndex(inclusive) to endIndex(exclusive). Define function in such a way that if second argument is not provided, string will print till the last possible index.
9. Define a function to swap two arrays.
10. Define a function to merge two sorted arrays of same size.

Assignment-32: Classes and Objects

1. Define a class Complex to represent a complex number. Declare instance member variables to store real and imaginary part of a complex number, also define instance member functions to set values of complex number and print values of complex number
2. Define a class Time to represent Time (like 3 hr 45 min 20 sec). Declare appropriate number of instance member variables and also define instance member functions to set values for time and display values of time.
3. Define a class Date to represent date (like d=31, m=12, y=2022). Declare appropriate number of instance member variables and also define instance member functions to set date and get date.
4. In question 3, define a methods to display date in the following pattern
 1. 31-12-2022
 2. 31-Dec-2022
5. Define a class Circle with radius as its property. Provide setRadius() and getRadius() methods. Also define methods to return area and circumference of circle.

Assignment-33: member function, static

1. Define a class Complex to represent a complex number with instance variables a and b to store real and imaginary parts. Also define following member functions
 1. void setData(int,int)
 2. void showData()
 3. Complex add(Complex)
 4. Complex subtract(Complex)
 5. Complex multiply(Complex)
2. Define a class Time to represent a time with instance variables h,m and s to store hour, minute and second. Also define following member functions
 1. void setTime(int,int,int)
 2. void showTime()
 3. void normalize()
 4. Time add(Time)
 5. bool is_greater(Time)
3. Define a class TestResult with properties roll_no, right, wrong, net_score. Also define class properties right_weightage, wrong_weightage. Provide methods to set and get all the properties.
4. Write a driver function main() to use TestResult class (Question 3). Create an array to 5 TestResult objects, set values to all the objects and display the results in sorted order (by net_score).
5. Define a class Matrix to represent a 3x3 order matrix. Provide appropriate methods and properties to the class. Also define following methods in the class:
 1. Matrix add(Matrix)
 2. Matrix sub(Matrix)
 3. Matrix multiply(Matrix)
 4. Matrix transpose()
 5. bool is_singular()

Assignment-34: constructor

1. Define a class Cuboid with length, breadth and height as instance member variables. Define constructors to initialise member variables.
2. Define a class Customer with instance members cust_id, name, email, mobile. Define non parameterised constructor and parameterised constructor in the class.
3. Define a class Time with hr,min,sec as instance member variables. Define constructor to initialise time object.
4. Define a class Book with booked, title and price as instance member variables. Define non parameterised and parameterised constructor in the class.
5. Define a class Complex with instance variables for real and imaginary part of a complex number. Define only one parameterised constructor in the class to initialise complex object. Also define showData method in the class to display object data. Now create an array of Complex class with size 5 and display values of each object.
6. Define a class Numbers with size and arr pointer as instance variables. Provide constructor to initialise instance members, dynamically allocate an array of given size and store address in arr variable of object. Also define destructor to deallocate memory of array. Implement deep copy using copy constructor to avoid memory issues.
7. Define a class student and write a program to enter student details using constructor and define member function to display all the details.
8. Define a class Date with d, m, y as instance variables. Initialise members using initialisers.
9. Define a class Room for a hotel management project with room number, room type, is_AC, price as instance variables. Initialise instance variables using constructor.
10. Define Circle class with radius as instance variable. Define two constructors in the class -non parameterised and parameterised.

Assignment-35: operator overloading

1. Define a class Complex with appropriate instance variables and member functions. Define following operators in the class:

1. +
2. -
3. *
4. ==

2. Create a class Time which contains hour, min and second as fields. Overload greater than. (>) operator to compare two time objects.
3. In Question-2, overload pre-increment and post-increment operator to increment Time object value by one second.
4. In Question-2, overload operator+ to add two Time objects.
5. Define a class matrix to represent 3x3 matrix. Provide appropriate instance methods. Also define operator +, operator -, operator * to perform addition, subtraction and multiplication operations respectively.

5. Consider following class Numbers

```
class Numbers
{
    int x,y,z;
public:
    // methods
};
```

Overload the operator unary minus (-) to negate the numbers.

6. Define a C++ class fraction

```
class fraction
{
```

```
    long numerator;  
    long denominator;  
public:  
    fraction (long n=0, long d=0);  
}
```

Define an operator + to add two fraction object.

7. In Question-6, define an operator < to compare two fraction objects.
8. Consider a class Distance

```
class Distance  
{  
    int km, m, cm;  
public:  
    //methods;  
};
```

Overload the operator+ to add two distance objects.

9. In Question-8, define operator pre decrement and post decrement to decrease distance by 1 cm.
10. Consider the following class Array

```
class Array  
{  
    int *p;  
    int size;  
public:  
    // methods  
};
```


Define constructor to allocate an array of given size (size is given through parameter).

Define a subscript operator to access element at given index. Define destructor to deallocates the memory of array.

Assignment-36: Operator Overloading, friend operator

1. Define a class Complex and overload following operators as a friend.
 1. +
 2. -
 3. *
2. Define a class Time with appropriate instance variables and member functions. Overload following operators
 1. << insertion operator
 2. >> extraction operator
3. In Question-2, over load operator= to perform copy of Time object.
4. Consider the following class Array

```
class Array
{
    int *p;
    int size;
public:
    // methods
};
```

Define operator = to perform deep copy of Array objects.

5. In Question 4, define a friend operator + to concatenate two Array objects.
6. In Question-1, provide a unary friend operator - to negate the real and imaginary part of a complex number.
7. Define a class Integer with instance variable of type int. Provide a friend logical operator not !. Also define a friend operator == to compare two Integer objects.
8. Create a Coordinate class with 2 instance variables x and y. Overload comma operator such that when you write c3 = (c1 , c2) then c2 is assigned to c3. Where c1, c2, and c3 are objects of 3D coordinate class.
9. In Question-8, provide overloaded insertion and extraction operators.

10. Define a class Student with roll no, name and age as instance variables. Create setStudent() and showStudent() methods in the class. Overload operator == to compare two Student objects.

Assignment-37: Pointer, this, DMA

1. Define a class Person with name and age as instance variables. Provide parameterised constructor with two formal arguments name & age, to initialise instance variables.
2. Define a class Complex with instance variables a and b to store real and imaginary part of a complex number. Provide setData() method with formal arguments with the name a and b, to set the values of instance variables. Also define showData() method to display instance member variable values.
3. In question-2, write a non member function in which instantiate Complex class dynamically. Initialise instance variables and display their values.
4. Define a class Distance with instance variables hr, min and sec. Provide instance methods setTime() and showTime(). setTime() method has formal arguments with the same name as instance variables. Also define setters and getters.
5. In question-4, define a method to dynamically create an array of Time objects with specified size (received through argument) and return array.
6. In question-4, define a method to sort Time object array.
7. In question-4, define a method to display Time object array values. Array is received through argument.
8. Define a class String with char pointer and length of the string as instance member variables. Define a method to dynamically create an array of char type and hold the input string in such a way that the array length is just enough to accommodate the input string.
9. In question-8, define methods to convert string into uppercase, lowercase.
10. In question-8, define methods to print string, return string, return length of string.

Assignment-38: Inheritance

1. Define a class Person with name and age as instance variables as private members. Provide setters and getters as protected methods. Derive a class Employee from Person class with private instance variable salary. Provide public methods setEmployee() and showEmployee().
2. Define a class Circle with radius as private instance variable and setRadius(), getRadius(), getArea() as public instance methods. Define a class ThickCircle as a subclass of Circle with thickness as private instance variable and getThickness(), setThickness() as public instance methods. Provide an overridden method getArea() to calculate area of thick portion of circle.
3. Define a class Coordinate with x and y as instance variables. Define overloaded versions of getDistance(), first with no argument calculating distance from origin, and second with one argument of Coordinate type to calculate distance between two coordinates. Also define other methods if required.
4. Define a class Shape with shapeName as instance variable. Provide setter and getter.
5. In question 3 & 4, derive a class StraightLine from Shape with two Coordinate type objects as instance members. Provide methods to setLine(), getDistance() and showLine().
6. Define a class Game with an array of 5 int variables as instance member to store the score of each of 5 rounds. Provide method to setScore(int round, int score). Also provide method getScore(int round).
7. In question 6, derive a class GameResult with an array of 5 int type variables to store the result in each round. Result value is 2 for win, 0 for loose and 1 for draw. Provide methods to set Result and getResult. Write method to calculate final result of the game.
8. Define a class Actor with name, age as instance variables and setter, getters as instance methods. Define a class TVActor as a derived class of Actor with instance variable to store number of TV projects done or running and define setter, getter. Also define setTVActor() and showTVActor(). Define a class MovieActor as a derived class of Actor with instance variable to store number of movies doe or running and define setter, getter. Also define setMovieActor() and showMovieActor().

C, C++ and DSA in Depth

Assignment-39: File Handling

1. Write a C++ program to copy content of file and store in other file.
2. Write a C++ program to search a word in a file.
3. Define a class `Employee` with `empid`, `name` and `salary` as instance variables. Also provide instance methods to input and display `Employee` record.
4. In question 3, define a member function to store new employee record in a file.
5. In question 3, define a member function to print all employee records stored in a file.
6. In question 3, define a member function to search an employee by his `empid` in a file. Print his record if found, other wise display a message of search failed.
7. In question 3, define a member function to edit employee data of given `empid`.
8. In question 3, define a member function to delete an employee record of given `empid`