Date:

Page No.

EX NO : 02	WRITE A POINT THAT REPRSENTS A 2-D POINT IN A PLANE. WRITE MEMBER FUNCTION TO (A, B, C)
DATE :	

AIM :

To Write a c++ program, a paint class that represent a 2-D paint in a plane.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : This class will have attributes to represent the x and y Coordinates of the paint.

STEP 3 : Initialize a paint with x and y.

STEP 4 : Set the x and y Coordinates of the paint.

STEP 5 : Display the paint in the form (x, y)

STEP 6 : Calculate the distance between two paints.

STEP 7 : Check whether two paints are equal.

STEP 8 : Test the class by Creating instances.

STEP 9 : End of the program

Date:

Page No.

EX NO : 03	DESIGN AND IMPLEMENT A CLASS THAT REPRESENTS A HARMONIC PROGRESSION (HP)
DATE :	IMPLEMENT FUNCTIONS TO DO THE FOLLOWING (A, B, C, D)

AIM:

To write a C++ program, design and implement a class that represents a harmonic progression (HP) and a implement functions.

ALGORITHM:

STEP 1: Start the program.

STEP 2: This class will handle the generation of the harmonic progression (HP) and its corresponding arithmetic.

STEP 3: Initialize the HP with base (first) term and the common difference for the corresponding AP.

STEP 4: Generate the first n terms of the harmonic progression.

STEP 5: Calculate the sum of the first n terms of the harmonic progression.

STEP 6: Calculate the sum of an infinte harmonic progression.

PAD

Date:

Page No.

STEP 7 : Calculate the nth term of the HP

STEP 8 : Generate the Corresponding AP from the HP.

STEP 9 : Test the Class by Creating instances.

STEP 10 : End of the program.

Date: Page No.

EX NO : 04	DESIGN AND IMPLEMENT A CLASS TO REPRESENT A SOLID OBJECT
DATE :	

AIM:

To write a C++ program, design and implement a class to represent a Solid object.

ALGORITHM:

STEP 1: Start the program

STEP 2: This class will represent different types of 3D Solid objects like a Sphere, cylinder, cube or Cuboid.

STEP 3: Initialize the Solid object with the type and relevant dimensions.

STEP 4: Set or update the dimension of the Solid object.

STEP 5: Calculate the Volume of the Solid based on its type.

STEP 6: Calculate the Surface area of the Solid based on its type.

STEP 7: Display the type of Solid object.

STEP 8: Test the class by Creating instances.

STEP 9: End of the program.

Date:

Page No.

EX NO : 05	DESIGN A CLASS REPRESENTING TIME IN HH:MM:SS. WRITE A FUNCTIONS TO (A, B, C, D)
DATE :	

AIM :

To write a c++ program. design a class representing time in hh:mm:ss. write functions to duration of time.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Define the Constructor to initialize attributes hours, minutes and seconds.

STEP 3 : Set and Show time.

STEP 4 : Find the different between two time objects

STEP 5 : Add a given duration to a time object.

STEP 6 : Conversion of the time object to second.

STEP 7 : End of the program.

Date:

Page No.

EX. NO : 06	DESIGN A 3×3 MATRIX CLASS AND DEMONSTRATE THE FOLLOWING (A,B)
DATE :	

AIM :

To write a C++ program, design a 3×3 matrix class and demonstrate the addition and multiplication matrix.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Define init method.

STEP 3 : Define str method to display the matrix. This method will format the matrix for printing displaying it in errors.

STEP 4 : Addition of two matrices using Operator Overloading.

STEP 5 : Multiplication of two matrices using Operator Overloading.

STEP 6 : Maintain a Count of the number of object Created.

STEP 7 : End of the program.

PAD

Date:

Page No.

EX NO : 07	DESIGN A CLASS CALLED STRING TO REPRESENT A STRING DATA TYPE CREATE A DATA MEMBER IN THE CLASS TO REPRESENT A STRING USING AN ARRAY OF RIZE 100. WRITE THE FOLLOWING THE FUNCTIONALITY AS MEMBER FUNCTION
DATE :	

AIM :

To write a C++ program, design a class called String to represent a String data type. Create a data member in the class to represent a string using an array of size 100.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Define init Constructor.

STEP 3 : This method create a new cstring object by Copying the String from an existing object.

STEP 4 : Concatenate two Strings.

STEP 5 : Find the length of the String.

STEP 6 : Reverse the String.

STEP 7 : Compare two String.

STEP 8 : End of the program.

PAD

Date:

Page No.

EX NO : 08	DESIGN A CLASS CALLED STRING TO REPRESENT A STRING DATA TYPE. CREATE A DATA MEMBER IN THE CLASS TO REPRESENT A STRING WHOSE SIZE IS DYNAMICALLY ALLOCATED. WRITE THE FOLLOWING AS MEMBER FUNCTION (A, B, C, D, E, F)
DATE :	

AIM :

To write a C++ program, design a class called CString to represent a String data type Create a data member in the class to represent a String whose size is dynamically allocated.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Create a template class with a data member that represents an array of element.

STEP 3 : Implement exception handling for array bound Violation.

STEP 4 : Implement a Sorting String.

STED 5 : End of the program.

PAD

Date: Page No.

EX NO : 09	CREATE A CLASS TO REPRESENT A 2-D SHAPE AND DERIVE CLASSES TO REPRESENT A TRIANGLE, RECTANGLE, AND CIRCLE WRITE A PROGRAM
DATE :	USING RUN TIME POLYMORPHISM TO COMPUTE THE AREA OF ,THE FIGURE

AIM :

To write a c++ program, Create a class to represent a 2-D Shape and derive classes to represent a triangle, rectangle and Circle. write a program using run time polymorphism to Compute the area of the figure

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Define the base Class.

STEP 3 : Define the derived class triangle, rectangle and Circle.

STEP 4 : Implement run time polymorphism.

STEP 5 : End of the program.

PAD

Date: Page No.

EX. NO : 10	DEFINE A CLASS TEMPLATE REPRESENTING A SINGLE DIMENSIONAL ARRAY. IMPLEMENT A FUNCTION TO SORT THE ARRAY ELEMENTS INCLUDE A MECHANISM TO DETECT AND THROW AN EXCEPTION FOR ARRAY BOUND VIOLATIONS
DATE:	

AIM:

To write a C++ program, define a class template representing a single dimensional array implement a function to sort the array elements.

ALGORITHM:

STEP 1 : Start the program.

STEP 2 : Defing the class template for array.

STEP 3 : Implement function to insert elements.

STEP 4 : Implement a Sorting function.

STEP 5 : Create an exception class for array Violations.

STEP 6 : Handle Violations in the class.

STEP 7 : Integrate Sorting and Violation.

STEP 8 : Test the class.

STEP 9 : End of the program.

Date:

Page No:

EX NO : 11	DEMONSTRATE THE USE OF THE VECTOR STL CONTAINER
DATE :	IMPLEMENT A TELEPHONE DIRECTORY USING FILES

AIM :

To write a C++ program, demonstrate the use of the Vector STL Container. Implement a telephone directory using files.

ALGORITHM :

STEP 1 : Start the program.

STEP 2 : Define a Constructor for initializing Contact data. Implement a method to write Contact data to a file and read from a file.

STEP 3 : Use STL Vector to maintain a dynamic list of Contact objects.

STEP 4 : Implement file handling for the telephone directory.

STEP 5 : Implement the telephone directory Operations.

STEP 6 : End of the program.

Date:

Page No.

RESULT :

Thus, the above program has been executed Successfully.

Date: Page No.

EX NO: 01	WRITE A CLASS TO REPRESENT A COMPLEX NUMBER WHICH HAS MEMBER FUNCTIONS TO DO THE FOLLOWING (A, B, C)
DATE:	

AIM:

To Write a c++ program of a class to represent a Complex Number which has member functions.

ALGORITHM:

STEP 1: Start the program.

STEP 2: Define the class Complex Number.

STEP 3: Define the init method that have two arguments, real and imaginary.

STEP 4: Set the real and imaginary parts of the Complex Number.

STEP 5: Display the Complex number in a readable format ("a + bi")

STE 6: Add two Complex Numbers.

STEP 7: Subtract One Complex Number from another.

STEP 8: Multiply two Complex Numbers.

STEP 9: Multiply the Complex Number by a Scalar Value.

STEP 10: Test the class by Creating instances.

STEP 11: End of the program.