Assignment -33 **A Job Ready Bootcamp in C++, DSA and IOT** **MySirG**

Virtual Function, Abstract Class

1. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called triangle and rectangle from the base shape. Add to the base class, a member function get_data() to initialise base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.
   Using these three classes, design a program that will accept dimensions of a triangle or a rectangle interactively, and display the area.
   Remember the two values given as input will be treated as lengths of two sides in the case of rectangles, and as base and height in the case of the triangles, and used as follows:
   Area of rectangle = x * y Area of triangle = 1/2 * x * y
2. Extend the above program to display the area of circles. This requires the addition of a new derived class 'circle' that computes the area of a circle. Remember, for a circle we need only one value, its radius, but the get_data() function in the base class requires two values to be passed. (Hint: Make the second argument of get_data() function as a default one with zero value.)
3. Using the concept of pointers, write a function that swaps the private data values of two objects of the same class type.
4. Create a base class called shape. Use this class to store 2 double type values that could be used to compute the area of figures. Derive 2 specific classes called triangle and rectangle from the base shape. Add to the base class a member function get_data() to initialise base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in derived classes to suit their requirements. Using these 3 classes, design a program that will accept the dimensions of the shapes interactively and display area.
5. Create a base class called Photon. Use this class to store double type value of wavelength that could be used to calculate photon energy. Create class calculate_photonEnergy which will photon energy.
   Using these classes, calculate photon energy.
6. Extend above to display the area of circles. For a circle,only one value is needed i.e. radius but in get_data() function 2 values are passed.
7. Create a base class called Matrix. Use this class to store 4 int type values that could be used to calculate determinants and create matrices. Create class calculate_determinant which will calculate the determinant of a matrix.
   Using these classes, calculate the determinant of the matrix.
8. Create a base class called proof. Use this class to store two int type values that could be used to prove that triangle is a right angled triangle. Create a class **compute** which will determine whether a triangle is a right angled triangle.

Using these classes, design a program that will accept dimensions of a triangle, and display the result.

(Summary: Prove that triangle is a right angled triangle using pythagoras theorem).

9. Create a base class called volume. Use this class to store two double type values that could be used to compute the volume of figures. Derive two specific classes called cube and sphere from the base shape. Add to the base class, a member function get_data() to initialise base class data members and another member function display_volume() to compute and display the volume of figures. Make display_volume() as a virtual function and redefine this function in the derived classes to suit their requirements.

   Using these three classes, design a program that will accept dimensions of a cube or a sphere interactively, and display the volume.

10. Create a base class called shape. Use this class to store two double type values that could be used to compute the area of figures. Derive two specific classes called square and parallelogram from the base shape. Add to the base class, a member function get_data() to initialise base class data members and another member function display_area() to compute and display the area of figures. Make display_area() as a virtual function and redefine this function in the derived classes to suit their requirements.

    Using these three classes, design a program that will accept dimensions of a square or a parallelogram interactively, and display the area.