OOP With C++ (Assignment - 4)
CSE, 3rd Semester
Deadline October 22, 2020

Prepared By: Deepak Uniyal (Assistant Professor CSE, GEU)

Note -

- Create your GitHub profile as taught in lectures and then push all your programs to folders named according to assignment. Example - when you push codes of this assignment, they should be inside Assignment3 folder.
- Please keep in mind that you don't commit all the codes together. Keep on committing codes module wise or question wise whatever seems available.

1. Discuss the ways using suitable examples in which inheritance promotes software reuse, saves time during program development and helps prevent errors.
2. Draw an inheritance hierarchy for students at a university. Use Student as the base class of the hierarchy, then include classes UndergraduateStudent and GraduateStudent that derive from Student . Continue to extend the hierarchy as deep (i.e., as many levels) as possible. For example, Freshman, Junior and Senior might derive from UndergraduateStudent , and DoctoralStudent and MastersStudent might derive from GraduateStudent . After drawing the hierarchy, discuss the relationships that exist between the classes. Also, implement the above hierarchy using C++ code.
3. Draw an inheritance hierarchy for classes Quadrilateral, Trapezoid, Parallelogram, Rectangle and Square. Use Quadrilateral as the base class of the hierarchy. Make the hierarchy as deep as possible. Also, implement the above hierarchy using C++ code and create functions to calculate perimeter and area of the shape.
4. Create a base Shape class which is inherited by TwoDShape and ThreeDShape and these shapes are further extended by their derived classes like square, rectangle, sphere etc. You should provide necessary methods like calculateArea, calculateVolume in these shapes.
5. Create any base class as per your wish and extend that into other derived class. You should be able to understand the calling order or default constructors, parameterised constructors and destructors of all the classes. You should implement hybrid inheritance in this example.

Reference - C++ How To Program By Deitel and Deitel