| **Assignment # 11**  ***Session*: Spring 2022 *Total marks*: 100**  ***Name*** : ***\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Roll number* : \_\_\_\_\_\_\_\_\_\_\_\_\_** |
| --- |

***Submission:***

• *Email instructor or TA if there are any questions. You cannot look at others’ solutions or use others’ solutions, however, you can discuss it with each other. Plagiarism will be dealt with according to the course policy.*

*• Submission after due time will not be accepted.*

**There should be a Report explaining your code and highlighting results. Follow this naming convention for your report RollNumber\_Assignment#.pdf e.g BSEE21001\_Assignment11.pdf.**

**Note:** From this assignment onwards, you will create default and parameterized constructor/s and destructor for every class.

**TASK 1:**

You are going to make an application for drawing different Geometrical shapes. You should break this application into objects and classes. In this Question, you are going to make one base class by the name Shape. Shape will have Surface Area (SA), Volume(V), Pi and R as attributes. Make these attributes private. This class will have two virtual member functions(public) set/get Volume (parameters will be Pi and R) and set/get Surface Area (Parameters will be Pi and R). You will have to write default constructor and destructor for this class as well.

|  |
| --- |

**TASK 2:**

In this task, you will make a class Sphere. Use public as access specifier and inherit form Shape. This class will have its own following protected attributes Pi, R, SA and V (Why aren’t we reusing the shape class attributes?). Override the set Volume and Surface Area member functions from Shape class with same parameters but with different implementations to calculate the Volume and Surface Area. You will also implement setters and getters for this class along with constructor and destructor.

Formulas:

SA = 4 \* Pi \* R \* 22

V= 4/3 \* Pi \* R3

|  |
| --- |

**TASK 3:**

In this Question, you will make a class Cone. Use public as access specifier and inherit from Sphere instead of Shape. This class will have two private attributes s and h (Why are we reusing Sphere class’s attributes) . Now override the functions of SA and V along with setters/getters of its own. Notice that SA and V will have different parameters this time so will you be able to override them without declaring a virtual method?

**Formulas:**

**SA = Pi \* R \* S + Pi \* R2**

**V = 1/3 \* Pi \* R2 \* H**

You must call these overridden functions by the object of shape. Your code must look like given below:

**Shape \*s;**

**Sphere sp;**

**s= &sp;**

**s->set\_SA(parameters)**

Execute the above code before and after using the virtual keyword with class members and report what difference do you see after using the virtual keyword.

//we know A C++ virtual function is a member function in the base class that you redefine in a derived class. It is declared using the virtual keyword. It is used to tell the compiler to perform dynamic linkage or late binding on the function. So by using virtual keyword me can access the function with the same name in parent and derived class.

|  |
| --- |