

Project 03, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Inheritance and Interface

In this project, you will use inheritance to design geometric shapes and interface to implement their common behaviors that can be performed on them. Create a project in Eclipse named **Project_03**. You are going to design multiple classes and interfaces in this project.

The main idea here is that higher dimensional geometric shapes inherit data from lower dimensional shapes. For example, a cube is a three-dimensional square, a sphere is a three-dimensional circle, and a glome is a four-dimensional circle. The circle, sphere, cylinder, and glome all share the attribute radius. The square and cube share the attribute side length. Follow the interfaces and inheritance described in the tables below. All shapes inherit getName() from the superclass Shape and Circle and Square classes override it. The equals() method compares based on the shape properties (instance variables) and toString() method returns a string with the name of the shape and value of the instance variables such as "Circle [Radius: 5.0]"

Specification:

Your program will consist of the following classes: **Shape, Circle, Square, Cube, Sphere, Cylinder, and Glome** and two interfaces: **Area and Volume**.

Your classes may only have the instance variables and methods specified in the table below and the methods defined in the two interfaces Area and Volume. You will implement the methods specified in the Area and Volume interfaces and have them return the appropriate value for each shape. Class Shape is an **abstract class** and will have a single public abstract method called getName that returns a string. Write accessor and mutator methods for all instance variables for each class where applicable.

Interface	Methods
Area	double getArea()
Volume	double getVolume()

Class	Instance Variable	Constructor	Methods	Extends	Implements
Shape	String name	Shape(String name)			
Circle	double radius	Circle(double radius, String name)	Overrides getName, equals and toString	Shape	Area
Square	double side	Square(double side, String name)	Overrides getName, equals and toString	Shape	Area
Cylinder	double height	Cylinder(double height, double radius, String name)	Overrides equals and toString	Circle	Volume
Sphere	None	Sphere(double radius, String name)	Overrides toString	Circle	Volume
Cube	None	Cube(double side, String name)	Overrides toString	Square	Volume
Glome	None	Glome(double radius, String name)	Overrides toString	Sphere	Volume

Note: the volume of a glome is $0.5(\pi^2)r^4$ where r is the radius.

JUnit Tester:

Write JUnit test classes to test all the classes (not for the abstract classes) you defined. You should thoroughly test all the classes and their methods.

Javadoc Style Comments: You have to make Javadoc style comments for all classes and methods including parameter and return description. Include the Javadoc folder "doc" with your code submission.

Important Note: Make sure the file names are correct and code is well commented. No late submission. You will get zero for late submission.

Submission:

Submit java files and Javadoc folder "doc" to the appropriate submission folder on the Canvas by the due time. You can zip all the java files and "doc" folder together and submit one zip file.

Project 03, Object-Oriented Programming, CSE 271, Spring 2020
Department of Computer Science and Engineering, Miami University

Inheritance and Interface

Grading Rubric:

Shape (Abstract Class)	
Abstract class and abstract method declaration	2
Constructor	1
getName() method declaration	1
Volume (Interface)	
Interface declaration	1
getVolume() abstract method	1
Area (Interface)	
Interface declaration	1
getArea() abstract method	1
Circle	
Inherits Shape and implements Area	2
Constructor	1
Accessors and Mutators (each worth 1 point)	2
getName() method	1
equals() method	3
toString() method	2
getArea() method	3
Square	
Inherits Shape and implements Area	2
Constructor	1
Accessors and Mutators (each worth 1 point)	2
getName() method	1
equals() method	3
toString() method	2
getArea() method	3
Cylinder	
Inherits Circle and implements Volume	2
Constructor	1
Accessors and Mutators (each worth 1 point)	2
equals() method	3
toString() method	2
getVolume() method	3
Sphere	
Inherits Circle and implements Volume	2
Constructor	1
toString() method	2
getVolume() method	3
Cube	
Inherits Square and implements Volume	2
Constructor	1
toString() method	2
getVolume() method	3
Glome	
Inherits Sphere and implements Volume	2
Constructor	1
toString() method	2
getVolume() method	3
Testing using JUnit	
Test constructors and getName()	4
Test equals and toString methods (each worth 1 point)	9
Test getArea() and getVolume() methods (each worth 1 point)	6
Javadoc	
Javadoc style comments (for each method 0.25 points and for each class and interface 0.25)	8
Submitted correct Javadoc files (only deduction)	(-5)
Total	100