**Object-Oriented programming Project**

|  |  |  |  |
| --- | --- | --- | --- |
| Name: |  | Department: |  |
| Student ID: |  | Room Number: |  |
| Due Date: | **May 31, 23 : 59** | | |

* **Format of File name**: ProjectOne.zip
* The zip file will contain **two** files, namely report file with file format “**ProjectOneReport** “ to write class diagrams using ML and Source **code file that contains the codes of all classes.**

1. **General Objective**: To write a Java program that includes the basic concepts of Object Oriented programming such as
2. Data Encapsulation: Enclosing both data and method in one class
3. composition(HAS-A relation between two classes )
4. Inheritance (IS-A relationship between parent class and child class)
5. **Polymorphism (IS- A relation between parent class and child class)**
6. **Method Overriding**
7. **Exception Handling**

**II: Specific Objectives**

1. To write code that requests a user to enter input from key board and display output on screen
2. To write a java application that includes Interface, supper class ,subclass, and method overriding principles

3. To write the code of a method that checks the validity of input using Exception handling concept

**Part 1: Problem Description to Simulate Computation of Geometrical Shapes**

Suppose we want to develop a java application that computes the area and the volume of several types of geometrical shapes for a mathematical study. Geometrical shapes are composed of varices. Geometrical shapes can be two dimensional (**2D**) or three dimensional (3D).

1. **2D Shapes**: Two dimensional shape (2D) has **no height**. Hence, we can simulate only the computation of its surface area. These shapes can be represented by its **vertices (2D points)** in **two dimensional** space.
2. **3D shapes**: Based on the number of base surface areas, three dimensional (**3D**) geometrical shapes can be pyramid or prism.

**B.1) Prism:** It has base surface area and top surface area (both surfaces are identical). Prism has normal height between its base surface and its top surface. It has at least three or more parallelogram lateral surfaces. We can simulate the computation of total surface area (base surface area+ top surface area+ lateral surface area).

**B.2) Pyramid**:Instead of top surface area, a pyramid has **apex point** A(x,y,z) directly above the center of the base area of the pyramid. Hence, **pyramid** has only one **base** surface area. It has **normal height** from its **apex point** to the center of its base area. It has also **slant height** from apex point to the vertices of its base surface.

* **The detailed description of the following six geometrical shapes is given below.**

1. **2D shapes:**

**1.1) Equilateral Triangle** hasthree vertices (points): **V1**(x1, y1), **V2**(x2,y2) and ***V3***(X3,y3).

**Functional requirement:**

* Request the user to enter the coordinates of the three vertices of the triangle and assign these values to the member variables of this class by **setter** methods.
* **Compute the following quantities using the formula in part 3**
* Compute the length of the three sides of the equilateral triangle
* Compute the surface area of the equilateral triangle using the three sides
* **Override** the **toString()** method of **Object** class to display the information as shown in the **output** of the program(**See part 2**)

**Non-functional requirement (exception handling):**

* If the three vertices are collinear (**lie on the same line**), they cannot form a triangle **because** these points form a **straight line**. Check this by comparing **their slopes**. Hence, the program should display the error message and request the user to enter **non-collinear** vertices
* If the three vertices (points) **are not** **collinear** and if the three sides **are not** equal, **display warning message** to user and request the user to **continue**.

**1.2) Square has four vertices, namely :**upperLeft, upperRight and LowerLefrt and LowerRight: **UL**(x1,x2),**UR**(x2,y2),**LL**(x3,x3) and **LR**(x4,y4). However, UL and LR vertices are enough to compute **surface area** of the square.

**Functional requirement:**

* Request the user to enter the coordinates of the **UpperLeft** and **LowerRight** vertices of the square and assign these values to member variables of this class by **setter** methods.
* **Compute the following quantities using the formula in part 3**
* Compute the length of the side of the square from these vertices.
* Compute the surface area of the square using its side.
* **Override** the **toString(**) method of **Object** class to display the information as shown in the **output** of the program( **See part 2)**

**Non-functional requirement (exception handling):**

* If the upper left vertex **UL**(x1,y1) and lower right vertex **LR**(x4,y4) are the same, the program should display the error message and request the user to enter **two different** vertices.
* if **x4** is less than **x1,** the program should display the error message and request the user to enter correct vertices
* If **y4** is less than **y1**, the program should display the error message and request the user to enter correct vertices

1. **Prism(3D shapes)**

**2.1) EquilaterlaTriangularPrism** has equilateral triangular shape at its base with three vertices and a normal height**:**

**Functional requirement:**

* Request the user to enter the coordinates of the three vertices of the triangle (the base shape of the prism) and assign these values to member variables of this class by setter methods.
* **Request the user to enter the normal height of the triangular prism and assign this value to a member variable of this class.**
* Compute the length of the three sides of the equilateral triangular base of the prism.
* **Compute the following quantities using the formula in part 3**
* Compute the equilateral triangular **base area** of the prism from its three sides
* **Compute the perimeter of the equilateral triangular base of the prism**
* **Compute the lateral surface area of the equilateral triangular prism from the perimeter of the base of the prism and normal height of the prism.**
* **Compute the surface area of the equilateral triangular prism** by adding its base area, top area and lateral surface area. **Note** that the top area of the prism is the same as its base area.
* **Compute the volume of the equilateral triangular prism from base area and normal height of prism**
* **Override the toString() method of EquilateralTriangle class to display the information as shown in the output of the program**( **See part 2)**

**Non-functional requirement (exception handling):**

* If the tree vertices are collinear (lie on the same line), they cannot form a triangle because these points form a straight line. Check this by comparing their slopes. Hence, the program should display the error message and request the user to enter non-collinear vertices
* If the three vertices (points) are not collinear and if the three sides are not equal, **display warning message** to user and request the user to continue.
* **If the height of the prism is zero or negative, the program should display the error message and request the user to enter valid height**
  1. **SquarePrism** has square shape at its base with upper left and lower left vertices as well as **normal** height.

**Functional requirement:**

* Request the user to enter the coordinates of the **UpperLeft** and **LowerRight** vertices of the square (base of the prism)and assign these values from the member variables of this class.
* **Request the user to enter the normal height of the square prism** and assign the value to the member variable of this class.
* **Compute the following quantities using the formula in part 3**
* Compute the side of the square from these vertices.
* Compute the **base surface** area of the square(base of prism) from its side
* **Compute the perimeter of the square(base of the prism)from its side**
* **Compute** the **lateral surface** area of the square(base of prism) from perimeter of the base of the prism and **normal height of the prism**.
* Compute the **surface area** of the square prism by adding its base area, top area and lateral surface area. Note that the top area of the prism is the same its base area.
* Compute the **volume** of the square prism from base area and **normal height** of prism
* Override the **toString**() method of **square class** to display the information as shown in the **output** of the program( **See part 2).**

**Non-functional requirement (exception handling):**

* If the upper left vertex **UL**(x1,y1) and lower right vertex **LR**(x4,y4) are the same, the program should display the error message and request the user to enter **two different** vertices.
* if **x4** is less than **x 작거나 같을 때 1,** the program should display the error message and request the user to enter correct vertices
* If **y1** is less than **y4(작거나 같을 때)**, the program should display the error message and request the user to enter correct vertices
* **If the height of the prism is zero or negative, the program should display the error message and request the user to enter valid height**

**3. Pyramid (3D)**

* 1. **EquilateralTriangularPyramid** has equilateral triangular shape at its base which three vertices. In addition to normal height, the pyramid has slant **Functional requirement:**
* Request the user to enter the coordinates of the three vertices of the triangle (base of the pyramid) and assign these values to member variables of this class.
* **Request the user to enter the normal height of the triangular pyramid and assign the value to member variable of this class.**
* **Request the user to enter the slant height of the triangular pyramid and assign the value to member variable of this class**
* **Compute the following quantities using the formula in part 3**
* Compute the length of the three sides of the equilateral triangle(base shape of the pyramid)
* Compute the triangular **base area** of the pyramid from its three sides
* **Compute the perimeter of the equilateral triangle (base of the pyramid)**
* **Compute the lateral surface area of the triangular pyramid from perimeter and slant height**
* **Compute the surface area of the triangular pyramid** by adding its base area and lateral surface area. **Note that** pyramid has only base area and it has no top area.
* **Compute the volume of the triangular pyramid from base area and normal height of pyramid**
* **Override the toString() method of EquilateralTrianglePrisim class to display the information as shown in output of the program**(**See part 3).**

**Non-functional requirement (exception handling):**

* If the tree vertices are collinear (lie on the same line), they cannot form a triangle because these points form a straight line. Check this by comparing their slopes. Hence, the program should display the error message and request the user to enter non-collinear vertices
* If the three vertices (points) are not collinear and if the three sides are not equal, **display warning message** to user and request the user to continue.
* If the normal height of the pyramid is zero or negative, the program should display the error message and request the user to enter valid height
* If the normal height is greater than or equal to slant height, the program should display the error message and request the user to enter valid normal height and slant height.
  1. **SquarePyramid** has square shape at its base with upper left and lower left vertices as well as **normal** height and slant height.

**Functional requirement:**

* Request the user to enter the coordinates of the **UpperLeft** and **LowerRight** vertices of the square (base of the prism) and assign these values to member variables of this class
* Request the user to enter the normal height of the square pyramid and assign this value to member variable of this class
* **Request the user to enter the slant height of the square pyramid and assign the value to member variable of this class**
* **Compute the following quantities using the formula in part 3**
* Compute the sides of the square from these vertices.
* Compute the **base surface** area of the square from its side
* Compute the perimeter of the square(base of the pyramid)
* Compute the **lateral surface** area of the triangular prism from the **perimeter and slant height**.
* Compute the **surface area** of the triangular pyramid by adding its base area and lateral surface area. Note that pyramid has no top surface area. It has only base surface area.
* Compute the **volume** of the square prism from base area and **normal height** of prism
* Override the **toString**() method of **SqarePrism** class to display the information as shown in the **output** of the program(**See part 2).**

**Non-functional requirement (exception handling):**

* If the upper left vertex **UL**(x1,y1) and lower right vertex **LR**(x4,y4) are the same, the program should display the error message and request the user to enter **two different** vertices.
* if **x4** is less than **x1,** the program should display the error message and request the user to enter correct vertices
* If **y4** is less than **y1**, the program should display the error message and request the user to enter correct vertices
* **If the height of the prism is zero or negative, the program should display the error message and request the user to enter valid height.**
* **If the normal height is greater than or equal to slant height, the program should display the error message and request the user to enter valid normal height and slant height.**

**Summary: The program contains the following classes and interface**

1. **“Poin2D” class:** A concrete class called **point2D class** is needed to represent 2D points. The point2D class has X-coordinate and Y-coordinate member variables. The point2D class should be used as **member field** for 2D class (triangle and square) using the principle of composition (has-a relationship).
2. **“Shapes” Interface: This interface has two methods**

* **getDistance()** method to calculate the distance between two points in two dimensional coordinate space. This method computes the length of the side of an equilateral triangle and the sides of **the square.**
* **getArea() method to calculate the surface area of 2D shapes**

1. Equilateral triangle and square classes: these classes implement Shape Interface
2. Equilateral triangular prism and square prism extend Equilateral triangular and square respectively.
3. Equilateral triangular pyramid and square pyramid extend Equilateral triangular prism and square prism respectively

f) **TestShapes** class:

* The test class contains the **main** () method of the application.
* In this class, do the following operations for each of the six classes.

1. Write a code that request the user to create an object of a classes by using non-parametrized constructor(see the input format in part 2)
2. Write a code that request the user to enter the member **fields** of the class and assign these values to member fields by appropriate setter () method of the class.
3. Write a code that displays the output in the given format(see the out format in part 2)
4. **Additional Implementation requirements that are common to all classes.**

* **Add** the proper **member fields** for each class
* Each member field should have **privat**e access modifier
* In each class, each member field should has **setter and getter** methods.
* Include **exception handling codes that checks the validity of user input.**

**Q1.** From the description of the problem, represent the problem by designing class diagrams using UML (unified modeling language). The class diagram should include the three relationships among classes, namely Uses-A relationship, Has-A relationship and IS-A relationship. The class diagram should also include the data type of member variables, the return type of member methods**.**

**Q2**. Using your class diagrams in Q1, implement your program by applying principle of inheritance without using polymorphism as well as by applying principle of inheritance using polymorphism.

**Part 2: Input and output format of the program**

|  |
| --- |
| **Input Format**  === Equilateral Triangle ===  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 2  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 3  Please input y-coordinate of the Vertex 3 of the triangle: 0  Error: the three vertices are collinear. Please enter non-collinear vertices  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 3  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 1.5  Please input y-coordinate of the Vertex 3 of the triangle: 2  Warning: the three sides are not equal, but continue your operation.  === Square ===  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 0  Please input y-coordinate of the Lower Right of the Square : 1  Error: x-coordinate of upper left vertex should be less than x-coordinate of lower right vertex.  Please enter valid coordinates.    Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  Error: y-coordinate of upper left vertex should be less than y-coordinate of lower right vertex.  Please enter valid coordinates.  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 1  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  === Equilateral Triangular Prism ===  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 1  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 2  Please input y-coordinate of the Vertex 3 of the triangle: 0  Error: the three vertices are collinear. Please enter non-collinear vertices  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 3  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 1.5  Please input y-coordinate of the Vertex 3 of the triangle: 2  Warning: the three sides are not equal, but continue your operation.  Please **enter** normal height: -5  Error: Normal height should be positive  Please enter positive value for normal height: 5  === Square Prism ===  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 0  Please input y-coordinate of the Lower Right of the Square : 1  Error: x-coordinate of upper left vertex should be less than x-coordinate of lower right vertex.  Please re-enter valid coordinates.    Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  Error: y-coordinate of upper left vertex should be less than y-coordinate of lower right vertex.  Please enter valid coordinates.  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 1  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  Please input Normal Height: -5  Error: normal height should be positive  Please input positive value for height: 5  === Equilateral Triangular Pyramid ===  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 2  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 4  Please input y-coordinate of the Vertex 3 of the triangle: 0  Error: the three vertices are collinear. Please enter non-collinear vertices  Please input x-coordinate of the Vertex 1 of the triangle: 0  Please input y-coordinate of the Vertex 1 of the triangle: 0  Please input x-coordinate of the Vertex 2 of the triangle: 3  Please input y-coordinate of the Vertex 2 of the triangle: 0  Please input x-coordinate of the Vertex 3 of the triangle: 1.5  Please input y-coordinate of the Vertex 3 of the triangle: 2  Warning: the three sides are not equal, but continue your operation.  Please input Normal Height: -5  Error: Normal height should be positive  Please input positive value for height: 5  Please input Slant Height: 4  Error: Slant Height should be larger than normal Height.  Please input correct Slant Height: 7  === Square Pyramid ===  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 0  Please input y-coordinate of the Lower Right of the Square : 1  Error: x-coordinate of upper left vertex should be less than x-coordinate of lower right vertex.  Please enter valid coordinates.    Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 0  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  Error: y-coordinate of upper left vertex should be less than y-coordinate of lower right vertex.  Please enter valid coordinates.  Please input x-coordinate of the Upper Left of the Square : 0  Please input y-coordinate of the Upper Left of the Square : 1  Please input x-coordinate of the Lower Right of the Square : 1  Please input y-coordinate of the Lower Right of the Square : 0  Please input Normal Height: -5  Error: normal height should be positive  Please enter positive value for normal height: 5  Please input Slant Height: 4  Error: Slant Height should be larger than normal Height.  Please input correct Slant Height: 7 |

|  |
| --- |
| **Output format**  Type of shape: Equilateral Triangle  Coordinates of Vertex One: 0.0, 0.0  Coordinates of Vertex Two: 3.0, 0.0  Coordinates of Vertex Three: 1.5, 2.0  Length of side 1: 3.0  Length of side 2: 2.5  Length of side 3: 2.5  Surface Area: 3.0  Type of shape: Square  ●coordinates of upper left vertex : 0.0, 1.0  ●coordinates of lower right vertex : 1.0, 0.0  ●Side of a square: 1.00  ●Surface Area: 1.00  Type of shape: Equilateral Triangular Prism  ●coordinates of Vertex One : 0.0, 0.0  ● coordinates of Vertex Two: 3.0, 0.0  ● coordinates of Vertex Three : 1.5, 2.0  ● Length of side 1: 3.0  ● Length of side 2: 2.5  ● Length of side 3: 2.5  ● Normal Height: 5.0  ● Surface Area: 46.0  ● Volume: 15.0  Type of shape: Square Prism  ●coordinates of upper left vertex: 0.0, 1.0  ● coordinates of lower right vertex: 1.0, 0.0  ● Side of a square: 1.00  ● Normal Height: 5.0  ● Lateral Surface Area: 20.00  ● Surface Area: 22.00  ● Volume: 5.00  Type of shape: Equilateral Triangular Pyramid  ● coordinates of vertex One : 0.0, 0.0  ● coordinates of vertex Two : 3.0, 0.0  ● coordinates of vertex Three: 1.5, 2.0  ● Length of side 1: 3.0  ● Length of side 2: 2.5  ● Length of side 3:  ● Normal Height: 5.0  ● Slant Height: 7.0  ● Surface Area: 31.0  ● Volume: 5.0  Type of shape: Square Pyramid  ● coordinates of upper left vertex : 0.0, 1.0  ● coordinates of lower right vertex: 1.0, 0.0  ● Side of a square: 1.00  ● normal height: 5.0  ● Slant Height: 7.0  ● Surface Area: 15.00  ● Volume: 1.67 |

**Part 3: Formula for surface area and volume of shapes.**

1. **Surface area of 2D shapes**

* **Surface area of circle(SAC): SAC** = PI\*r2,r is radius of circle
* **Surface area of equilateral triangle**(**SAT):**

**SAT** = **,**

**Where S**1 ,s2 and s3 are length of three sides of the triangle **and p is** perimeter of the triangle divided by 2( p =(s1+s2+s3)/2 ).

* **surface area of square(SAS): SAS** = s2 (s= side of the square);

1. **Surface area of 3D prisms:** 2\***Base area +P\*h**
2. **Surface area of triangular prism(SATPR)**

* SATPR= 2\***SAT**+**P\*nh**, where **p = s1+S2+S3 =3\*s**
* **SAT** is surface area of the equilateral triangle as the base
* **P** = perimeter of the equilateral triangle as the base
* **S** = the side of the equilateral triangle as a base.
* **nh** = the **normal height** of triangular prism (**not slant height**)

1. **Surface area of sq**uare prism (**SASPR**)

* **SASPR**= **2\*SAS+P\*nh**, where **p = s1+s2+s3+s4=4\*s.**
* **SAS** is surface area of the ***square as the base*** of square prism.
* **P** = perimeter of the square as a base of square pyramid
* **S** = the side of the square as a base.
* **nh** = Norma height of the square prism (**not slant height**)

1. S**urface area of 3D pyramids: Base area +1/2\*P\*h.**
2. **Surface area of Equilateral triangular pyramid (SATPY).**

* **SATPY**= **SAT**+**1/2\*P\*sh**, where **p = s1+S2+S3 =3\*s**
* **SAT** is surface area of the equilateral triangle as the base
* **P** = perimeter of the equilateral triangle as the base
* **S** = the side of the equilateral triangle as a base.
* **sh** = the **slight height** of the apex of the triangular pyramid

1. **Surface area of square pyramid(SASPY):**

* SASPY= **SAS**+1**/2\*P\*sh**, where **p = s1+s2+s3+s4=4\*s.**
* **SAS** is surface area of the ***square as the base*** of the pyramid
* **P** = perimeter of the square as a base of square pyramid
* **S** = the side of the square as a base.
* **sh** = the **slight height** of the apex of the square pyramid

1. **Volume of 3D prisms: Base Area\*height**
2. **Volume of equilateral triangular prism**:

* **VTPR= SAT\*nh**
* **SAT** is surface area of the equilateral triangle as the base
* **nh** = normal height of the triangular prism

1. **Volume of square prism:**

* **VSPR= SAS\*nh.**
* **SAS** is surface area of the ***square as the base*** of the pyramid
* **nh** = **normal** height of the square prism

1. **Volume of 3D pyramids: 1/3\*Base Area\*height**

1. **Volume of equilateral triangular pyramid(VTPY)**

* **VTPY= 1/3\*SAT\*nh**
* **SAT** is surface area of the equilateral triangle as the base
* **nh** = the normal height of the pyramid(**not slant** **height)**

1. **Volume of square prism(VSPY)**

* **VSPY= 1/3\*SAS\*nh.**
* **SAS** is surface area of the ***square as the base*** of the pyramid
* **nh** = the normal height of the apex of the square pyramid