

```

"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
Filename: main.py
This file demonstrates polymorphism with geometric shapes:
    It creates instances of `Cylinder`, prompts for information,
    and prints out the details of each cylinder.
"""

from Cylinder import Cylinder

def main():
    shapes = []

    try:
        cylinder = Cylinder()
        cylinder.ask()
        shapes.append(cylinder)

    except ValueError:
        print("Invalid input. Please enter valid numeric values.")

    if shapes:
        print("\nShape Information:")

        for shape in shapes:
            print(shape)

    else:
        print("\nNo shape information entered.")

if __name__ == "__main__":
    main()

"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
Filename: Shape.py
This file defines the abstract Shape class:
    It implements the `Shape` class, an abstract base class representing
    geometric shapes.
"""

from abc import ABC, abstractmethod

```

```

class Shape(ABC):
    def __init__(self, shapeName="", shapeColor=""):
        self._shapeName = shapeName
        self._shapeColor = shapeColor

    def getShapeName(self):
        return self._shapeName

    def setShapeName(self, shapeName):
        while not shapeName.isalpha():
            print("Error: Please enter a valid shape name (letters only).")
            shapeName = input("Enter the shape name: ")
        self._shapeName = shapeName

    def getShapeColor(self):
        return self._shapeColor

    def setShapeColor(self, shapeColor):
        while not shapeColor.isalpha():
            print("Error: Please enter a valid shape color (letters only).")
            shapeColor = input("Enter the shape color: ")
        self._shapeColor = shapeColor

    @abstractmethod
    def volume(self):
        pass

    @abstractmethod
    def ask(self):
        pass
"""
Author: Coral S. Schmidt Montilla
Student Number: 148830
Filename: Cylinder.py
This file defines the Cylinder class:
    It implements the `Cylinder` class, a concrete subclass of `Shape`,
    representing a cylinder.
"""

from Shape import Shape

class Cylinder(Shape):
    def __init__(self, shape_name="", shape_color="", radius=0.0, height=0.0):
        super().__init__(shape_name, shape_color)
        self._radius = radius

```

```

        self._height = height

def get_radius(self):
    return self._radius

def set_radius(self, radius):
    if radius >= 0:
        self._radius = radius
    else:
        raise ArithmeticError("Radius cannot be negative.")

def get_height(self):
    return self._height

def set_height(self, height):
    if height >= 0:
        self._height = height
    else:
        raise ArithmeticError("Height cannot be negative.")

def volume(self):
    return 3.141592653589793 * self._radius ** 2 * self._height

def ask(self):
    while True:
        try:
            self.setShapeName(input("Enter the shape name: "))
            self.setShapeColor(input("Enter the shape color: "))
            while True:
                radius_input = input("Enter the radius of the cylinder: ")
                try:
                    self.set_radius(float(radius_input))
                    break
                except ValueError:
                    print("Error: Please enter a valid numeric value for the
radius.")
            while True: # This loop is for the height input
                height_input = input("Enter the height of the cylinder: ")
                try:
                    self.set_height(float(height_input))
                    break
                except ValueError:
                    print("Error: Please enter a valid numeric value for the
height.")
        except ArithmeticError as e:

```

```

        print(f"Error: {e}")
    break # Break out of the main input loop when everything is
successfully entered
    except ValueError:
        print("Error: Please enter valid input.")
    except ArithmeticError as e:
        print(f"Error: {e}")
    except:
        print("An unexpected error occurred.")

def __str__(self):
    return f"Cylinder (Name: {self.getShapeName()}, Color:
{self.getShapeColor()}, Volume: {self.volume():.2f})"

```

Output:

```

r Science/Advanced Programming/Asig_3/main.py"
Enter the shape name: 123
Error: Please enter a valid shape name (letters only).
Enter the shape name: Cylinder
Enter the shape color: 93
Error: Please enter a valid shape color (letters only).
Enter the shape color: Blue
Enter the radius of the cylinder: d
Error: Please enter a valid numeric value for the radius.
Enter the radius of the cylinder: 234
Enter the height of the cylinder: -4
Error: Height cannot be negative.
Enter the height of the cylinder: 23456

```

Shape Information:

Cylinder (Name: Cylinder, Color: Blue, Volume: 4034925686.41)

PS C:\Users\coral\OneDrive\Desktop\Computer Science\Advanced Programming\Asig_3> █