In-class Exercise 5: Inheritance

For this exercise, you may work with another student. If you do, turn in **ONLY one copy** of the project with both names on it.

Turn in your project at the end of class and name it as follows:

- If you have not completed the whole implementation and the comments, name your file:
  A250_TEMP_E5_Yourlastname_Yourfirstname
  (or **A250_TEMP_E5_Yourlastname_Yourfirstname_Otherstudentlastname_Otherstudentfirstname**)
  The complete project is due **next week on Wednesday/Thursday**.

- If you have completed the entire exercise, name your file:
  A250_E5_Yourlastname_Yourfirstname
  (or **A250 _E5_Yourlastname_Yourfirstname_Otherstudentlastname_Otherstudentfirstname**)

Using the project **ex_05_inheritance**, implement **two (2) classes**, the **Circle** class and the **Cylinder** class as described below. The **Main.cpp** file already contains testing cases.

Class Circle

Write the **interface** in the **Circle.h** file, and write the **implementation** in the **Circle.cpp** file.

- **Member variables:**
  - A double named **radius** that stores the value of the radius of the circle.
  - A double named **pi** that stores the value of pi.
- **Default constructor:**
  - Initializes the radius to 0.0 and pi to 3.142.
- **Overloaded constructor:**
  - **Parameters:** a double that stores a new value for the radius, and a double that stores a new value for pi.
  - Initializes all member variables to the values passed by the parameters.
- Function **getRadius**
  - Returns the value of the radius.
- Function **getPi**
  - Returns the value of pi.
- Function **setRadius**
  - **Parameter:** A double that stores a new value for the radius.
  - Re-sets the value of radius to the value passed by the parameter.
- Function **setPi**
  - **Parameter:** A double that stores a new value for pi.
  - Re-sets the value of pi with the value passed by the parameter.
- Function **calculateArea**:
  - Returns the area of the circle as a double.
  - Area formula: **π * radius * radius**
- Function **printDimensions**:
  - Outputs the dimensions of the circle in the following format:

Radius: #
Pi: #

Where "#" will be replaced by the actual value.

- No need to format the decimals; the testing cases will take care of that.

- **Destructor**
  - Left empty.

Class Cylinder

Write the **interface** in the **Cylinder.h** file, which **inherits** from the class **Circle**, and write the **implementation** in the **Cylinder.cpp** file.

- **Member variable:**
  - A double named **height** that stores the value of the height of the cylinder.
- **Default constructor:**
  - Initializes the height of the cylinder to 0.0.
- **Overloaded constructor:**
  - **Parameters:** a double that stores a new value for the radius, a double that stores a new value for pi, and a double that stores a new value for the height.
  - Initializes its member variable to the variable passed by the parameter and calls the parent's overloaded constructor to pass the values of the parent's member variables. You need to use the **syntax** discussed in the slides.
- Function **getHeight**
  - Returns the value of the height.
- Function **setHeight**
  - **Parameter:** A double that stores a new value for the height.
  - Re-sets the value of height to the value passed by the parameter.
- Function **calculateVolume**:
  - Returns the volume of the cylinder as a double.
  - Volume formula: (**π * radius * radius) * height**
- Function **printDimensions**:
  - Re-defines the parent's function **printDimensions**.
  - To output the **radius** and **pi**, call the parent's function **printDimensions**.
  - Outputs the dimensions of the circle in the following format:

    Radius: #
    Pi: #
    Height: #

    Where "#" will be replaced by the actual values.
  - No need to format the decimals; the testing cases will take care of that.
- **Destructor**
  - Left empty.

Expected Output