



Module 12:

Inheritance

COP2274
In-class Assignments

M12A Class called Farm that inherits from class called Pig

1. Create a base class called **Pig** with a protected member variable called weight (double)
 - Define a default constructor that initializes the member variable as 100
 - Define a custom constructor that initializes the member variable with its parameter
 - Define a member function called *displayWeight()* to print the weight of one pig as shown in the test case

M12A Class called Farm that inherits from class called Pig

2. Create a class called **Farm** that inherits (public) from the class **Pig** with an extra private member variable called quantity (int)
 - Define a default constructor that initializes weight as 100 and quantity as 50
 - Define a custom constructor that initializes each member variable (weight or quantity) with its corresponding parameter
 - Define a member function to display the combined weight of all pigs on the farm as shown in the test case

M12A Class called Farm that inherits from class called Pig

3. In your main(), create two **Pig** and **Farm** objects by calling their default constructors and display the weight of one pig or the combined weight as shown in the test case. And then update the existing **Pig** and **Farm** objects by calling an anonymous object and display the updated weight(s) as shown in the test case

Test case

```
The weight of one pig is 100 Pounds  
The combined weight of 50 pigs: 5000 Pounds
```

```
The weight of one pig is 75 Pounds  
The combined weight of 10 pigs: 810 Pounds
```

M12B Class called Cuboid that inherits from class called Rectangle

1. Create a base class called **Rectangle** with two private member variables called length (int) and width(int)
 - Define a default constructor that initializes length and width as 0
 - Define a custom constructor that initializes each member variable with its corresponding parameter
 - Define a member function to find the area of a rectangle
 - Define a member function to print length and width as shown in the test cases

M12B Class called Cuboid that inherits from class called Rectangle

2. Create a class called **Cuboid** (representing a box) that inherits (public) from the class **Rectangle** with an extra private member variable called height
 - Define a default constructor that initializes length, width, height as 0
 - Define a custom constructor that initializes each member variable with its corresponding parameter
 - Define a member function to find the volume of a cuboid
 - Define a member function to print length, width, and height as shown in the test cases

M12B Class called Cuboid that inherits from class called Rectangle

3. Test the **Rectangle** and **Cuboid** classes in your main() by creating objects and hardcoding values as shown in the test cases

Test cases

```
Rectangle object 1:  
Length: 0  
Width: 0  
Area: 0  
Rectangle object 2:  
Length: 5  
Width: 4  
Area: 20  
  
Cuboid object 1:  
Length: 0  
Width: 0  
Height: 0  
Volume: 0  
Cuboid object 2:  
Length: 5  
Width: 4  
Height: 8  
Volume: 160
```

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

1. Define a class called **Scores** with two protected member variables, one for the number of assignment scores (int), as well as a pointer to a variable of type double (double *) for the dynamic array of assignment scores
 - Write a member function called *getScores()* to get the scores from the user, by allocating the dynamic array with size based on the number of assignments from the user.
 - Write a member function called *print()* to print the list of scores as shown in the test cases

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

2. Define a class called **Percentage** that inherits (public) from the class **Scores** with an extra protected variable for the percentage (double)
 - Write a member function called *calcPercentage()* to calculate the percentage of assignment scores (average) and update the member variable for the percentage.
 - Write a member function called *print()* to print the list of scores and the total percentage as shown in the test cases

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

3. Define a class called **Grade** that inherits (public) from the class **Percentage** with an extra private variable for the letter grade (string)
 - Write a member function called *calcGrade()* to calculate the letter grade based on the percentage of assignment scores and update the member variable for the letter grade.
 - Write a member function called *print()* to print the list of scores, the total percentage and the final letter grade as shown in the test cases

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

4. In your main(), create a default **Grade** object and then test all the member functions of the **Grade** object, *getScores()*, *calcPercentage()*, *calcGrade()*, and *print()* as shown in the test cases

Notes:

- *Do not declare/define any constructor for Scores, Percentage, or Grade class. If you don't declare/define constructors, the compiler automatically create a default constructor for you.*

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

Notes:

- You may use the values below to calculate the letter grade.
A = 90.0-100, **A-** = 87.0-89.9, **B+** = 84-86.9, **B** = 80-83.9, **B-** = 77-79.9
C+ = 74-76.9, **C** = 70-73.9, **C-** = 67-69.9, **D+** = 64-66.9, **D** = 60-73.9
D- = 57.0-59.9, **E** = 0-59.9
- To round the percentage to 1 decimal place to calculate the letter grade, you may include the library header `<cmath>` and use:
`percentage = round(percentage * 10.0) / 10.0;`

M12C Class called Grade that inherit from class called Percentage that inherits from class Scores

Test cases

```
How many scores would you like to enter? 5
Enter 5 scores separated by a space: 97.6 87.5 65.5 99.5 55.1

Assignments Scores:
Assignment 1: 97.6
Assignment 2: 87.5
Assignment 3: 65.5
Assignment 4: 99.5
Assignment 5: 55.1

Total Percentage: 81.0%
Final Grade: B
```