# CMPE-50 Object-Oriented Concepts and Methodologies, Tarng, Spring 2021
## Homework #1

Due: 2/18/2021 before the class
The submission of the homework should be the cpp files with the output in the code comment. Each problem needs to have a complete program, meaning, it needs to contain the main() function and some test code and data. Therefore, you need to submit six cpp files in total. Do not zip the files. Name the files in the following way: CMPE50-HW-1-1.cpp, CMPE50-HW-1-2.cpp, etc.

Total 50 points

1. [15 pts] (Based on Programming Project 4.9)
   Write a program that asks for the user's height, weight, and age, and then computes clothing sizes according to the formulas:

   - Hat size = weight in pounds divided by height in inches and all that multiplied by 2.9.
   - Jacket size (chest in inches) = height times weight divided by 288 and then adjusted by adding 1/8 of an inch for each 10 years over age 30. (Note that the adjustment only takes place after a full 10 years. So, there is no adjustment for ages 30 through 39, but 1/8 of an inch is added for age 40.)
   - Waist in inches = weight divided by 5.7 and then adjusted by adding 1/10 of an inch for each 2 years over age 28. (Note that the adjustment only takes place after a full 2 years. So, there is no adjustment for age 29, but 1/10 of an inch is added for age 30.)

   Use functions for each calculation. Your program should allow the user to repeat this calculation as often as the user wishes.

2. [10 pts] (Based on Programming Project 4.15)
   You have invented a vending machine capable of deep frying twinkies. Write a program to simulate the vending machine. It costs $3.50 to buy a deep-fried twinkie, and the machine only takes coins in denominations of a dollar, quarter, dime, or nickel. Write code to simulate a person putting money into the vending machine by repeatedly prompting the user for the next coin to be inserted. Output the total entered so far when each coin is inserted. When $3.50 or more is added, the program should output "Enjoy your deep-fried twinkie" along with any change that should be returned. Use top-down design to determine appropriate functions for the program.

3. [5 pts] (Based on Programming Project 4.12)

Write an overloaded function `max` that takes either two or three parameters of type `double` and returns the largest of them.

4. [10 pts] (Based on Practice Program 5.1)

   Write a function that computes the average and standard deviation of four scores. The standard deviation is defined to be the square root of the average of the four values: $(s_i - a)^2$, where $a$ is average of the four scores $s_1$, $s_2$, $s_3$, and $s_4$. The function will have six parameters and will call two other functions. Embed the function in a driver program that allows you to test the function again and again until you tell the program you are finished.

   The six parameters are the four scores s1, s2, s3, s4, average and standard deviation. For average and standard deviation, the parameters should be pass-by-reference so that their values are returned to the calling function. The function prototype should look like the following:

   ```
   void avg_stdev(double s1, double s2, double s3, double s4,
           double &average, double &std_dev);
   ```

5. [5 pts] (Based on Practice Program 7.3)

   Write a function named `swapFrontBack` that takes as input an array of integers and an integer that specifies how many entries are in the array. The function should swap the first element in the array with the last element in the array. The function should check if the array is empty to prevent errors. Test your function with arrays of different length and with varying front and back numbers.

   The prototype of the function is given as follows:

   ```
   void swapFrontBack(int arr[], int size);
   ```

6. [5 pts]

   Write a function named countEvens that takes as input an array of positive integers and an integer that specifies how many entries are in the array. The function should return the count of even numbers in the array. Test your function with arrays of different length and with varying count of even numbers.