Problem Set – Pass by Reference Functions. Create an IPO for each problem within this document. Save the document and upload to Blackboard for grading. Then create a C++ program for each problem. Upload the .CPP files to Blackboard for grading.

1. Allow the user to enter a quantity and price, use ctl+z to stop. Use one function to compute the total (quantity times price), tax (7% of total) and total order (total plus tax). The function should be passed the quantity and price by value and total, tax and total order by reference. Display total, tax and total order in main. Sum and display total of all orders and tax for all orders and display after the loop (all data is processed).

input	process	output
Quantity and price.	Multiply quantity by price to get the total. Find tax as 7% of the total. Add tax to total to find the total order. Pass total, tax, and total order by reference to update them. Sum and display total, tax, and total order after all entries are done.	Each order's total, tax, total order, and final sums for all.

2. Enter the weight of a package and zip code. Use ctl+z to stop. Use a single function to do the computations specified next. Pass these weight and zip code by value, and pass postage, area charge and weight charge by reference. Compute postage to be sum of weight charge and area charge. Use tables below to find the charges. Compute weight charge to be weight x weight charge per ounce. Find the area charge in the table based on zip code. Then compute postage to be area charge plus weight charge. The function should return the weight charge, area charge and postage. Display area charge, weight charge and postage. Count and display the number of entries made.

   Area Table – Used to determine the area charge

   <u>Area</u> <u>Area Charge</u>

60171	$2.00
60172	$2.50
60635	$3.00
All others	$5.00

Weight Table – used to determine the weight charge

Weight	Weight Charge per Ounce
>100	0.02
>50	0.03
All other	0.05

input	process	output
Weight of the package and ZIP code.	Use the tables to find the correct area charge from the ZIP code and weight charge per ounce from the weight. Multiply weight by its rate to get the weight charge. Add the weight charge and area charge to get total postage. Pass postage, area charge, and weight charge by reference. Display all charges and count the number of packages entered.	Weight charge, area charge, total postage, and total number of entries.

3. Enter the student's last name, credit hours and financial aid, use ctl+z to stop. Pass credit hours and financial aid to a function by value. Pass tuition and tuition owed by reference. Compute tuition to be credit hours times $250. Compute tuition owed to be tuition minus the financial aid. Display student's last name, tuition and tuition owed. Sum and display total tuition owed by all students, count of number of entries and average amount owed by students.

input	process	output
Student's last name, credit hours, and financial aid.	Multiply credit hours by $250 to get tuition. Subtract financial aid from tuition. Pass tuition by reference to return the value. Keep count of students and sum total tuition. Compute average tuition at the end.	Student's last name, tuition owed, total tuition for all students, and average tuition.

4. Enter a number of widgets, use ctl+z to stop. Pass the number to a function by value, use ctl+z to stop. Use a single function to determine the cost per widget using the cost table below. Then compute extended price (number of widgets x cost per widget) and 7% sales tax. Finally compute total order to be extended price plus sales tax. Pass cost per widget, extended price, sales tax and total order by reference. For each line, display number of widgets, cost per widget, extended price, sales tax and total order. Sum all total orders and display when there is no more data to process.

Cost Table

Number of Widgets	Cost Per Widget
10000 and up	4.00
5,000 and up	5.00

All other amounts 10.00

input	process	output
Number of widgets ordered.	Find cost per widget using the cost table. Multiply number of widgets by cost per widget for extended price. Find tax as 7% of the extended price. Add tax and extended price for total order. Pass cost per widget, extended price, tax, and total order by reference. Display each and sum totals at the end.	Cost per widget, extended price, tax, total order, and total of all orders.

5. Enter the amount of investment, the 5 year interest rate and 10 year interest rate, use ctl+z to stop. Pass the amount and interest rates to a function by value. Pass variables representing five year amount and ten year amount to the same function by reference. Compute the five year amount and ten year amount using the formula below. Display the amount of the investment, the five year amount and the ten year amount.

   Five year amount = amount of the investment x (1 + 5 year rate) raised to 5^{th} power.

   Ten year amount = amount of the investment x (1+10 year rate) raised to 10^{th} power.

   Note: Enter the 5 year rate and 10 year rate in decimal form, i,e 5% is entered as 0.05.

   Also Note: you need to use the pow built in function. Recall the pow function syntax:

   pow (base, exponent)

   In this case, base is 1 + 5 year rate and exponent is 5.

Another line will be: base is 1 + 10 year rate and exponent is 10.

Use #include<math.h> for the pow function.

input	process	output
Investment amount, 5-year interest rate, and 10-year interest rate.	Use the pow function to calculate the 5-year and 10-year amounts. Five-year amount = investment × (1 + 5-year rate)^5. Ten-year amount = investment × (1 + 10-year rate)^{10}. Pass both values by reference.	Investment, five-year amount, and ten-year amount.