Problem Set – More on Pass By Value Functions. Create an IPO for each of the problems below. Save the document with the IPO’s and then upload to Blackboard. Next write code for the problems. Then upload the .cpp files to Blackboard.

1. Allow the user to enter a quantity and price, use ctl+z to stop. Use a function to compute the total (quantity times price). The function should be passed the quantity and price and then return the total. Use another function to compute 10% discount if the total is over $10,0000.00 and 5% for any amount equal to or lower than $10,000.00. The second function should receive the total, check which discount rate to charge, compute discount amount and then compute the discount total (total – discount amount). It should return the discount total. Display total and discount total. Sum total and discount total and display at the end.

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| Quantity (int)  - Price (float)<  - Use Ctrl+Z to stop input | - Use a function calculateTotal(quantity, price) to compute total  - Use another function applyDiscount(total) to  • Check if total > 10000.00 → apply 10% discount  • Else → apply 5% discount  • Compute discount amount  • Compute discounted total = total - discount amount  Accumulate total and discounted total | - Display total and discounted total for each entry  - Display final sum of all totals and all discounted totals |

1. Enter players last name, number of hits and at bats at the keyboard, use ctl+z to stop. Use a function to compute batting average. Pass the hits and at bats to the function. The function should return batting average (at bats / number of hits). Display last name and batting average. Give a count of the number of players entered and display the count after the loop.

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| - Player's last name (string)  - Number of hits (int)  Number of at bats (int)  Use Ctrl+Z to stop input | Use a function calculateBattingAverage(hits, atBats) to compute batting average  • Formula: hits / atBats  Count number of players entered | - Display player's last name and batting average  Display total number of players entered |

1. Enter the destination city, miles travelled and gallons used for a trip, use ctl+z to stop. Use a function to compute miles per gallon. Pass miles travelled and gallons used to the function. The function should return miles per gallon. Use another function to compute gas cost. Pass to this function gallons used. Each gallon costs $3.50. Compute and return the cost. Display destination city, miles per gallon and cost of gas. Sum and display the total cost of gas.

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| - Destination city (string)  Miles travelled (double) Gallons used (double) | Use function calculateMPG(miles, gallons) to compute miles per gallon  • Formula: miles / gallons  Use function calculateCost(gallons) to compute gas cost  • Formula: gallons × $3.50  Accumulate total gas cost | Display destination city, miles per gallon, and cost of gas  - Display total cost of gas after loop ends |

1. Allow the employee to enter last name, job code and hours worked, use ctl+z to stop. Use a function to determine the pay rate. Pass to this function the job code and it should return rate of pay. Use the following rates based on Job code: L is $25/hr, A is $30/hr and J is $50/hr for respective pay rates. Write another function to determine the gross pay. Pass to this function the hours worked and pay rate and return gross pay. Give time and a half for overtime. Display last name and gross pay. Sum and display total of all gross pay.

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| Employee last name (string) Job code (char: 'L', 'A', or 'J') Hours worked (double) | Use function getPayRate(jobCode) to return hourly rate:  • 'L' → $25/hr  • 'A' → $30/hr  'J' → $50/hr  - Use function calculateGrossPay(hours, rate):<br>  • If hours ≤ 40 → gross = hours × rate  • If hours > 40 → gross = (40 × rate) + ((hours - 40) × rate × 1.5)  - Accumulate total gross pay | Display last name and gross pay  - Display total gross pay after loop ends |

1. Allow the user to enter student last name, credit hours and district code, use ctl+z to stop. Use a function to compute tuition owed. First, write a function to determine the cost per credit hour. Charge In district (code of I) $250 per credit hour. Out of district (code of O) is $550 per credit hour. Write another function to compute tuition cost. Display student name and tuition cost. Sum and display total of all tuition costs.

|  |  |  |
| --- | --- | --- |
| INPUT | PROCESS | OUTPUT |
| Student last name (string)  - Credit hours (int)  - District code (char: 'I' or 'O') | Use function getRatePerCredit(districtCode)  'I' → $250/credit  • 'O' → $550/credit  - Use function calculateTuition(creditHours, rate)  • tuition = creditHours × rate | Display student last name and tuition cost  - Display total tuition cost after loop ends |