Problem Set – While Loops. Develop IPO for each of the problems below and then save within this document. Then Write the code files using C++. Upload both IPO and .CPP files to Blackboard.

1. Allow any number of users to enter a quantity and price at the keyboard (use ctl+z to stop). Compute the extended price (quantity times price). If quantity is over 1000 give a 10% discount. Display quantity, price and extended, discount amount and discounted price for each entry. Keep a sum of the total for all the discounted prices. Display the total of discounted prices after all entries have been entered.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Qty, price | Enter values for quantity and price | In loop: quantity: QUANTITY price: PRICE extended price: EXTENDED PRICE discounted amount: DISCOUNT AMOUNT discounted price: DISCOUNT PRICE “Enter quantity and price: " "or use (Ctrl + Z) to end" |
| Extended price, total | Start of loop >> extended price = quantity \* price Create if: if quantity > 1000, discount 10% discount amount = extended price \*.10 Discount price = extended price – discount amount  Else: continue | End of loop: Total discounted prices: TOTAL DISCOUNT |
| Discount price, discount amount | Display quantity, price per item, extended price, discount amount, discount price Allow user to continue loop, or end loop |  |
| Total discount | If loop ends: display total of discounted prices |  |
|  |  |  |

1. Allow any number of players to enter last name, number of hits and at bats at the keyboard (use ctl+z to stop). Compute batting average (hits/ at bats). Display last name and batting average for each player. Keep a count of the number of players (or entries) made. Display the count after all entries have been made.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Last name | Allow user’s to add values to last name, number of hit, and at bats | During loop- Player: LAST Batting Average: AVERAGE |
| # of hits | Create loop  Starting point (when entering values for last name, number of hits, and at bats) | Finished loop: Entries entered: # OF PLAYERS |
| At bats | Make equation for average (hits/ at bats) Display Last name and Average |  |
| Average | Add +1 to count of entries |  |
| # of players | When done with loop - Display total count of entries |  |

1. Enter destination city, miles travelled to get there and gallons of gasoline used for any numberof trips entered at the keyboard (use ctl+z to stop). Compute miles per gallon (miles travelled / gallons used). Display the destination city and miles per gallon for each trip entered. Sum the miles travelled and give a count of the number of trips made. Display these at the end of the program.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Destination | Allow user to enter values for destination, miles and gallons used | During loop: Destination: DESTINATION Miles per gallons: MPG Enter location, miles traveled, and gallons used. Or end loop |
| Trips | Start loop once values are entered Create equations for mpg, total miles, trips | End loop: Total miles travel: TOTAL MILES Total trips: TRIPS |
| Miles, mpg, gallons | Mpg = MILEs / GALLONS Total miles = TOTAL MILES + MILES Trips = TRIPS + 1 |  |
| Total miles | Display DESTINATION + MPG Allow user to start loop again, or end loop |  |
|  | If loop ended: Display total miles and trips made |  |

1. Allow the employee to enter last name, job code and hours worked (use ctl+z to stop). Calculate pay. (Job code L is $25/hr, A is $30/hr and J is $50/hr). Give time and a half for overtime. Display last name, job code, hours worked and pay for each employee. Sum the pay for each employee as well as count the entries made. After all entries are made, compute and display the average pay and the number of entries made.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Last name, code | Allow user to enter values for last name, job code, and hours |  |
| Hours, rate, pay, pay sum | Start loop when values are added: IF statement: if code = L, rate = 25 if code = A, rate = 30 if code = J, rate = 50 | During loop >> last name: LAST Job code: CODE Rate per hour: RATE Total pay: pay "Enter (1) last name, (2) job code (L, A, or J), and (3) hours worked (Ctrl+Z to end):” |
| Average, entries | Create anther if statement for OT occurrence If hours > 40 ot time = hours – 40 ot rate = rate \* 1.5 ot pay = ot time \* ot rate non ot time = hours – ot time non ot pay = non ot time \* rate pay = ot pay + non ot pay  Else (40 or less hours) pay = hours \* rate | After loop >> Average pay: AVERAGE Entries: ENTRIES |
|  | Add sums together: pay sum = pay sum + pay Keep track of entries: entries = entries + 1 |  |
| Ot time, ot pay, ot rate | Display last name, code, hours, rate, and pay Allow user to end loop or continue it |  |
| Non ot pay, non ot time | If user ends loop: Equation for average: average = pay sum / entries Display average and total entries |  |

1. Allow the user to enter student last name, credit hours and district code for any number of students (use ctl+z to stop). Compute tuition owed. In district (code of I) is charged $250 per credit hour. Out of district (code of O) is $550 per credit hour. Display student name and tuition owed for each entry. Sum the amount of tuition owed for all students as well and the total credit hours taken and finally the number of students who entered data. Display total tuition, total credit hours taken and count of number of students at the end.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Last name code | Allow user to enter values for last name, code, and credits Start loop upon value input | During loop >> Name: LAST NAME Student tuition total: TUITION “Enter Last name, credit hours, and district code ('I' for In district. 'O' for out of district): "  "(Ctrl + Z when done): " |
| Credits, students | Start of Loop: if statement: if code = I, credit per hour = 250 If code = O, credit per hour = 550 | After loop >> Total tuition: TOTAL TUITION total credits: TOTAL CREDITS total students: STUDENTS |
| Credit per hour, tuition | Equations: tuition (per student) = credits \* rate per hour  Tuition total (all students) = tuition total + tuition (per student) total credits (all students) = total credits + credits (per student) students = students + 1 |  |
| Tuition total, total credits | Display last name, tuition (per student)  Allow user to enter another student, or end loop |  |
|  | If end loop: Display total tuition (all students), credits (all students), and students registered |  |