Program Assignments – While Loops. Develop an IPO for each of the problems below. Place the IPO into your repository. Then write code for each problem and place those files (py) into your repository. Paste the link to your repository into the Assignment Completion Link on Blackboard.

1. Display the odd numbers starting at 1 and ending with 25. Use a while loop structure for this problem.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| Nothing | * For 1 through 26 print number * Add 1 to number | * The numbers 1 -25 |

1. Allow the user to enter a start value, stop value and increment value from the keyboard. Display all the numbers from the start value to stop value using the increment value as you proceed. Use a while loop structure for this problem.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * start value * stop value * increment | * while start values does not equal end value * print value and add increment | * start value * integers * end value |

1. Prompt the user on whether they want to do this program **(just before the while loop**). “Yes” entry means they want to continue. Any other response indicates they will stop the program. This response is the loop control variable. If the user answers “Yes “then go into the while loop.

Once in the while loop. You are to prompt the user for their last name and two exam scores. Compute the average exam score. Display last name and average. After the loop, display a count of the number of students who entered data.

Finally, the **last statements** **within the while loop** will ask the user if they want to do this loop again. In other words the user needs to be prompted again. The reason is that the end of the loop takes execution to the while condition to be evaluated again. It can not take us to the first few lines of code that prompt the user for the first time. That code is out of the loop. Therefore, we need a second prompt at the bottom, inside the loop.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * no | * while answer does not equal yes ask prompt again | * thanks for using this program |
| * yes * name * exam score 1 * exam score 2 | * average exam score * total users + 1 * prompt if they want to use the program agaim | * last name * average score * total users * prompt |

1. Prompt the user on whether they want to do this program **(just before the while loop**). Yes means they want to continue. Any other response indicates they will stop the program. This response is the loop control variable. If the user answers Yes then go into the while loop.

Once in the while loop. You are to prompt the user for employee last name, hours worked and rate of pay. Compute gross pay. Give the employee time and a half for hours worked over 40. Sum the gross pay and count the number of employees.

For each employee display their last name and gross pay.

After the loop (all data entered) display the sum of all the gross pays, and count of the number of employees. Compute and display the average pay.

Finally, the **last statements** **within the while loop** will ask the user if they want to do this loop again. In other words the user needs to be prompted again. The reason is that the end of the loop takes execution to the while condition to be evaluated again. It can not take us to the first few lines of code that prompt the user for the first time. That code is out of the loop. Therefore, we need a second prompt at the bottom, inside the loop.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * no | * while answer does not equal yes end program | * thanks for using this program |
| * yes * name * hours * rate of pay | * overtime = hours – 40 * if overtime > 0 multiply overtime by rate \*1.5, multiple 40\* rate, add the two values * else hours \* rate of pay * total employees = + 1 * prompt to use again? | * name * gross pay * total number of employees |

1. Prompt the user on whether they want to do this program **(just before the while loop**). Yes means they want to continue. Any other response indicates they will stop the program. This response is the loop control variable. If the user answers Yes then go into the while loop.

Once in the while loop. You are to prompt the user for quantity and price of an item. Compute extended price (quantity times price of an item. If the extended price is greater than 10000.00 compute a discount of 25%. All other orders get a 10% discount. For each order display extended price, discount amount (extended price x discount percent), total (extended price – discount amount).

For each order sum the discount amount.

After the loop (all data entered) display the sum of all the discounts.

Finally, the **last statements** **within the while loop** will ask the user if they want to do this loop again. In other words the user needs to be prompted again. The reason is that the end of the loop takes execution to the while condition to be evaluated again. It can not take us to the first few lines of code that prompt the user for the first time. That code is out of the loop. Therefore, we need a second prompt at the bottom, inside the loop.

|  |  |  |
| --- | --- | --- |
| Input | Process | Output |
| * no | * while answer does not equal yes end program | * thanks for using this program |
| * yes * quantity * price per unit | * quantity \* price per unit * if price per unit > 1000 multiply by .25 for discount * else discount is .1 * subtract discount from total price * add discount price to total discount | * price * discount percent * discounted price * sum of discounts |