

Session 6 Assignment Problems – Loop Logic 1

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

Input	Process	Output
None, counter starts at 1	Initialize counter= 1	Odd numbers 1,3,5...25
	counter<=25 Display counter Add 2 to counter	

2. 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	Set current value= startvalue	The sequence of numbers based on user input
Stop value	While current value<= stop value	
Increment	Display current value	
	Current value= current value+ increment	

3. 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
Do program (yes/no)	Initialize student count=0 While doprogram is “yes”	Individual lastname
Last name	Compute average=(score 1 + score 2) /2	Individual average
Score 1	Display last name and average	Total student count
Score 2	Increment student count by 1	
	Prompt doprogram(“yes” to continue) After loop: display student count	

- 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
-------	---------	--------

doloop(yes/no) Lastname hours worked payrate	Initilize total gross pay= 0, empcount= 0	Individual last name and gross
	While doloop = "yes", if hours >40 gross = $(40 \times \text{rate}) + (\text{hours} - 40) \times \text{rate} \times 1.5$	Summary: total gross pay, empcount, avg pay
	Else: gross= hours x rate Total gross pay += gross emp count +=1 Display last name, gross Re reprompt doloop	
	After loop:avg=total gross pay/ emp count.	

5. 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
Do program(yes/no)	Initialize total discounts=0	Extended price
quantity	While do program is "yes"	Discount amount
Unit price	-extended price = quantity x	Total price

	unitprice	
	If extended price > 10000 then discount rate = .25, else discount rate = .10	Sum of all discount amount
	discountamount = extended price * discount rate	
	Total price = extended price - discount amount	
	Display extended price, discount amount, total price	
	Prompt doprogram('yes' to continue.)	
	After loop: display total discounts	

01/07/2026