**IPO chart:**

|  |  |  |
| --- | --- | --- |
| Input: | Processing: | Output: |
| User input for student's last name, credit hours, and district code.  User input ends when Ctrl+Z (Windows) is pressed. | Define a function calculateTuitionOwed that calculates tuition owed based on credit hours and district code ('I' or 'O'). If the district code is not recognized, it returns a tuition owed of 0.0.  Initialize a variable to keep track of the total tuition owed (totalTuitionOwed).  Use a while loop to repeatedly ask the user for student information and perform the following steps for each student:  Read and store the student's last name, credit hours, and district code.  Calculate the tuition owed using the calculateTuitionOwed function.  Display the last name and tuition owed for each student.  Update the total tuition owed.  End input when the user presses Ctrl+Z . | For each student, display the last name and tuition owed.  After processing all students, display the total tuition owed. |

**Code:**

def calculateTuitionOwed(creditHours, districtCode):

if districtCode == 'I':

return creditHours \* 250.0

elif districtCode == 'O':

return creditHours \* 550.0

else:

return 0.0

totalTuitionOwed = 0

try:

while True:

lastName = input("Enter student's last name (Ctrl+Z or Ctrl+D to stop): ")

creditHours = float(input("Enter credit hours: "))

districtCode = input("Enter district code (I for In district, O for Out of district): ").upper()

tuitionOwed = calculateTuitionOwed(creditHours, districtCode)

print(f"Last Name: {lastName}")

print(f"Tuition Owed: ${tuitionOwed:.2f}\n")

totalTuitionOwed += tuitionOwed

except EOFError:

pass

print(f"Total Tuition Owed: ${totalTuitionOwed:.2f}")