Obstacles overcome:

Determining how to detect any errors in the information that was input, however this was solved by simply checking each input for errors with an if statement checking if either the string was empty or if the value was negative and outputting an error.

Ending the program if any error was detected: this was done by creating a return statement which would end the main() function.

Creating a reduced tax rate for teachers and nurses, this was done by checking the occupation string and changing a tax rate variable for the bracket between $55,000 and $125,000.

Checking the amount of money in each bracket, this was done by checking the amount of total taxable income, and if it exceeded a certain bracket this meant that any brackets below it had the full amount of taxable income. Then, the amount of money in the highest tax bracket each person was being taxed at was determined.

Had certain values as integers instead of doubles such as taxable income, which should be a double to track the number of cents on the dollar.

|  |  |
| --- | --- |
| Name, Taxable Income, Occupation, # Children | Case Tested & Output |
| Kunal, 150000, sales, 0 | Upper bracket |
| Kunal, 125000, sales, 0 | Equal to 2nd bracket edge |
| Kunal, 55000, sales, 0 | 1st bracket edge |
| Kunal, 150000, sales, 5 | Child deductions fail above 125000 |
| Kunal, 0, student, 5 | Child deductions will not go negative |
| , 150000, sales, 0 | Name empty error |
| Kunal, -150000, sales, 0 | Negative income error |
| Kunal, 150000, , 0 | Occupation empty error |
| Kunal, 150000, sales, -5 | Negative child error |
| Kunal, 150000, teacher, 0 | Teacher reduced tax rate |
| Kunal, 150000, nurse, 0 | Nurse reduced tax rate |
| Kunal, 55000, teacher, 0 | Teacher reduced tax rate underneath reduced tax threshold |
| Kunal, 100000, sales, 0 | Between brackets |
| Kunal, 40000, sales, 0 | Within lowest bracket |
| Kunal 100000, sales, 5 | Child deductions |
|  |  |