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 | function:compute\_prices  input:qty, price  references:&extprice, &discount, &discountprice  extprice=qty\*price  if(qty>1000)  discount=extprice\*.1  discountprice=extprice-discount  else  discount=0  discountprice=extprice | inside loop  qty  price  extprice  discount  discountprice |
| price | sumdiscountprice=0 | outside loop  sumdiscountprice |
|  | while not eof |  |
|  | compute\_prices(qty, price, discount, discountprice) |  |
|  | sumdiscountprice+=discountprice |  |
|  | display: qty, price, extprice, discount, sumdiscountprice  ask prompt again |  |
|  | end of loop |  |
|  |  |  |

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 |
| lname | function:compute\_bataverage  input:nhits, nbats  references:bataverage  bataverage=nhits/nbats | inside loop  lname  bataverage |
| nhits | ncount=0 | outside loop  ncount |
| nbats | while not eof |  |
|  | compute\_bataverage(nhits, nbats, bataverage) |  |
|  | ncount+=1 |  |
|  | display:lname, bataverage  ask prompt again |  |
|  | end of loop |  |

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 |
| city | function:compute\_mpg  input:miles, gallons  references:mpg  mpg=miles/gallons | Inside loop  city  mpg |
| miles | summiles=0  ncount=0 | outside loop  summiles  ncount |
| gallons | while not eof |  |
|  | compute\_mpg(miles, gallons, mpg) |  |
|  | summiles+=miles  ncount+=1 |  |
|  | display: city, mpg  ask prompt again |  |
|  | end of loop |  |

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 |
| lname | function:compute\_pay  input:code, hours  reference:pay  switch(code)  case L:  payrate=25  case A:  payrate=30  case J:  payrate=50  default  incorrect code  if(hours>40)  overtime=hours-40  pay=(payrate\*40)+(payrate\*1.5\*overtime)  else  pay=payrate\*hours | inside loop  lname  code  hours  pay |
| code | function:compute\_average  input:sumpay, ncount  reference:averagepay  averagepay=sumpay/ncount | outside loop  averagepay  ncount |
| hours | sumpay=0  ncount=0 |  |
|  | while not eof |  |
|  | compute\_pay(code, hours, pay) |  |
|  | sumpay+=pay  ncount+=1 |  |
|  | Display: lname, code, hours, pay  ask prompt again |  |
|  | end of loop |  |
|  | compute\_average(sumpay, ncount, averagepay) |  |

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 |
| lname | function:compute\_tuition  input:credithours, code  reference:tuition  switch(code)  case I:  creditrate=250  case O:  creditrate=550  default  incorrect code  tuition=credithours\*creditrate | inside loop  lname  tuition |
| credithours | sumtuition=0  sumcredithours=0  ncount=0 | outside loop  sumtuition  sumcredithours  ncount |
| code | while not eof |  |
|  | compute\_tuition(credithours, code, tuition) |  |
|  | sumtuiton+=tuition  sumcredithours+=credithours  ncount+=1 |  |
|  | display:lname, tuition  ask prompt again |  |
|  | end of loop |  |