Best practice requirements for source code

1. Properly indented / spaced
2. Appropriate comments (including your name and assignment identification at the beginning)
3. Appropriate & meaning variable names (avoid single letter variable unless used for short-lived iteration)

Do NOT use any C++ syntax.

|  |
| --- |
| Homework : \_\_\_1.A\_\_\_\_\_\_  1.A Source code  //Christopher Badolato CH432391  //Assignment 1A  //8/27/2018  //This program will print the salary of a  //chemical company pay based on the total sales of the employee  #include <stdio.h>  #include <stdlib.h>  int main() {  //Initialize total sales  float totalSales = 0, salary = 0, quit = -1;  //while totalSales is not entered by the user as -1  //we will continue taking sale values for calculation  while(totalSales != quit){  printf("Enter sales in dollar (-1 to end): ");  scanf("%f", &totalSales);  //if our value entered DOES NOT equal -1 preform salary calculations.  if(totalSales != quit){  salary = (totalSales \* .09) + 200;  printf("Salary is: $%.2f\n\n", salary);  }  //otherwise we want to quit the loop.  else{  break;  }  }  return 0;  } |
|  |
| 1.A Program output  Enter sales in dollar (-1 to end): 5000.00  Salary is: $650.00  Enter sales in dollar (-1 to end): 1234.56  Salary is: $311.11  Enter sales in dollar (-1 to end): -1  Process returned 0 (0x0) execution time : 9.861 s  Press any key to continue. |
| Homework : \_\_1.B\_\_\_\_\_\_\_  1.B Source Code  //Christopher Badolato CH432391  //Assignment 1B  //8/27/2018  //This program will print the interest charged of a loan  //by calculating simple interest.  #include <stdio.h>  #include <stdlib.h>  int main() {  //Initalize all values as floats for simple calculation of interest.  int quit = -1;  float loanPrincipal = 0, interestRate = 0;  float interestCharge = 0, termOfLoan = 0;  //loop until our loan value is -1 or is somehow initialized as -1.  while(loanPrincipal != quit){  printf("Enter loan principal (-1 to end): ");  scanf("%f", &loanPrincipal);  //if our loan principal DOES NOT equal -1, get the rest of our values  // then we perform the calculation  if(loanPrincipal != quit){  printf("Enter interest rate: ");  scanf("%f", &interestRate);  printf("Enter term of loan in days: ");  scanf("%f", &termOfLoan);  interestCharge = loanPrincipal \* interestRate \* (termOfLoan/365);  printf("The interest charge is $%.2f\n\n", interestCharge);  //if our value for loan principal is -1 break the loop  //and exit the program.  }  else{  break;  }  }  return 0;  } |
| 1.B Program output  Enter loan principal (-1 to end): 1000.00  Enter interest rate: .1  Enter term of loan in days: 365  The interest charge is $100.00  Enter loan principal (-1 to end): 1000.00  Enter interest rate: .08375  Enter term of loan in days: 224  The interest charge is $51.40  Enter loan principal (-1 to end): -1  Process returned 0 (0x0) execution time : 32.756 s  Press any key to continue. |
|  |
|  |
| Homework : \_\_1.C\_\_\_\_\_\_  1.C Source Code  //Christopher Badolato CH432391  //Assignment 1C  //8/27/2018  //This program will create a hollow square of the length of the user input  //using for loops to fill out the square and place asterisks where needed.  #include <stdio.h>  #include <stdlib.h>  int main() {  //initialize our row and column values to 0.  int squareSide, row = 0, column = 0;  //Get square size from users.  printf("Enter the size of the hollow square: ");  scanf("%d", &squareSide);  //First we need to draw the top  //add one each iteration to go through each row.  while(row < squareSide){  printf("\*");  row++;  }  printf("\n");  row = 2;  /\*since we have already printed the top we will need to skip to our  second row and start printing there.  while our row value is less than the square size AND  while our columns value is less than our square size we want to print \*  ONLY on our first and last columns.  \*/  while(row < squareSide){  while(column < squareSide){  //makes sure we only print on the first and last columns.  if(column == squareSide -1 || column == 0){  printf("\*");  }  //otherwise print a space.  else{  printf(" ");  }  //go to the next column.  column++;  }  //reset the column value each row we draw and also print a new line.  column = 0;  row++;  printf("\n");  }  //reset our row value and draw the bottom of the square to complete the box.  row = 0;  while(row < squareSide){  printf("\*");  row++;  }  return 0;  } |
| 1.C Program output  As you can see from the pictures my output is correct it was just not correctly copying over  and just adding spaces was suspicious and looked weird.  Enter the size of the hollow square: 4  \*\*\*\*  \* \*  \* \*  \*\*\*\*  Process returned 0 (0x0) execution time : 3.240 s  Press any key to continue.  Enter the size of the hollow square: 10  \*\*\*\*\*\*\*\*\*\*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \* \*  \*\*\*\*\*\*\*\*\*\*  Process returned 0 (0x0) execution time : 6.682 s  Press any key to continue. |
|  |