



DEPARTMENT OF  
**SOFTWARE TECHNOLOGY**

## Hands-On Exam 1: Cost Splitting

Whenever a friend or loved one has a celebration coming up, some people would suggest 1 big gift whose price is split among those who agree to the idea. Sometimes the computation can get messy so a program to help them know the exact amount would be helpful. For this problem, a programmer has started developing the program but has yet to finish it. Your task is to complete the missing parts to complete the program.

Assumptions:

- Assume all inputs are valid (ie. no negative inputs)
- The maximum amount for the gift cannot exceed 9999.00
- All displayed monetary values must be limited to 2 decimal places

The user will be asked to input the following information:

- The number of people in the group
- The price of the gift
- The number of people who backed out

The program will take the inputs and display the amount each person will contribute for the gift.

Sample runs:

```
Enter no. of people in the group: 5
Enter total cost of the present: 2000
Number of people backing out: 3

Original contribution per person:      400.00
Additional contribution per person:    600.00
New contribution cost per person:      1000.00
```

```

Enter no. of people in the group: 10
Enter total cost of the present: 5500
Number of people backing out: 4

Original contribution per person:      550.00
Additional contribution per person:    366.67
New contribution cost per person:      916.67

Enter no. of people in the group: 7
Enter total cost of the present: 5300
Number of people backing out: 2

Original contribution per person:      757.14
Additional contribution per person:    302.86
New contribution cost per person:      1060.00

Enter no. of people in the group: 4
Enter total cost of the present: 3300
Number of people backing out: 0

Original contribution per person:      825.00
Additional contribution per person:     0.00
New contribution cost per person:      825.00

```

Task 1: Implement computeNewContribution() function (8 pts)

This function will compute the new cost per person and return the value.

Task 2: Implement displayBreakdown() function (10 pts)

This function receives 3 values as parameter and will display the information(the last 3 lines) seen in the above sample run. It is required that the spacing, alignment, and text are the same as that in the sample runs.

Task 3: Complete the main() function (32 pts)

The main function is where the variables are declared and where the inputs will be done. It will also call the functions.

There are 2 files provided for this problem, both of which you need to complete and submit:

- LASTNAME-CostMain.c - contains the main() function
- LASTNAME-Cost.c - pre-defined and user-defined functions for the program

**DELIVERABLES:**

- (1) Your C program source file **LASTNAME-Cost.c** with your own last name as filename. For example, if your lastname is SANTOS, then the source file should be named as SANTOS-Cost.c. Upload your source file in AnimoSpace before the indicated submission time.
- (2) Your C program source file **LASTNAME-CostMain.c** with your own last name as filename. For example, if your lastname is SANTOS, then the source file should be named as SANTOS-CostMain.c. Upload your source file in AnimoSpace before the indicated submission time

**TESTING & SCORING:**

- Your program will be compiled via gcc -Wall. Thus, for each function that does not compile successfully, the score for that function is 0.
- Your program will be tested by your instructor with other main() (which may contain different values from the ones given to you) and with function calls of different parameter values.
- Full credit will be given for the function only if the student's implementation are all correct for all the test values used by the instructor during checking AND only if the student's implementation complied with the requirement and did not violate restrictions. Deductions will be given if not all test cases produce correct results. No credit is given if restrictions were not followed.