

# Assignment 4

CS 135: Computer Science I  
Spring 2008

## Objectives

1. Learn and demonstrate use of `if-then-else` statements in C/C++
2. Learn and demonstrate use of relational operators in C/C++
3. Learn and demonstrate prototyping, defining, and calling simple value returning functions.

## 1 Part I: Bending Moment 2 (10 points)

Design, code, and test a C/C++ program to compute and display the maximum bending moment,  $m$ , of a beam, which is supported on both ends as shown in Figure 1.

The formula for maximum bending moment is:

$$m = xw \frac{(l - x)}{l}$$

where  $x$  is the distance of the weight,  $w$ , from the end of the beam.  $l$  is the length of the beam. Your program should read the values of  $l$ ,  $x$ , and  $w$ , from standard input and write the maximum bending moment of the beam to standard output.

The major difference from assignment two is that you have to ensure that the inputs are valid. Handle cases when the user inputs incorrect values for  $l$ ,  $x$ , and  $w$ . When incorrect values are entered the program should print an appropriate message to the standard output and exit.

For this section of the assignment, please create and run at least **five** test cases and make sure that your program handles these situations. These test cases are the four cases below

1. Incorrect value of  $w$
2. Incorrect value of  $l$

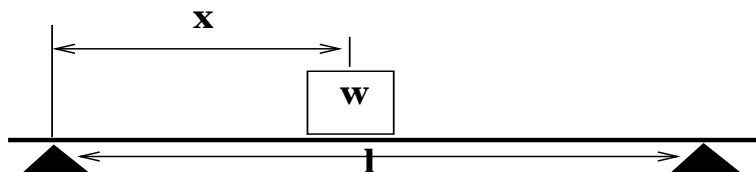


Figure 1: Beam supported on both ends with a weight placed on the beam.

3. Incorrect value of  $x$  that is greater than  $l$
4. Correct values for  $x$ ,  $w$ , and  $l$

and one other incorrect input case that you will have to come up with on your own.

Like in assignment 2, your program should print the maximum bending moment for a weight that is half the distance as the the weight above ( $0.5 * x$ ) Your program must **prototype, use, and define** a function to compute the maximum bending moment according to the formula above. You need not worry about lining up output for this part. Sample Testing (the executable's name is `bender`).

```
sushil@flash ~/classes/135/assignments/as4/code > ./bender
Enter x (distance of w), w (weight), and l (beam length)
10 20 1
x = 10, w = 20, l = 1
Length l : 1 cannot be less than x: 10
Error in input, Exiting...
sushil@flash ~/classes/135/assignments/as4/code > ./bender
Enter x (distance of w), w (weight), and l (beam length)
10 20 10
x = 10, w = 20, l = 10
The maximum bending moment for distance 10 is: 0
The maximum bending moment for distance 5 is: 50
sushil@flash ~/classes/135/assignments/as4/code > ./bender
Enter x (distance of w), w (weight), and l (beam length)
10 20 -100
x = 10, w = 20, l = -100
Length l: -100 cannot be negative
Length l : -100 cannot be less than x: 10
Error in input, Exiting...
```

## 2 Scoring in Pong (10 points)

Design, code, and test a C/C++ program that computes scores for a computer game, such as pong. Your program should generate the number of blocks (`blocks`) and the number of misses (`misses`) using a random number generator. Both `blocks` and `misses` are in the range 0 through 100. Your program should generate and compute the score for two sets of `blocks` and `misses`.

We compute scores as follows:

$$\text{partialScore} = (\text{blocks}^2 - \text{misses}) \times 10$$

If the number of misses is greater than or equal to the number of blocks, then the bonus is 1000 otherwise the bonus is 100. The complete score is the sum of the `partialScore` and `bonus`

$$\text{score} = \text{partialScore} + \text{bonus}$$

Your program should write the number of `blocks`, `misses`, `partialScore`, `bonus`, and `score` to standard output. Your output should look like it does below. Sample Testing (the executable's name is `pong`).

```

sushil@flash ~/classes/135/assignments/as4/code > ./pong
  Blocks  Misses  Partial  Bonus  Score
    78     50 60340.00 100.00 60440.00
    65     51 41740.00 100.00 41840.00
sushil@flash ~/classes/135/assignments/as4/code > ./pong
  Blocks  Misses  Partial  Bonus  Score
    52     64 26400.00 1000.00 27400.00
    31     87  8740.00 1000.00  9740.00
sushil@flash ~/classes/135/assignments/as4/code > ./pong
  Blocks  Misses  Partial  Bonus  Score
    46     69 20470.00 1000.00 21470.00
    72     49 51350.00 100.00 51450.00

```

### 3 Handing it in

Turn in a Folder (Binder) containing:

1. Cover sheet with
  - (a) Assignment Number
  - (b) Section Number
  - (c) Your name
  - (d) Your email
  - (e) Your TA's name
2. Source code (.cpp) file (s) on a CD/USB stick with your name and section number written on your CD/USB stick.
3. **Hardcopy of your source code (printout)**
4. A printout of sample runs of your program on (at least) the test cases listed above.

Ask an instructor or TA if you have questions.