

Assignment #6

Boolean Expressions & Functions

Military Academy

Goal

For this assignment you will gain experience writing compounded Boolean expressions, using functions, Boolean variables, repetition, & selection statements.

*** NOTE: THIS IS A DOUBLE ASSIGNMENT – IT COUNTS AS 2 ASSIGNMENTS TOWARD YOUR FINAL GRADE.**

REMINDER: All assignments are to be completed individually. You may not get help from anyone other than the class tutors, LRC tutors or Dr. Rousseau.

Instructions

Design, implement in C++ & test the problem outlined in the problem statement. Submit the solution into Canvas.

Problem Statement

A military academy accepts candidates according to the following height and weight requirements based upon their sex.

<u>Sex</u>	<u>Min. Height</u>	<u>Max. Height</u>	<u>Min.Weight</u>	<u>Max. Weight</u>
Male	65 in.	80 in.	130 lbs.	250 lbs.
Female	62 in.	75 in.	110 lbs.	185 lbs.

Design and implement a program in C++ that informs the user of the acceptance status of each candidate applying based on the qualifications stated above. The user will input a group of candidates at a time. For each candidate they will provide a sex, height & weight (**The gender should be read into a single character variable**). Each of the input values should be error checked. (See below for the minimum and maximum constraints). For each candidate, program will output whether each candidate was accepted, rejected based on height, weight, or height & weight. The user will input 'X' or 'x' for the gender when they are finished entering in all the applicants. The program will output the number of candidates accepted and the percentage of candidates accepted. Run the code 3 times to test it (**do not use a for loop for this**).

Requirements

You will be evaluated on your appropriate use of loops, Boolean expressions, the Boolean data type, functions, selection and repetition statements.

- You must first write out a **design** for the code. It does not need to be created with a tool, but it needs to be readable. You can use pseudocode or a flowchart. Indicate where you will be including functions and design the functions.
- The user will input the sex a single character, height and weight will be input as whole numbers.
- Error check** the sex, height and weight according to the following valid values and ranges:
 Sex: 'm', 'f', 'M', 'F'
 Height: Between 24" and 110" inclusively (including 24 and 110).
 Weight: Between 50 lbs. and 1400 lbs. inclusively.
 * The error messages are specified in the Expected Input / Output section.

- You **cannot use toupper** or **tolower** (or write the code to do the same as the pre-defined function). Lower & upper case values for sex must be accomplished via compounding Boolean Expressions by using **&&** or **||** operators).
- The candidate's height requirements should be checked. The result should be stored in a Boolean variable named **acceptableHeight**. The variable will be true if the candidate meets the requirements and false if they don't.
→ this must be accomplished using a single assignment statement that assigns a Boolean expression. This will be a long compounded expression and you may **not** break it into multiple expressions or use any form of selection statement to accomplish this task. In other words, **NO** selection statements to assign based on male or female candidates. You will have once in your code:
acceptableHeight = boolean expression;
- The candidate's weight requirements should be checked. The result should be stored in a Boolean variable named **acceptableWeight**.
→ the same restrictions for acceptableHeight apply here
- Using **acceptableHeight** and **acceptableWeight** variables, **determine whether the candidate's acceptance status**. Output whether the candidate was accepted, rejected based on height only, weight only or both height and weight. Output the messages are specified in the Expected Input/Output section. (You will need to use selection statements to output the correct message).
- At the end of the program **output the number of candidates that were accepted** and **the average of candidates accepted** with respect to the overall number of candidates entered.
- For this program you will need to use a **minimum of 4 functions** (including a function to output the class heading). The functions should be designed using good functional design practices discussed in class. Here are some tips:
 - Think about code that is **repetitive** and create a function for it. Remember that if the repetitive code uses different values then make those values arguments. If you are cutting and pasting code or copying code and changing values then you are probably not writing the function properly.
 - Make sure that each function manages **1 task (not 1 line of code)**, but 1 task.
 - Functions should be generic when it makes sense to do so. In other words, don't have 2 functions to accomplish the same task with different values hard-coded! Pass those values as parameters.

Additional Requirements:

- Use appropriate data types and variable names throughout the code.
- Use constants appropriately.
- Adhere to the style guidelines for the class
- Adhere to proper style & structure guidelines for the class
- The goal of this assignment is to demonstrate the use of compound Boolean Expressions, Boolean variables, functions, selection and repetition statements, so please be careful in choosing the correct constructs, writing appropriate functions, and writing the Boolean Expressions. Don't replicate Boolean expressions – use Boolean variables rather than repeating the same expression twice.

Recommended Steps

1. Before you code. **Design your program**. Sketch out the algorithm using any method you prefer – it doesn't have to be in excel, but make sure it is readable. Think about which tasks would be good candidates for functions. Your design should include your algorithm decomposition. You can use pseudocode or a flowchart for this. You could also use a combination of code/pseudocode. However, your design needs to include the prototype an design for each function in addition to main. I recommend desk-checking your design before you begin to code to ensure your logic is correct.

Building your code incrementally will really help with this assignment. After each step **TEST YOUR CODE!**

2. Write the **input statements** (**don't error check yet!**)
3. Write the statement that will determine if the candidate has an **acceptable height**.
(Test this statement thoroughly – check the boundaries. Remember: True will output 1, False will output 0).
4. Write the statement that will determine if the candidate has an **acceptable weight**. This will be similar to acceptable height.
5. Write the statements that will **output the candidate's acceptance status**. (Test using all branches testing)
6. **Add the primary loop** that will allow the user to input candidates until the user inputs an 'X' or an 'X'.
7. Add the statements to calculate and output the **total candidates** and **percentage accepted**.
8. Add in the error checking for the inputs.
9. If you designed the code well with functions, write one function as you go to accomplish its task then test and document it.

Expected Input (in green) / Output (in blue):

Output should include your class heading. Line numbers should be printed with your code and screenIO.txt files. Run your code 3 times with the following input → the output should be exactly as indicated here. **Do not use a for loop for the 3 test runs. Execute the code 3 separate times. Each test run should put in a separate text file.**

TEST RUN #1

Please enter the candidate's information (enter 'X' to exit).

Sex: m

Height: 80

Weight: 130

This candidate has been ACCEPTED!

Double Spaced

Triple Spaced

Please enter the candidate's information (enter 'X' to exit).

Sex: m

Height: 65

Weight: 250

This candidate has been ACCEPTED!

Please enter the candidate's information (enter 'X' to exit).

Sex: f

Height: 62

Weight: 109

This candidate has been rejected based on the WEIGHT requirement.

Please enter the candidate's information (enter 'X' to exit).

Sex: x

2 candidate(s) accepted!

That's 67%!

TEST RUN #2

Please enter the candidate's information (enter 'X' to exit).

Sex: k
***** Invalid sex; please enter M or F *****
Sex: M
Height: 65
Weight: 250

This candidate has been ACCEPTED!

Please enter the candidate's information (enter 'X' to exit).

Sex: m
Height: 64
Weight: 130

This candidate has been rejected based on the HEIGHT requirement.

Please enter the candidate's information (enter 'X' to exit).

Sex: M
Height: 80
Weight: 251

This candidate has been rejected based on the WEIGHT requirement.

Please enter the candidate's information (enter 'X' to exit).

Sex: male
Height: 81
Weight: 49
***** Invalid weight; please enter a weight in lbs between 50 and 1400. *****
Weight: 129

This candidate has been rejected based on the HEIGHT and WEIGHT requirements.

Please enter the candidate's information (enter 'X' to exit).

Sex: F
Height: 75
Weight: 1401
***** Invalid weight; please enter a weight in lbs between 50 and 1400. *****
Weight: 110

This candidate has been ACCEPTED!

Please enter the candidate's information (enter 'X' to exit).

Sex: f
Height: 23
***** Invalid height; please enter a height in inches between 24 and 110. *****
Height: 76
Weight: 185

This candidate has been rejected based on the HEIGHT requirement.

Please enter the candidate's information (enter 'X' to exit).

Sex: female

Height: 111

***** Invalid height; please enter a height in inches between 24 and 110. *****

Height: 62

Weight: 109

This candidate has been rejected based on the WEIGHT requirement.

Please enter the candidate's information (enter 'X' to exit).

Sex: F

Height: 61

Weight: 186

This candidate has been rejected based on the HEIGHT and WEIGHT requirements.

Please enter the candidate's information (enter 'X' to exit).

Sex: J

***** Invalid gender; please enter M or F *****

Sex: p

***** Invalid gender; please enter M or F *****

Sex: F

Height: 63

Weight: 110

This candidate has been ACCEPTED!

Please enter the candidate's information (enter 'X' to exit).

Sex: x

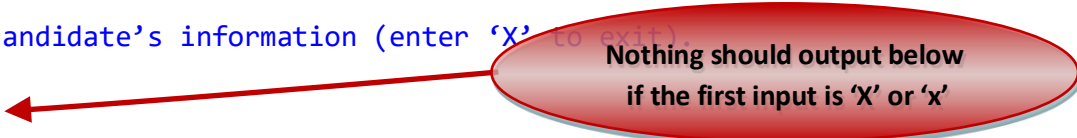
3 candidate(s) accepted!

That's 33%!

TEST RUN #3

Please enter the candidate's information (enter 'X' to exit).

Sex: x



Nothing should output below
if the first input is 'X' or 'x'

Submitting Your Assignment

Your assignment should be submitted as a Single PDF and a zip file with all of your code & the header file. I won't look at the zip file unless I find inconsistencies between your code and output. Make sure you submit all of your work right side up and in the order specified.

Create a Single PDF (in the FOLLOWING ORDER)

- 1 – Your complete design for your program (including functions).
- 2 – Output for test run #1 (on a separate page → cut and paste into a text file within eclipse)
- 3 – Output for test run #2 (on a separate page)
- 4 – Output for test run #3 (on a separate page)
- 5 – Your header file for your functions printed to pdf from eclipse
- 6 – A listing of the main function (conforming to style discussed in class → properly documented)
- 7 – A listing of each function in the order it was called from main.

Use ilovepdf.com (or software of your choice) to merge your documents into a single pdf. Be sure to check your files (they should not be over 200kb). If they are too large you may need to compress them after you merge.

You will also need to create a **zip file with main.cpp, all functions, and your header file.**

Submit your files

Go to the Assignments link in Canvas and click on Submit Assignment in the upper right-hand corner. Submit your PDF followed by your zip file (2 files).