

CS 150 Lab 5

If statement, bool data type, type casting.

The main objective of today's lab is to use `if` statements, `bool` data types and type casting to solve a complex problem.

- Be sure your output looks exactly like the specified output.
- Be sure to submit the folder 05LabPUNetID with the completed project(s) to **CS150-01 Lab** when you are done.
- Show the instructor or TA your solution to each problem before submitting it.

Coding Standards

Before beginning this lab, we are going to discuss the Coding Standards that will be used for the remainder of this course. The Coding Standards document can be found as a link on the main CS150 page.

Lab 5.1

Save this project as '05SoccerXXXXXXXX', where XXXXXXXXX is your PUNet ID. The goal of this program is to solve the following problem:

We want to calculate the number of goals scored per game by the Pacific University Men's Soccer team. So far they have scored 1, 0, 2, 0, and 0 goals in their first five games. Your program must prompt the user for the five goal totals one at a time. Calculate the average number of goals scored per game and whether or not the team averages more than 2.10 goals a game. Display both of these values to this user. In your C++ code, represent whether or not the average number of goals scored per game is greater than 2.10 as a `bool` variable. A sample input and output is shown below.

Sample output is on the following page. Before you start, answer the following questions. The instructor and TA will want to see the answers to these questions before looking at your source code.

1) Briefly describe the data and information that your program will need to use.

2) List the variable declarations necessary to store the data and information listed in 1.

3) What decision statements will you need to use in your program? For EACH decision statement specify what logical and/or relational expressions you will need to use for said decision statement.

Your output must look exactly as follows:

```
*****
*   Pacific University Men's Soccer Goal Tabulator   *
*****
Please enter the first game goal total: 1
Please enter the second game goal total: 0
Please enter the third game goal total: 2
Please enter the fourth game goal total: 0
Please enter the fifth game goal total: 0
```

The average number of goals scored per game: 1.4
On average, the team does not score more than 2.1 goals a game.

After you run your program with the given data, use the following two data sets:

Game one: 5
Game two: 2
Game three: 3
Game four: 1
Game five: 2

Game one: 2
Game two: 0
Game three: 1
Game four: 3
Game five: 1

Check the output for each of the above data sets is correct before submitting your assignment.

Challenge!

For this lab you will create a new Visual Studio Project that will contain your source code. Save this project as '05LeapXXXXXXXX', where XXXXXXXX is your PUNet ID. The goal of this program is to determine if a given year is a leap year or not.

A year is a leap year if it is divisible by 4. The only exception to this is if it is a century year. Then it is a leap year only if it is divisible by 400. In case you're wondering, these are the rules for the Gregorian calendar, which began to be adopted in 1582 when they realized that having a leap year every 4 years resulted in the days of the year being very off after a several centuries. (The actual length of a year is 365.24219 days, not an even 365.25 days).

You need to determine a single relational expression that represents the above rules and use this in your program.

Once you have completed your program, test it on the following values:

- 1900 – Not a leap year
- 1960 – A leap year
- 1989 – Not a leap year
- 2000 – A leap year

Notes for the Challenge:

C++ has logical operators that allow you to create more complex *relational expressions*. These include the **and operator** (&&) and the **or operator** (||). The *and operator* is two ampersand characters, the *or operator* is two vertical pipes (the shift of the backslash key).

An and operator is true **only if both** operands are true, an or operator is true **if at least one** operand is true.

You can use these in the following way:

```
int x = 9, y = 99;
bool result;

result = (y>x) && (x != 1); //the value stored in result is True
result = (y>x) && (x == 1); //the value stored in result is False
result = (y==x) || (x == 1); //the value stored in result is False
result = (y>x) || (x == 1); //the value stored in result is True
```

A Practice Lab

For this lab you will need to write a program called '05AreaXXXXXXXX', where XXXXXXXX is your PUNet ID that will ask the user for the length and width of a rectangle and also ask the user to input the area of the rectangle. Your code will check if the value of the area input by the user is correct. Display one of the following messages; depending on if the user-supplied value of area is correct..

- **Yes! The area of the rectangle was calculated correctly by the user!**
- **Sorry! The correct area of the rectangle is ###.**

In the second message, display the correct area of the rectangle in place of the ###.