



Lab3: if-else, nested if-else

UNT CSCE1030

TA Polina Nemkova

Options

The lab is due on today by 11.59pm

1. WORK BY YOURSELF: This lab instructions are available on Canvas. If you feel comfortable with this assignment, you can do it by yourself. Ask me to check it in the very end before you submit it.
2. WORK WITH ME: Follow the steps I describe on the slide.

What Do We Need to Submit?

Four files for two tasks:

- Lab3A.cpp
- Lab3B.cpp
- Lab3C.cpp

Step1: Prepare the working space

1. Pull up the **.pdf file for Lab3** from your canvas account;

2. Start **Putty**

(cse01.cse.unt.edu
cse02.cse.unt.edu
cse03.cse.unt.edu
cse04.cse.unt.edu
cse05.cse.unt.edu
cse06.cse.unt.edu)

Printf **and** Scanf

Why do we use it and not **cin/cout**?

printf()

printf does the same as **cout**, but faster. How to use **printf()**?

Example 1 (with string):

```
printf ("text");
```

Example 2 (with variables):

```
printf("The final values are a:%d b:%d c:%lf d:%e\n",a,b,c,d );
```

scanf()

scanf does the same as **cin**, but faster. How to use **scanf()**?

Example 1:

```
scanf("%d%d",&a,&b);
```

Format specifier	Description
%d	Integer Format Specifier
%f	Float Format Specifier
%c	Character Format Specifier
%s	String Format Specifier
%u	Unsigned Integer Format Specifier
%ld	Long Int Format Specifier

General if-else structure

```
#include<iostream>

using namespace std

int main ()
{
    ###declare values
    if (number < 5) ##condition
    {
    }
    else ##alternative
    {
    }

    return 0;
}
```


rand() function for getting a random value

function

rand

<cstdlib>

```
int rand (void);
```

Generate random number

Returns a pseudo-random integral number in the range between 0 and `RAND_MAX`.

This number is generated by an algorithm that returns a sequence of apparently non-related numbers each time it is called. This algorithm uses a seed to generate the series, which should be initialized to some distinctive value using function `srand`.

`RAND_MAX` is a constant defined in `<cstdlib>`.

A typical way to generate trivial pseudo-random numbers in a determined range using `rand` is to use the modulo of the returned value by the range span and add the initial value of the range:

```
1 v1 = rand() % 100;      // v1 in the range 0 to 99
2 v2 = rand() % 100 + 1;  // v2 in the range 1 to 100
3 v3 = rand() % 30 + 1985; // v3 in the range 1985-2014
```

Notice though that this modulo operation does not generate uniformly distributed random numbers in the span (since in most cases this operation makes lower numbers slightly more likely).

C++ supports a wide range of powerful tools to generate random and pseudo-random numbers (see `<random>` for more info).

Don't forget to seed your random value:
/ initialize random seed: */*
`srand (time(NULL));`

Submission

1. Go to WinSCP and copy your files from the CSE machine on your computer.
2. If you want, show me execution of your programs (to be sure that there is no problems).
3. Submit Lab4A.cpp, Lab4B.cpp, Lab4C.cpp on Canvas.