Computational Problem Solving (CS100)

Shafay Shamail, Arif Zaman and

Safi Ullah Chaudhary

Lab 3 Objectives:

- Practice if statements (called conditionals)

Lab Guidelines

1. Make sure you get your work graded before the lab time ends.
2. You put all your work onto the LMS folder designated for the lab (i.e. “Lab02”) before the time the lab ends.
3. Talking to each other is NOT permitted. If you have a question, ask the lab assistants.
4. If you are a hot-shot C++ expert, you are still not allowed to use any feature of C++ that has not been covered in class or in the lab.
5. The object is not simply to get the job done, but to get it done in the way that is asked for in the lab.

Review: Conditional Statements

The basic syntax of an "**if statement**" is shown at the right. Look at it carefully.

```
if (a==0)
{
        cout << "nothing" << endl;
}
```

The statement can have an **else** clause as shown in the next box.

Note where the semi-colons are present, and where they are missing.

Note the doubled equal signs, which test to see if the two sides are equal.

```
if (a==0)
{
        cout << "nothing" << endl;
}
else
{
        cout << "something" << endl;
}
```

Other **comparisons** your can do are given in the table at the right. Note that if you compare strings they will be compared in dictionary order.

`==`	equals	`!=`	not equal
`<`	less than	`<=`	less than or equal
`>`	greater than	`>=`	greater than or equal

When there are **more than two** (mutually exclusive) **alternatives**, you can use the else-if construction as shown on the right.

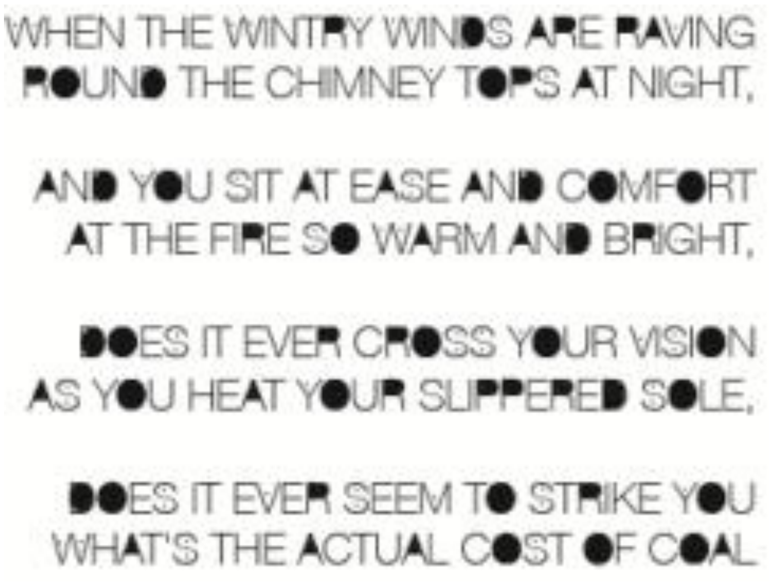
Mutually exclusive means that only one of the many possibilities can be true.

```
if (a==0)
{       cout << "nothing" << endl;
}
else if (a>0)
{       cout << "positive" << endl;
}
else // here a must be negative
{       cout << "negative" << endl;
}
```

You can combine different comparison using and, or and not.

`!(a)`	not
`0<a && a<1`	and (both sides must be true)
`a<-1 || 1<a`	or (at least one side must be true)

Task 0: Filling in HOLES


WHEN THE WINTRY WINDS ARE RAVING
ROUND THE CHIMNEY TOPS AT NIGHT,

AND YOU SIT AT EASE AND COMFORT
AT THE FIRE SO WARM AND BRIGHT,

DOES IT EVER CROSS YOUR VISION
AS YOU HEAT YOUR SLIPPERED SOLE,

DOES IT EVER SEEM TO STRIKE YOU
WHAT'S THE ACTUAL COST OF COAL

When you were really bored, I am sure you filled in holes in the letter like this in your books:

Write a program that asks for the user to enter a one digit number. It should then tell the user how many "holes" are in the digit.

For example:

```
Enter a one digit number (0, 1, ... 9): 8
The number 8 has 2 holes.
```

```
Enter a one digit number (0, 1, ... 9): 6
The number 6 has 1 holes.
```

```
Enter a one digit number (0, 1, ... 9): 3
The number 3 has 0 holes.
```

After you go home you could try to think about how you could do this for a two digit number.

How about for a four letter word, stored as a string? (You will need to google to find out how you can get the second letter of a four letter string in C++).

CHECKPOINT 1

Show your program and result to the TA.

Task 1: Making Snide Remarks

For the following program we will assume that:

On the average, freshmen are 18 years old, and
On the average most people are one year older for every year of university.

Write a program that

1. Asks the user to type in his/her age.
2. Gets the age from the user, and puts it in an integer variable called **`age`**
3. Asks the user to enter their year in LUMS, which will be a number between 1 (for freshman) and 4 (for senior).
4. Gets the year from the user, and puts it in an integer variable called **`year`**
5. If a person is older that their class average age, write out a message that says "**`Hello Grandpa`**"
6. If a person is younger than their class average age, write out "**`Hello Smarty pants`**"
7. If a person is as old as their class average age, write out "**`Hello Average`**"

There are a total of twelve possibilities (for each of four years, the person could be younger, just right, or older, but you don't need so many cases, if you compute **`age-year`**).

Try out different ages and years to make sure that your program works. The screen should look something like the following (this is just an example). The yellow writing is what the user typed in. Your program should work for whatever the user types in. Everything else is typed by the computer.

```
Enter your age: 20
Enter your year (1-4): 1
Hello Grandpa
```

CHECKPOINT 1

Show your program and result to the TA.

Task 2: String text input

We will modify the previous program so that instead of getting the **`year`** variable as an integer between 1 and 4, we will ask the user to enter a **`string`** variable called **`syear`**. This string will be a two letter string **`Fr`**, **`So`**, **`Ju`** or **`Se`**, for Freshman, Sophomore, Junior or Senior respectively.

Because we now have a string, it is not so simple to subtract **`age-syear`**. One way is to use the string **`syear`** is to use it to create a number between 1 and 4 and store it in **`year`**, and then proceed as we did before.

The easiest way to do this is to make a four-way else-if (look again at the last case on the first page for inspiration).

The screen should now look like

```
Enter your age: 15
Enter your year (Fr, So, Ju or Se): Fr
Hello Grandpa
```

The usual abbreviation for Junior and Senior is **`Jr`** and **`Sr`**. Make your program so that it will accept these abbreviations (as well as **`Ju`** and **`Se`**).

CHECKPOINT 2

Show your program and result to the TA.

Task 3: String text output

There is not much to explain. Just rework the program so that now its output looks like what is shown below

The following show different outputs that might occur based on differing text typed in to the questions.

```
Enter your age: 29
Enter your year (Fr, So, Jr or Sr): Fr
Hello Grandpa Freshie
```

```
Enter your age: 12
Enter your year (Fr, So, Jr or Sr): So
Hello Smarty pants Soph
```

```
Enter your age: 20
Enter your year (Fr, So, Jr or Sr): Jr
Hello Average Junior
```

```
Enter your age: 20
Enter your year (Fr, So, Jr or Sr): Sr
Hello Smarty pants Senior
```

CHECKPOINT 3

Show your program and result to the TA.