

# Comp 11 Lectures

Mike Shah

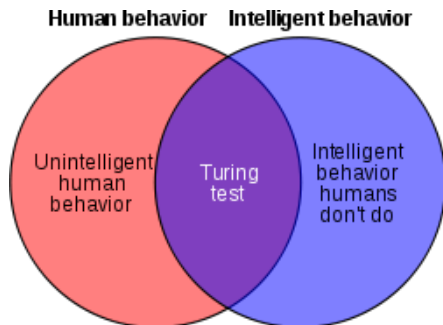
Tufts University

June 6, 2017

# Decision Structures

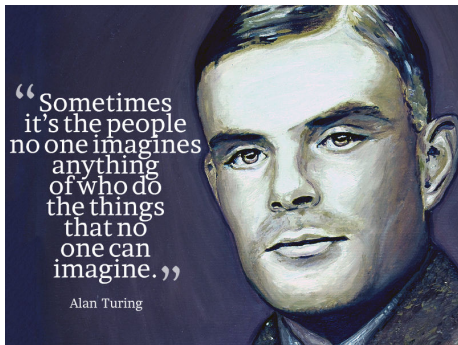
# Comp 11 - Pre-Class warm up

The Turing test, developed by Alan Turing in 1950, is a test of a machine's ability to exhibit intelligent behaviour equivalent to, or indistinguishable from, that of a human.



# Lecture

# Alan Turing



**Figure 1:** Alan Turing, one of the great Computer Scientists. You may have heard of something called the Turing Test.

# How do computers think?

Understanding how computers (appear) to make decisions.

- Can a human make the distinction between the difference of talking to a human or a robot?
- A silly question maybe a few years ago, but we're getting closer.
- So how do computers appear to make decision?

# Decision Structures



## First Tool: Selection Statements

# Conditional Statements

- Can a human make the distinction between the difference of talking to a human or a robot?
- A silly question maybe a few years ago, but we're getting closer.
- So how do computers appear to make decision?

# if-statement semantics

```
1 // Introduction to a conditional statement
2 if ( A condition is true ){
3     // Execute all of the code in between the opening '{' and
4     // closing '}'
5 }
```

Listing 1: Conceptual example of an if-statement

# if-statement

```
1 #include <iostream>
2 int main(){
3     int x = 5;
4     if (x > 3){
5         std::cout << "x is greater than 3";
6     }
7 }
```

Listing 2: First if-statement with a greater than comparison operator

## if-statement (2)

```
1 #include <iostream>
2 int main(){
3     if (true){
4         std::cout << "the conditional is true";
5     }
6     return 0;
7 }
```

Listing 3: if-statement bool forcing execution of code block

## if-statement (3)

```
1 #include <iostream>
2 int main(){
3     if (false){
4         std::cout << "the code between the curly braces will
5         not execute";
6         std::cout << "this is because the conditional is false"
7         ;
8     }
9     return 0;
10 }
```

Listing 4: Not executing code block—essentially dead code in this case

## if-statement (same code as previous slide)

```
1 #include <iostream>
2 int main(){
3     bool condition = false;
4     if (condition){
5         std::cout << "the code between the curly braces will
6         not execute";
7         std::cout << "this is because the conditional is false"
8         ;
9     }
10    return 0;
11 }
```

Listing 5: Evaluating a variable. Code will execute depending on variable state

## Some operators in conditional statements

- 1 A is greater than B
- 2 A is less than B
- 3 A is greater than or equal to B
- 4 A is less than or equal to B
- 5 A is equal to B

```
1 if (A > B)
2 if (A < B)
3 if (A >= B)
4 if (A <= B)
5 if (A == B)
```

Listing 6: C++ syntax for expression.  
Remember C++ is case sensitive

There are other operators you can look up here:

<http://www.cplusplus.com/doc/tutorial/operators/>



# if (A==B)

== is for equality

It is a very common mistake to write (A=B) within a condition, which evaluates to true, because one equals sign means assignment. Be careful!

= is for assignment

We learned this when we created variables.

```
1 e.g.  
2 int x = 5;  
3 x = 5 + 7;  
4 x == 5+7; // This evaluates  
              to true, but does not do  
              anything useful alone.
```

**Listing 7:** The number of equals signs matters!

# if/else statement

```
1 if ( a > b){  
2     Do something  
3 }  
4 // Execute any remaining code
```

Listing 8: Execute 1 block of code or do not execute it

```
1 if ( a > b){  
2     Do something  
3 }  
4 else  
5 {  
6 // Always execute this block  
7 // if the above condition  
8 // evaluates to false  
9 // In this case, a <= b must  
10 // be true  
11 }
```

Listing 9: Choose 1 block of code or another

## if/else if statement

```
1 if ( a > b){  
2     Do something , because a is greater than b  
3 }  
4 else if (a == b)  
5 {  
6     // Execute this block conditionally. If the above  
7     // condition evaluates to false  
8     // In this case, a <= b must be true  
9 }  
// execute more code below
```

Listing 10: Conditionally execute one block or the other

## if/else if/else statement

```
1 if ( a > b){
2     Do something , because a is greater than b
3 }
4 else if (a == b)
5 {
6     // Execute this block conditionally. If the above
7     // condition evaluates to false
8     // In this case, a <= b must be true
9 }
10 else {
11     // Always execute this code
12 }
```

Listing 11: Three possible blocks of code could execute

## Second Tool: Iteration

# while

```
1 #include <iostream>
2 int main(){
3     int x = 10;
4     while (x > 0){
5         std::cout << x << "\n";
6         x = x - 1;
7     }
8 }
```

Listing 12: While loop example

## while (2)

```
1 #include <iostream>
2 int main(){
3     int x = 10;
4     while (x > 0){
5         std::cout << x << "\n";
6         // x = x - 1; What happens if I do not decrement this
7         value?
8     }
```

Listing 13: Infinite Loop

# Break and Continue

```
1 #include <iostream>
2 int main(){
3     int x = 10;
4     while (x > 0){
5         std::cout << x << "\n";
6         if (x == 5){
7             break;
8         }
9     }
10    return 0;
11 }
```

Listing 14: Stopping and continuing control flow



## range based for-loop - iterate through entire collection

```
1 #include <iostream>
2 #include <string>
3 int main(){
4     std::string s = "abcdefg";
5     for(char c: s)
6         std::cout << c << "\n";
7 }
8 return 0;
9 }
```

Listing 15: Iteration over a collection

## for (a subset of data)

```
1 #include <iostream>
2 #include <string>
3 int main(){
4     std::string s = "abcdefg";
5     for (int i =0; i < 3; ++i){
6         std::cout << s.at(i) << "\n";
7     }
8     return 0;
9 }
```

Listing 16: iteration using an index

- The AND and OR statements in C++ are important for creating more powerful control flow.
- AND
- OR

## Truth Tables (And operator)

P: I have a drivers license Q: I am at least 16 years old.

You can drive if you have a drivers license and are at least 16 years old.

P	Q	P And Q
T	T	T
T	F	F
F	T	F
F	F	F

# Truth Tables (Or operator)

P: I am over 65 years old Q: I have a Student ID

You can get a discount at the movie theaters if you are over 65 OR have a student ID.

P	Q	P OR Q
T	T	T
T	F	T
F	T	T
F	F	F

# And and Or Example

```
1 if ( x > 1 && x < 10){
2     std::cout << "x is greater than 1 and less than 10 \n";
3 }
4
5 if ( x > 0 || x < 0){
6     std::cout << "x can not be 0 \n";
7 }
```

Listing 17: AND and OR example for making more complex conditional statements

# Scope

- It is worth noting that we have seen a lot of curly braces.
- The curly braces show the scope of our variables.
- The curly braces also separate what code gets execute, based on a conditional statement.



# Scope Example 1

```
1 #include <iostream>
2
3 int main() {
4
5     int x = 0;
6     if (x==0){
7         // i is locally declared and defined in a block
8         int i = 5;
9         std::cout << "x is equal to 0, and i = " << i << std::
10        endl;
11    }
12    return 0;
13 }
```

Listing 18: Creating a new variable local to a block of code

## Scope Example 2 – Error

```
1 #include <iostream>
2
3 int main(){
4
5     int x = 0;
6     if(x==0){
7         // i is locally defined
8         int i = 5;
9         std::cout << "x is equal to 0, and i = " << i << std::
10        endl;
11    }
12
13    std::cout << "x = " << x << std::cout;
14    // Is i within scope at this point?
15    std::cout << "i = " << i << std::cout;
16
17    return 0;
18 }
```

Listing 19: i is not defined in scope

## Scope Example 3 – Fixing error

```
1 #include <iostream>
2
3 int main(){
4
5     int x = 0;
6     // Move i's declaration outside of control block
7     int i = 5;
8     if (x==0){
9         std::cout << "x is equal to 0, and i = " << i << std::
10        endl;
11    }
12    std::cout << "x = " << x << std::cout;
13    std::cout << "i = " << i << std::cout;
14
15    return 0;
16 }
```

Listing 20: TBD

## In-Class Activity

<http://www.mshah.io/comp/11/activities/activity3.pdf>

## Activity Discussion

# Review of what we learned

- (At least) Two students
- Tell me each 1 thing you learned or found interesting in lecture.

5-10 minute break

# To the lab!

Lab: <http://www.mshah.io/comp/11/labs/lab3.pdf>

1

---

<sup>1</sup>You should have gotten an e-mail and hopefully setup an account at <https://www.eecs.tufts.edu/~accounts> prior to today. If not—no worries, we'll take care of it during lab!