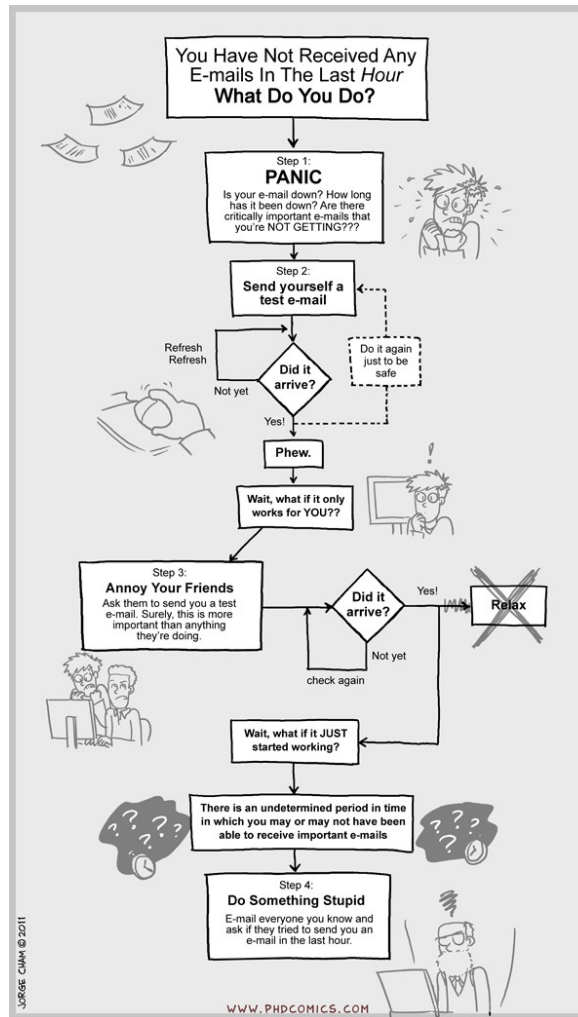


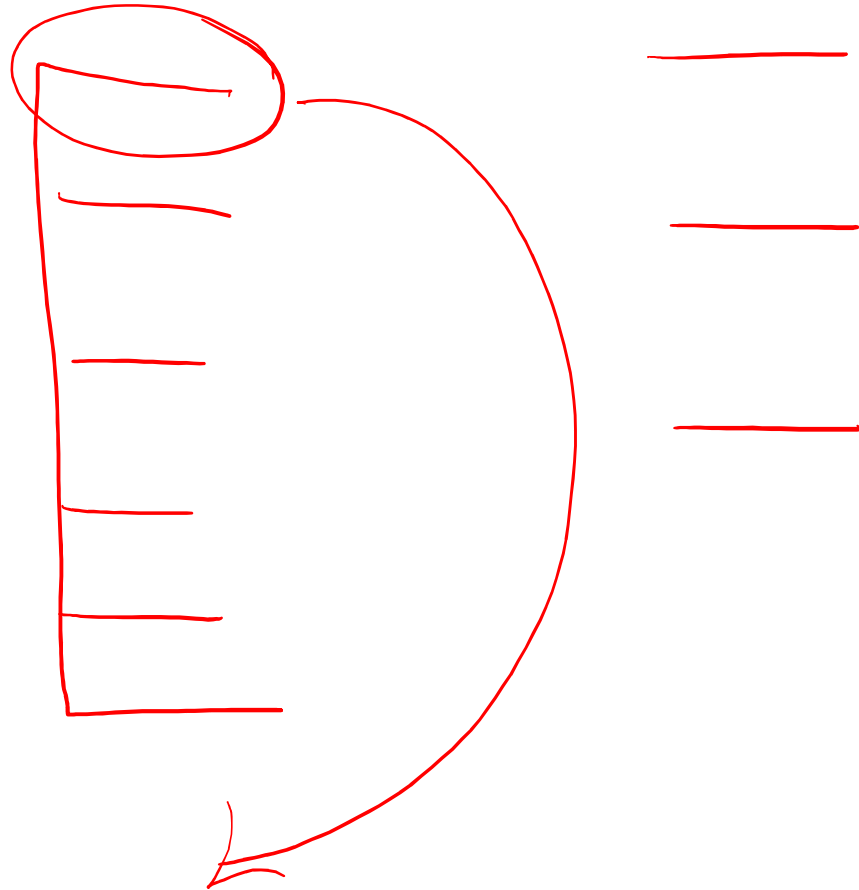
CS 161 Lecture 4 – More Conditionals – 19

January 2017



if/else if/else

```
if (<expression>) {  
    <statement>  
}  
...  
else if (<expression>) {  
    <statement>  
}  
...  
else {  
    <statement>  
}
```



Relational and Logical Operators

- == Equality
- >= Greater than or equal to
- <= Less than or equal to
- > Greater than
- < Less than
- != Not equal to
- || OR
- && AND
- ! NOT

Example

- `if(1+2)`
- `if(2-4)`
- `if(2-2)`
- `if(4==4)`
- `if((2+1) == 4)`
- `if(4.1 == 4)`
- `if(3 <= 4)`
- `if(4 >= 4)`
- `if(3.5 > 4)`
- `if(4 < 4)`
- `if(3+2*2 > 9)`
- `if((3+2)*2 > 9)`

F AD ?

F AD ? AD ?

Examples Continued

- AND: `if((1>2) && (2<5))`
- OR: `if((1>2) || (2<5))`
- NOT: `if(!(1>2) && (2<5))`

p	q	p && q	p q	!p
T	T	T	T	F
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

if/else Example

```
if (x > y) {  
    cout << "X is greater than Y" << endl;  
}
```


else if (x == y) { }

```
else {  
    cout << "Y is less than X" << endl;  
}
```

//Are these print statements always true?

if/else if/else Example

```
if (x > y) {  
    cout << "X is greater than Y" << endl;  
}  
else if (x == y) {  
    cout << "X is equal to Y" << endl;  
}  
else {  
    cout << "Y is less than X" << endl;  
}
```



Nested Decisions

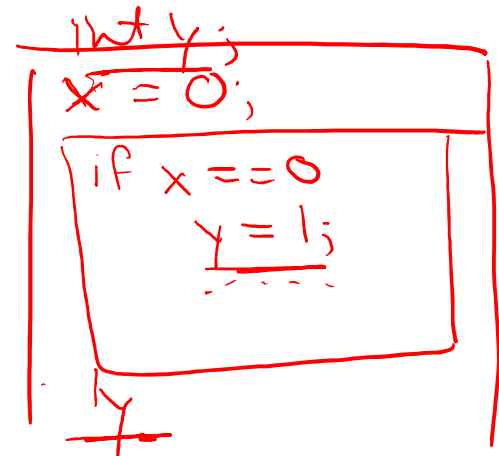
```
if (confused_on_class_procedure == True) {  
    if (procedure_in_syllabus == False) {  
        cout << "Email TA or Shannon" << endl;  
    } else {  
        bool still_confused = read_syllabus();  
        if (still_confused == True) {  
            cout << "Email TA or Shannon" << endl;  
        } else {  
            cout << "Good job!" << endl;  
        }  
    }  
}
```



Notes on Scope

- Block Scope: variables declared exist until the end of the block
- Blocks = {}

```
int main () {  
    int a = 0;  
    if (a == 0) {  
        int b = a; //a still exists in here  
    }  
    b = 1; //error: b does not exist out here  
    return 0;  
}
```



Alternative: Switch Statements

```
switch (<expression>) {  
    case <const-expression>:  
        <statement>;  
        ...  
        break;  
    case <const-expression>:  
        <statement>;  
        ...  
        break;  
    default:  
        <statement>;  
        ...  
        break;  
}
```

switch () {
 break
}

```
int a = 3;  
switch a {  
    case 1:  
        break;  
    case 2:  
        break;  
    case 3:  
        break;  
    default:  
        break;
```

Switch Statement Details

- Tests equality: `<expression> == <const-expression>`
 - If it's true, execute that code
- Need to have break statements otherwise it will fall through and execute everything else
- Default case is optional but a good idea

Demo

```
access.engr.orst.edu - PuTTY
1 #include <iostream>
2 using namespace std;
3
4 int main () {
5
6     if (1+2) {
7         cout << "1+2 worked" << endl;
8     }
9     if (2-4) {
10        cout << "2-4 worked" << endl;
11    }
12    if (2-2) {
13        cout << "2-2 worked" << endl;
14    }
15    if (4 == 4) {
16        cout << "4 == 4 worked" << endl;
17    }
18    if ((2+1) == 4) {
19        cout << "(2+1) == 4 worked" << endl;
20    }
21    if (4.1 == 4) {
22        cout << "4.1 == 4 worked" << endl;
23    }
24    if (3 <= 4) {
```

24,2-9 Top

```
22         cout << "4.1 == 4 worked" << endl;
23     }
24     if (3 <= 4) {
25         cout << "3 <= 4 worked" << endl;
26     }
27     if (4 >= 4) {
28         cout << "4 >= 4 worked" << endl;
29     }
30     if (3.5 > 4) {
31         cout << "3.5 > 4 worked" << endl;
32     }
33     if (4 < 4) {
34         cout << "4 < 4 worked" << endl;
35     }
36     if (3+2*2 > 9) {
37         cout << "3+2*2 > 9 worked" << endl;
38     }
39     if ((3+2)*2 > 9) {
40         cout << "(3+2)*2 > 9 worked" << endl;
41     }
42     if((1 > 2) && (2 < 5)) {
43         cout << "(1 > 2) && (2 < 5) worked" << endl;
44     }
45     if ((1 > 2) || (2 < 5)) {
```

24,2-9

70%

```
40         cout << "(3+2)*2 > 9 worked" << endl;
41     }
42     if((1 > 2) && (2 < 5)) {
43         cout << "(1 > 2) && (2 < 5) worked" << endl;
44     }
45     if ((1 > 2) || (2 < 5)) {
46         cout << "(1 > 2) || (2 < 5) worked" << endl;
47     }
48     if (!(1 > 2) && (2 < 5)) {
49         cout << "!(1 > 2) && (2 < 5) worked" << endl;
50     }
51
52
53     return 0;
54 }
```

```
~
~
~
~
~
~
~
~
~
~
```

40,2-16

Bot

Vimrc -> in home directory vim .vimrc

access.engr.orst.edu - PuTTY

- □ ×

```
1 set smartindent
2 set mouse=a
3 set nu
4 colorscheme default
```

```
~
~
```

Feedback

<https://tinyurl.com/yau3323k>