

# CS 161

## Intro to CS I

More Programming and Conditional  
Statements

# Odds and Ends...

- Make demo appointment (signup homepage)
- Assignment #2 posted/due next Sunday.
- Recitation quiz (study)

# Additional Operators

- Common operation: fetch/store same variable

var=var + 2; //increment variable contents

var=var \* 2; //double variable contents

- Assignment/operator combination (all ops supported):

var += 2;

var \*= 2;

~~var = var + 2;~~

- Pre/Post increment/decrement: ++ and –

- Example: age++ vs. ++age

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int age=5;
7
8     cout << age << endl; //This will print 5
9     cout << "post: " << age++ << endl; //Print age before incrementing, 5
10    cout << age << endl; //However, 6 is stored in age after ++ from above
11    cout << "pre: " << ++age << endl; //Increment age before printing, 7
12    cout << age << endl; //Age is 7 after ++ from above
13
14    //Undefined behaviour left up to compiler
15    //You would expect 8 8, and age would be 9 afterward
16    //g++ an clang++ give different answers
17    cout << ++age << " " << age++ << endl;
18
19    return 0;
20 }
```

2. ENGR

Re-attach Fullscreen Stay on top Duplicate Close

```
flip1 ~/cs161/private 154% g++ conditionals.cpp
flip1 ~/cs161/private 155% a.out
5
post: 5
6
pre: 7
7
9 7
flip1 ~/cs161/private 156% clang++ conditionals.cpp
conditionals.cpp:17:12: warning: multiple unsequenced modifications to 'age'
      [-Wunsequenced]
    cout << ++age << "  " << age++ << endl;
           ^           ~~~
1 warning generated.
flip1 ~/cs161/private 157% a.out
5
post: 5
6
pre: 7
7
8 8
flip1 ~/cs161/private 158% g++ conditionals.cpp -Wall
conditionals.cpp: In function 'int main()':
conditionals.cpp:17:42: warning: operation on 'age' may be undefined [-Wsequence-point]
    cout << ++age << "  " << age++ << endl;
           ^
```

flip1 ~/cs161/private 159% █

# Decisions in Life

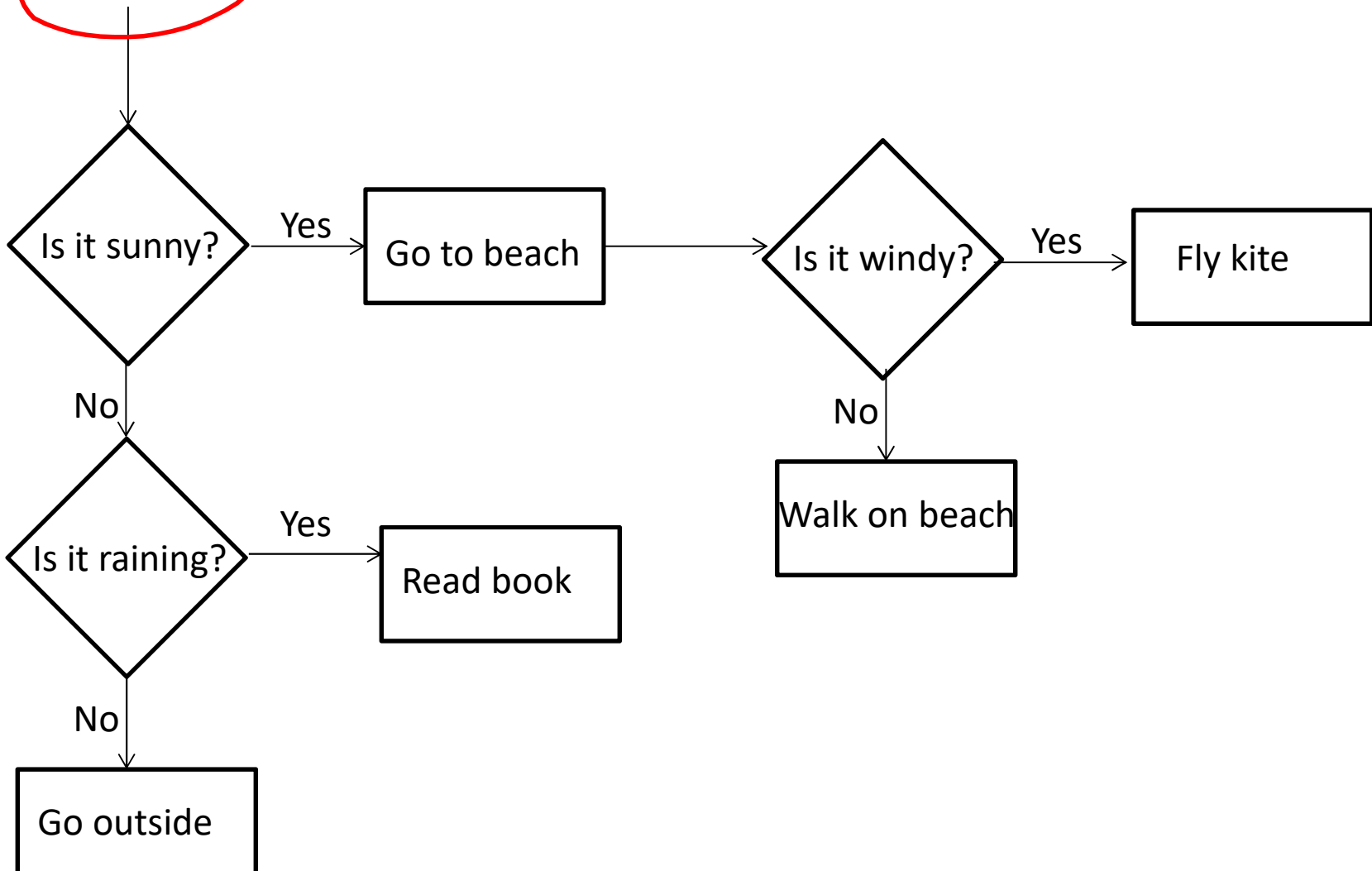
- What is a decision?
- When do we make decisions?
- How do we make decisions?
  - If it is sunny today
  - then I'll go to the beach and fly a kite
  - Else if it is raining today
  - then I'll stay inside and read a book
  - Else if it is snowing
  - then I'll go to the mountains to ski

# Decisions within Decisions

- What happens if there is no wind at the beach?
- How does this change our decisions?
  - If it is sunny today
    - then I'll go to the beach
      - if it is windy at the beach
        - then I'll fly a kite
        - if it is not windy at the beach
          - then I'll walk on the shore

*nested decisions*

# Flow chart for decisions





# Decisions in our programs

- Use an if/else

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

if (-1)

-

# What is the <expression>?

Could be a relational expression:

<expression> <relational op> <expression>

- Relational Ops

== - equal to

!= - not equal to

< - less than

> - greater than

<= - less than or equal to

>= - greater than or equal to

*binary*

# Examples

- $\text{if}(2 + 1)$
- $\text{if}(2 - 4)$
- $\text{if}(2 - 2)$
- $\text{if}(4 == 4)$
- $\text{if}((2+1) == 4)$
- $\text{if}(4.1 != 4)$
- $\text{if}(3 \leq 4)$
- $\text{if}(4 \geq 4)$
- $\text{if}(3.5 > 4)$
- $\text{if}(4 < 4)$
- $\text{if}(3+2*2 > 9)$
- $\text{if}((3+2)*2 > 9)$

*Handwritten red annotations:*  
- An arrow labeled "true" points to the first expression  $2 + 1$ .  
- An arrow labeled "true" points to the second expression  $2 - 4$ .  
- An arrow labeled "false" points to the third expression  $2 - 2$ .  
- A checkmark is drawn to the right of the second and third expressions.

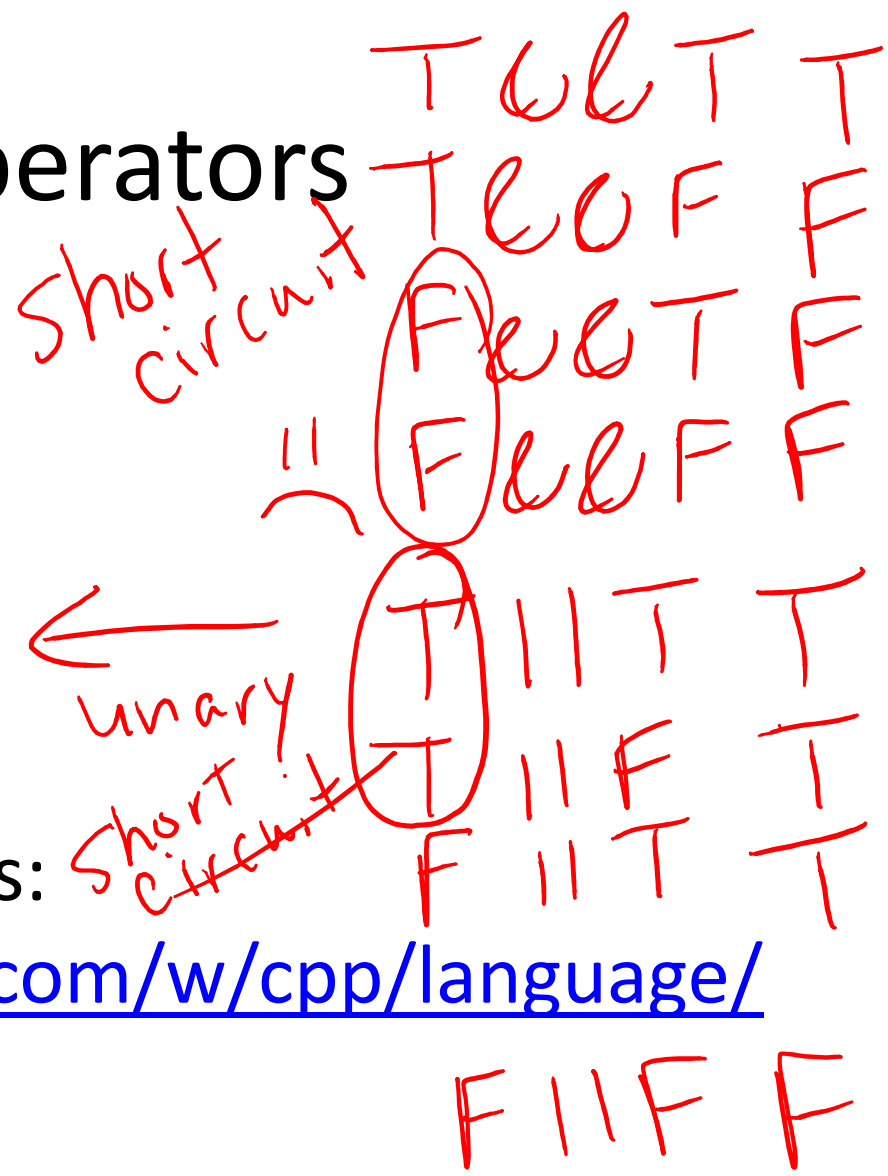
# Logical Operators

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

*! T F ! F T*

- Precedence of Operators:

[http://en.cppreference.com/w/cpp/language/operator\\_precedence](http://en.cppreference.com/w/cpp/language/operator_precedence)



*int x = 2, y = 3;*

## C++ If/Else Syntax...

```
if( x > y) {  
    std::cout << "X is greater than Y\n";  
}  
else {  
    std::cout << "X is less than Y\n";  
}
```

- When does this logic fail?

*int x = 2, y = 2;*  
*letter*

## C++ If/Else...

```
if( x > y) {  
    std::cout << "X is greater than Y\n";  
}  
else if( x < y) {  
    std::cout << "X is less than Y\n";  
}  
else {  
    std::cout << "X is equal to Y\n";  
}
```

*if ( > )*

*if ( < )*

*if ( == )*