CS 161
Intro to CS I

Loops/Begin Functions
Odds and Ends...

• Recitation Design for Assignment 3 is due Sunday on Canvas!!!

• Last day to demo Assignment 1!!!

• Read: conditionals (if/else and switch), loops (for, while, do-while), and begin functions
The while loop

1. for(x=1; x <= 100; x++)
   cout << "hello world\n";

VS.

1. int x=1;
2. while(x<=100) {
   cout << "hello world\n";
   x++;
}

0 or more
Common Mistakes

```c++
int x = 1;
while (x <= 100) {
    cout << "hello world\n";
    x++;  // What if we forget this?
}
```

What if we forget this?
The do/while loop

int x=1;
do {
    cout << "hello world\n";
    x++;
} while(x<=100);

• Difference b/w while and do/while?
```cpp
float f;
char c;

// cin ignores all leading whitespace,
// then reads characters of its expected
// type until it gets to a character not
// of its type OR whitespace without
// consuming the char not of its type or whitespace
do {
    // can use to see if a cin failed and then clear the failbit and ignore bad
    // this will not catch 4.6 error!!!!
    if(cin.fail()){
        cin.clear(); // clear the failbit
        cin.ignore(256, '\n'); // ignore/disregard our the input
    }
    cout << "enter an int: ";
    cin >> x; // what happens when you enter a non-integer or newline/enter?
    cout << "value of x: " << x << endl;
} while(cin.fail()); // don't forget the semicolon at the end of do-while

// enter a float: 
 cin >> f;
 cout << "value of f: " << f << endl;
```
The for Loop Examples

for(x=-100; x <= 100; x++)
    cout << “hello world” << endl;
for(x=2+2; x <= 17*3; x++)
    cout << “hello world” << endl;
for(x=0; x <= 100; x++)
    cout << “hello world” << endl;
for(x=0; x < 100; x++)
    cout << “hello world” << endl;
for(x=-100; x <= -1; x++)
    cout << “hello world” << endl;
The for Loop Examples

for(x=1; x <= 1; x++) {
    cout << "hello world" << endl;
}
for(x=1; x < 1; x++) {
    cout << "hello world" << endl;
}

• Why is it better to use curly braces?
The for Loop Pattern

for(<variable> = n; <variable> \leq p; <variable>++ ) {  
    <statement>;
    ...
}  
for(<variable> = n; <variable> \geq p; <variable>-- ) {  
    <statement>;
    ...
}
Nested for Loops

```cpp
for(x = 0; x < 10; x++) {
    for(y = 0; y < 10; y++) {
        cout << “hello world” << endl;
    }
}
```

• How many times is Hello World printed?

100
Reuse Variables

```cpp
for(x = 0; x < 10; x++) {
    cout << "The value of x is: " << x << endl;
}
for(x = 0; x < 10; x++) {
    cout << "The value of x is: " << x << endl;
}
```
Variables with same name

```cpp
int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 10; x++) {
        cout << "The value of x is: " << x << endl;
    }
}
```

• What is the output from this nested loop?
Infinite Loops

```cpp
int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 5; x++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```cpp
int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; y < 5; x++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```cpp
int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; x < 5; y++) {
        cout << "The value of x is: " << x << endl;
    }
}
```
Infinite Loops

```c++
int x;
for(x = 1; x <= 10; x++) {
    cout << "The value of x is: " << --x << endl;
}
```
Why is this good/bad?

for(int x = 0; x < 10; x++) {
    for(int y = 0; y < 10; y++) {
        cout << "hello world" << endl;
    }
}

• Where can we access x and y?

Can't access x or y
Can't access y
Looping Recap...

• for loops
  – Repeat for specific number of times
  – Example?

• while loops
  – Repeat while a condition is being met
  – Example?

• do while loops
  – Always do once, and repeat while condition is met
  – Example?
In-Class Exercise #2

• How would you determine if user entered a good positive int without using `cin.clear()` and `cin.ignore()`?

```
for (int i = 0; i < str.size(); i++)
    for all chars in string
        if (char is not 0-9) &
            bad = true,
            (str.at(i) != '0' &
             str.at(i) != '9')
```
```cpp
bool bad_input;
// cin ignores all leading whitespace,
// then reads characters of its expected
// type until it gets to a character not
// of its type OR whitespace without
// consuming the char not of its type or whitespace

do {
    bad_input=false; // assume good input each time, until given otherwise
    // can use to see if a cin failed and then clear the failbit and ignore bad
    /*if(cin.fail()){
        cin.clear(); // clear the failbit
        cin.ignore(256, '\n'); // ignore/disregard our the input
    }*/
    cout << "enter an int: ";
    cin >> s; // what happens when you enter a non-integer or newline/enter?
    cout << "value of x: " << s << endl;
    for(int i=0; i<s.size(); i++) { // for all the chars in s
        if((s.at(i) >= '0' && s.at(i) <= '9')) // check each to see if 0-9 digit
            bad_input=true;
    }
} while(bad_input);

x=atoi(s.c_str()); // now you know the string is good, so change ascii to int
```