ECE/CS 151
Intro to Programming I
Repetition (For vs. While Loops)
Chap. 3.5 - 3.8
Things to address...

• Assignment #1 Mistakes
  – Not following directions
  – No document provided
  – Program doesn’t compile
  – printf w/o any message

• Assignment #2 Mistakes
  – scanf(“%f”, &radius); when radius is a double
  – Use = instead of ==
Odds and Ends...

• What is in scanf?
  – %
  – *
  – width
  – type

• Helpful functions:
  – atoi()
  – isdigit(), tolower(), toupper() – #include <ctype.h>
```c
#include <stdio.h>

int main () {
    int x;
    char c;

    printf("Enter a number: ");
    scanf("%d", &x);
    //getchar(); /*Read the newline character*/
    printf("Enter a character: ");
    scanf("%c%c", &c);

    printf("%d  %c\n", x, c);

    return 0;
}
```

Better Solution
Multiple Decisions

• What if I want to make these same decisions for the whole year?
  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
  else if it is not windy at the beach
    then I’ll walk on the shore
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

• Repeat the process for 365 days
How do we do this?

• Repetition: for loops
  – Semantics
    • Repeat for a specific # of iterations w/ starting point, ending point, and an increment
  – Syntax
    for(x=1; x <= 365; x++) {
      <statement>;
      <statement>;
      ...
    }

The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

Starting point: Initialization
The for Loop

for (x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

Ending point: Continuation Test
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

• What do you notice about order?
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

• Same as x = x+1
• What about x = x + 2?
The for Loop

for(x=1; x <= 365; x++) {
    <statement>;
    <statement>;
    ...
}

• What do you notice about order?
The for Loop

for(x=1; x <= 365; x++) {
  <statement>;
  <statement>;
  ...
}

Test is False: Execution after loop
The for Loop Examples

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

```c
for(x=1; x <= 1; x++) {
    printf("hello world\n");
}
for(x=1; x < 1; x++) {
    printf("hello world\n");
}
```

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

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

for(x = 0; x < 10; x++) {
    for(y = 0; y < 10; y++) {
        printf("hello world\n");
    }
}

• How many times is Hello World printed?
Reuse Variables

```c
for(x = 0; x < 10; x++) {
    printf(“The value of x is: %d\n”, x);
}
for(x = 0; x < 10; x++) {
    printf(“The value of x is: %d\n”, x);
}
```
Variables with same name

int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 10; x++) {
        printf("The value of x is: %d\n", x);
    }
}

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

```c
int x;
for(x = 0; x < 10; x++) {
    for(x = 0; x < 5; x++) {
        printf("The value of x is: %d\n", x);
    }
}
```
Infinite Loops

int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; y < 5; x++) {
        printf("The value of x is: %d\n", x);
    }
}

Infinite Loops

```c
int x, y;
for(x = 0; x < 10; x++) {
    for(y = 0; x < 5; y++) {
        printf("The value of x is: %d\n", x);
    }
}
```
Infinite Loops

int x;
for(x = 1; x <= 10; x++) {
    printf(“The value of x is: %d\n”, --x);
}

The while loop

```c
for(x=1; x <= 100; x++)
    printf(“hello world\n”);
```

**VS.**

```c
int x=1;
while(x<=100) {
    printf(“hello world\n”);
    x++;
}
```
Common Mistakes

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

What if we forget this?
The do/while loop

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

- Difference b/w while and do/while?