Lecture 3

CIS 208

January 21st, 2005

myprog1.c

```
#include <stdio.h>
int  main(void) {
char a = 'a';
for (;;++a){
    if (!(a%2)) continue;
    printf("%c\n",a);
    if (a == 'g') break;
}
return 0;
}
```

```
%>myprog1
a
c
e
g
%>
```

myprog2

```
#include <stdio.h>
int main(void)
{char a;
while (a != ' ') {
scanf("%c", &a);
printf("%c\n",a);
}
return 0;
}
```

```
%>myprog2
aa s
Segmentation fault (core
                        dumped)

%> myprog2
aa s
a
a
%>
```

Increment & decrement

- j++ : j = j +1;
- j-- : j = j –1;

- Can not increment expressions, or values. Can't mix.

```
int j,k=2;
j = (k +2)++;
j = ++(k+2);
++5;
(++j)--;
```

WRONG
WRONG
WRONG
WRONG

Prefix and Postfix

++j != j++

Prefix: increment occurs before evaluation

Postfix: increment occurs after evaluation

```
int j,k,m = 2;
j = ++m;
k = m++;
```

< j := 3; k := 3; m := 5>

```
char a,b,c = ‘d’;
a = c--;
b = --c;
```

<a := ‘d’;b := ‘b’; c := ‘b’>

Conditional Operator ?

```
int j,k=10;
if (k <11) j = 5;
else j = 6;
```

```
j = (k<11) ? 5 : 6
```

Conditional operator ?

```
if (expr1) expr2;
else expr3;
```

```
expr1 ? expr2 : expr3;
```

Some examples

```
j = (l < m)? l+2: m –3;

(!(a%2)) ? continue : printf("%c\n",a);

(k =foo()) ? return k: ++k,return 0;
```

If-Else ladders

if {statements}

else if {statements}

.

.

.

else {statements

Switch cases

```
switch (expr) {
case 1:
     statements;
   break;
case 2:
     statements
   break;
. . .
default :
  statements;
```

```
int j;
if (j == 0) k = 2;
else if (j ==1) {
        k = 3;
        ++k;}
else if (j == 2) k = 4;
else k = 5;
```

```
int j;
switch (j) {
case 0:
        k = 2;
        break;
case 1: k = 3;++k; break;
case 2: k = 4; break;
default : k = 5;
}
```

```
char a;
switch (a) {
case ‘b’: break;
case ‘c’ : ++a;
case ‘d’ : printf(“%c\n”,a); break;
}
```

Precedence

a + b * c

a +(b * c)

a * b / c

(a * b) / c

a += b += c

a += (b +=c)

Precedence Chart

- page 84

- See webpage.

```
j = 2, k = 2;
printf("%d\n",j++);
printf("%d\n",++k);
```

For you to do

```
int j = 0;
printf("%d %d\n",j, j++);
```

What does this print out?