ECE 220 Computer Systems & Programming

Lecture 8 – Control Structures

September 21, 2017

- Midterm 1: Thursday, 9/28, 7pm to 8:30pm
- 1 sheet of notes allowed: A4/Letter size, hand-written, double-sided
- Conflict sign-up is due today at 10pm
- HKN review session: Saturday, 9/23, 11am-1pm, ECEB 1015

ECE ILLINOIS

ILLINOIS

Basic I/O (From Lecture 7)

#include <stdio.h>

- **printf examples**

```
printf("%d is a prime number", 43);
printf("43 + 59 in decimal is %d\n", 43+59);
printf("a+b=%f\n", a+b);
printf("%d+%d=%d\n", a, b, a+b);
```

- **scanf examples**

```
scanf("%c", &nextchar);
scanf("%f", &radius);
scanf(%d %d", &length, &height);
```

Formatting option: %d, %x, %c, %s, %f, \n,

Use "**man**" to look up library functions

C Programming Exercise (from Lecture 7)

```
int main(){
/* declare integer variables x, y and z */

/* set x to 5, set y to 3 */


/* increment x by 4 */

/* left shift x by y and then store the result to z */

/* print x, y, and z */

return 0;
}
```

Control Structures

Conditional Constructs

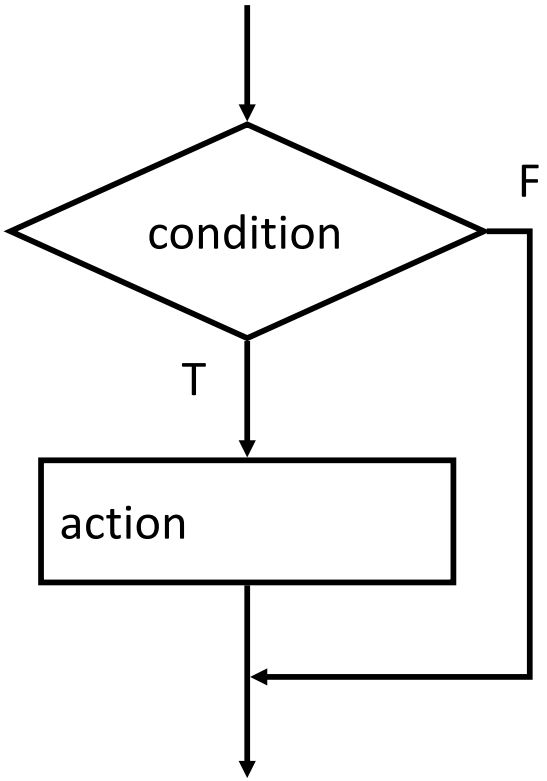
- if
- if - else
- switch

Iteration Constructs (loops)

- while
- do - while
- for

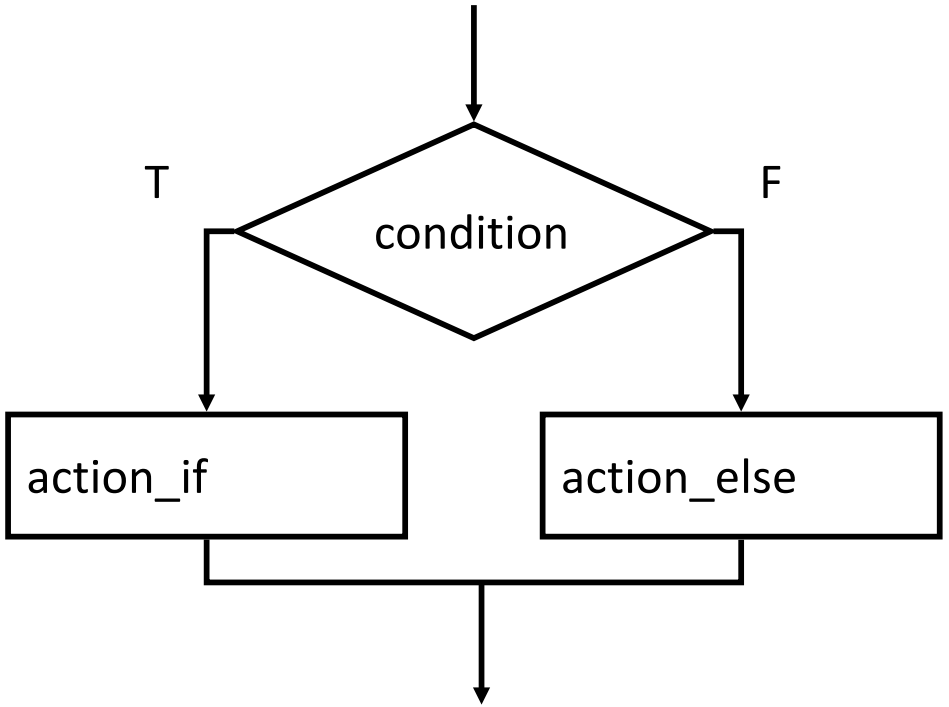
The if Statement (similar to BR in LC-3)

```
int x;
… //assign some value to x
if (x < 0)
   x = -x; //invert x only if x < 0


int y = 0;
if ((x > 5) && (x < 25))
{
   y = x * x + 5;
   printf("y = %d\n", y);
}
```



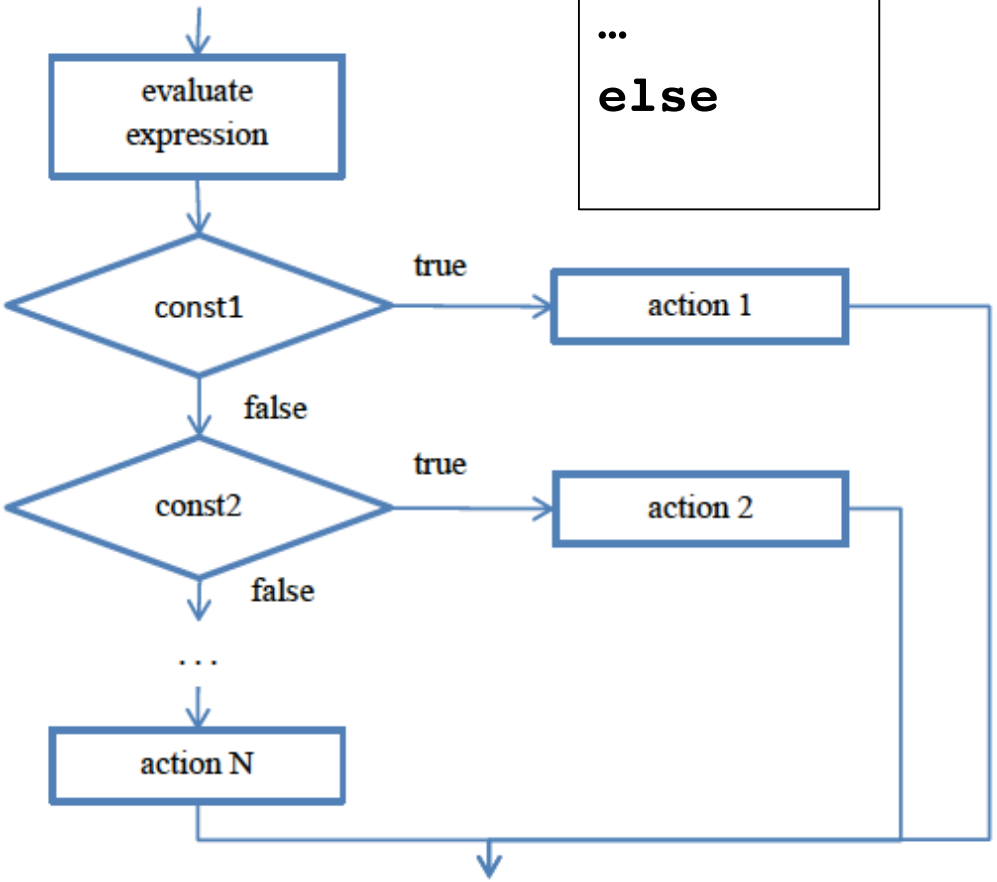
The if - else Statement

```
/*x and y are of type int*/
if (x < 0)
   x = -x;
else
   x = x * 2;

if ((x > 5) && (x < 25))
{
   y = x * x +5;
   printf("y = %d\n", y);
}
else
   printf("x = %d\n", x);
```



The switch Statement

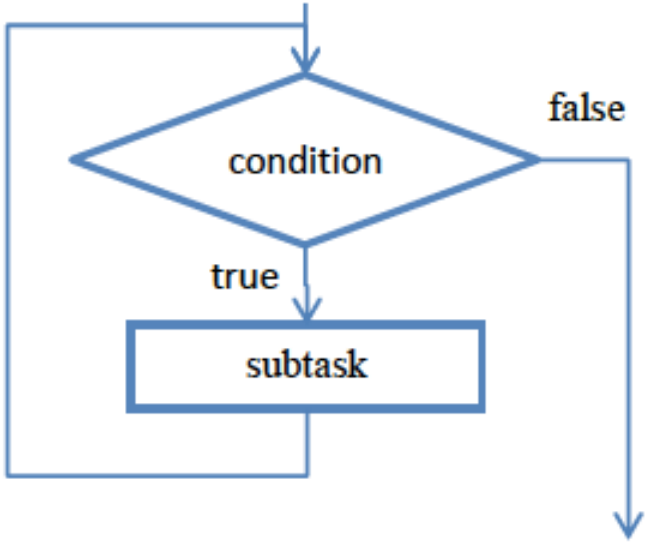
```
if
else if
else if
…
else
```




```
switch (expression)
{
    case const 1:
            action 1;
            break;
    case const 2:
            action 2;
            break;
    ...
    default:
            default action;
            break;
}

// notice the use of 'break'
```

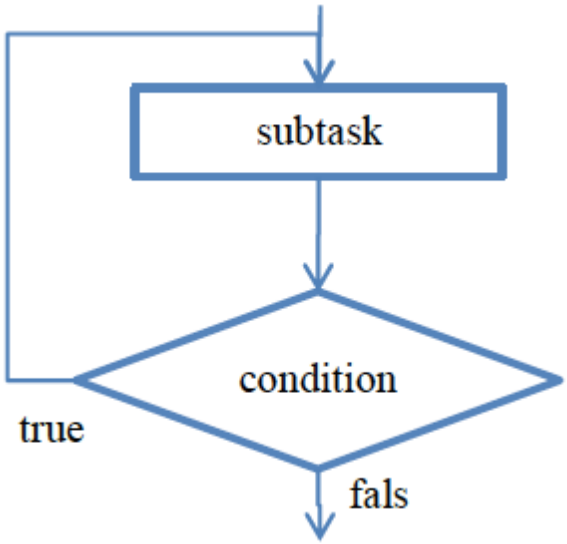
The while / do - while Statement

while: loop body may or may not be executed even once



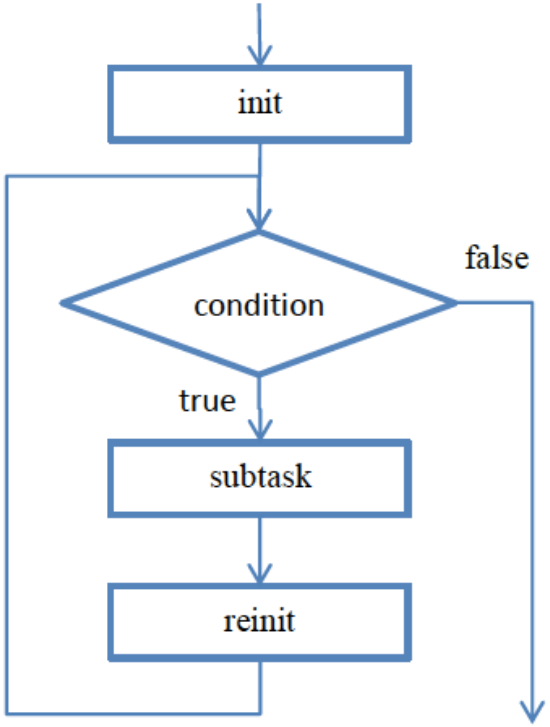

```
int x = 0;
while (x < 10) {
    printf("x=%d\n", x);
    x = x + 1;
}
```

do – while: loop body will be executed at least once




```
int x = 0;
do {
    printf("x=%d\n", x);
    x = x + 1;
} while (x < 10);
```

The for Statement




```
int x = 0;
while (x < 10) {
    printf("x=%d\n", x);
    x = x + 1;
}
```

```
int x;
for (x = 0; x < 10; x++)
{
    printf("x=%d\n", x);
}
```

- What would cause while loop or for loop to become infinite loops?

```
for (x = 0; x < 10; x++){
    if (x == 5)
        break;
    printf("x=%d\n", x);
} /* what would be the print out? What if
'break' is replaced with 'continue'? */
```

Nested Loops

inner loop is nested within the outer loop (similar to print hex example in LC-3)

```
for ()
{
    for ()
    {
        …
    } // inner loop to shift 4 bits to calculate each digit
    …
} // outer loop to print the 4 digits
```

Exercise:

1. Write a program to print an n x n identity matrix using nested loops.
2. What are some ways to break out of nested loops?
3. How to take user input for the value of n, for which n has to be >0 and <10?