Control Structures

CS2023 Winter 2004

Outcomes: Control Structures

- “C for Java Programmers”, Chapter 4
- After the conclusion of this section you should be able to
  - Work with all control stuctures in C
  - Identify the differences with Java's control structures

Control Structures in C

- Differences between Java & C control structures:
  - No boolean types: 0 is false, nonzero is true
  - C doesn't support labelled `break` and `continue`
  - Can't initialize variables on the fly:

```
for(int i = 0; i < length; i++) { ...}
```

if statements

- The body of the **`if`** statement is indented to the right, and all its instructions are aligned.

```
if(count == 0)
  printf("No values entered\n");
```

- Curly braces within a conditional statement when *only one* statement is present aren't necessary, but can be useful if more statements added later

```
if(condition) {
  single statement1
} else {
  single statement2
}
```

While loops

- A **while(1)** loop is equivalent to:

```
for(;;) {
        body
}
```

- The following

```
while(expr != 0)
    statement;
```

is equivalent to:

```
while(expr)
    statement;
```

for loops

- Idiom:

  for(i =0, *further initialization*; i < N; i++, *processing)*

- Sum numbers from 1 to 10:

```
sum =0;
for (i =1; i<= 10; i++)
   sum += i;
```

- Or:

```
for(i = 1, sum = 0; i <= 10; sum += i, i++)
;
```

- Not: (error in *C for Java Programmers*, p. 80-81)

```
for(i = 1, sum = 0; i <= 10; i++, sum += i)
;
```

Switch

```
switch(c) {
 case ' ' : cblank++;
            break;
 case '\t': ctabs++;
            break;
 case '*' : cstars++;
            break;
 default  : if(c >= 'a' && c <= 'z')
               clower++;
            break;
 }
```

Loops with empty body

If the body of the loop is empty, then the corresponding semicolon is always placed on a separate line, indented to the right:

```
for(i = 1, sum = 0; i <= 10; sum += i, i++)
      ;
```

```
for(i = 1, sum = 0; i <= 10; sum += i, i++);
```