(4-3) Selection Structures II in C
H&K Chapter 4

Instructor – Beiyu Lin

CptS 121 (May 15th, 2019)

Washington State University

Nested `if` statements (1)

- Consider the following scenario:

  Count the number of people in the room:

  - Only count on weekday
  - For each weekday, only count at 11am

Nested `if` statements (2)

- We can write a *nested* if statement to handle this situation:

```
char date;
int time;
int number_people;

if (date == 'M' || date == "T" || date = "W" || date = "T" || date = "F")
{
      if (time == 11)
      {
         /* call the function, count_number_people. */
         number_people = count_number_people();
      }
}
```


Nested `if` statements (3)

- Consider the following updated scenario:

  A high school baseball team awards merit points to players based on their offensive performance *and* the class standing ('f' = freshman, 'o' = sophomore, 'j' = junior, and 's' = senior). In particular, freshmen and sophomores earn an extra point for home runs, whereas juniors and seniors do not earn any points for singles. Write a C if-statement that, given an at-bat character and a class standing character, properly awards points.

Nested `if` statements (4)

- We can write an even more *nested* if statements to handle this situation:

```
char at_bat, class_standing;
int points;
…
if (at_bat == 's') {        /* single */
  if (class_standing == 'f') || (class_standing == 'o') {
          points = 1;
  } else {
   points = 0;
 }
} else if (at_bat == 'd') { /* double */
        points = 2;
} else if (at_bat == 't') { /* triple */
        points = 3;
} else if (at_bat == 'h') { /* home run */
   if (class_standing == 'j') || (class_standing == 's') {
        points = 4;
    } else {
        points = 5;
    }
} else { /* out */
    points = 0;
}
```


Nested `if` statements (3)

- Nested if statements vs. compound conditionals
  - Consider the following scenario: The National Weather Service would like to identify hourly weather reports in which the relative humidity is low (below 20%, the temperature is pleasant (between 75 and 85), and the winds are calm (0 to 10 m.p.h.). Assuming that the variables `humidity`, `temp`, and `wind_speed` hold those values, write an `if` statement that prints out a message when the conditions are met.

Nested `if` statements (4)

- Nested if statements vs. compound conditionals (cont.)
  - Alternative 1: Nested `if`

```
if (humidity < 20)
{if (temp >= 75)
  {if (temp <= 85)
     {if (wind_speed <= 10){printf("Perfect conditions!\n");}}
  }
}
```

  - Alternative 2: Compound `if` conditional

```
if ((humidity < 20) && (temp >= 75) && (temp <= 85)
    && (wind_speed <= 10)
  {
      printf("Perfect conditions!\n");
  }
```


Nested `if` statements (5)

- Important to note that the C compiler always matches an `else` with the most recent incomplete `if`
  - Example:

```
if (humidity < 20)
{
  if (temp <= 32)
  {
    printf("It's a cool, dry day.");
    if (wind < 10)
    {
      printf("Luckily, the winds are calm.");
    }
  }
  else
  {
    printf("The humidity is low, it's above freezing.");
  }
```

intended to do

Actually do

Nested `if` statements (6)

- Guidelines for using nested if statements
  - Use braces to enclose all if branches, even if they contain only one statement
    - This will obviate the problem of mismatching `if` and `else` branches
  - If possible, structure conditions so each alternative falls on false branch of previous condition (else if…)
  - In conditionals, don't mistake `=` for `==`
    - The C compiler won't be able to catch this error, and you're condition will always evaluate to the same Boolean result!

References

- J.R. Hanly & E.B. Koffman, *Problem Solving and Program Design in C (8th Ed.)*, Addison-Wesley, 2016.
- P.J. Deitel & H.M. Deitel, *C How to Program (7th Ed.)*, Pearson Education , Inc., 2013.

Collaborators

- Chris Hundhausen
- Andrew O'Fallon