The `switch` Statement and Programming Choice Structures

Larry Caretto
Computer Science 106

Computing in Engineering and Science

March 2, 2006

California State University Northridge

Outline

- Review choice programming
  - Statements (if, if-else, if-else-if)
  - Conditions; relational and logical operators
  - Type bool variables
  - Nested if statements
- End-of-file tests
- Programming exercises

Review Choice (if statements)

- Three structures if, if-else, and if-else-if
- Based on statement if (***<condition>***)
- Condition used relational operators (<, >, <=, >=, ==, !=) and logical operators not(!) and(&&) or(||)
- Condition evaluates to true or false
- In if-else and if-else-if only one block of code is executed

Review Type bool Variables

- Type bool variables have two possible values: true and false
- Can be used to hold result of expressions that give these values
- leapYear = year % 4 == 0 && (year % 100 != 0 || year % 400 == 0)
- Test bool variables in if statements and use with logical operators

```
if ( leapYear && month == 2 ) days = 29;
```


Review DeMorgan's Laws

- Have two bool expressions, a and b, that can have values of true or false
- Combinations of conditions for a and b satisfy both of the following
- !(a && b) = !a || !b
- !(a || b) = !a && !b
- Proved these using truth table
- Application to data validation follows

Review Data Validation

- Apply DeMorgan's Law to Validation

```
badData = x < Min || x > Max;
goodData = x >= Min && x <= Max;
goodData = !badData;
goodData = !(x < Min || x > Max);
!(a || b ) = !a  && !b
goodData = !(x < Min) && !(x>Max)
goodData = x >= Min && x <= Max
```

Exercise

- An example of an iteration problem, shown below, computes x = $\sqrt{A}$

$$x^{(n+1)} = \frac{x^{(n)}}{2} + \frac{A}{2x^{(n)}}$$

- Iterations continue until converged, defined as $|x^{(n+1)} - x^{(n)}| < \varepsilon_1 + \varepsilon_2 |x^{(n+1)}|$
- Coding this problem uses variables xNew and xOld for $x^{(n+1)}$ and $x^{(n)}$ and e1 and e2 for error tolerances ε_1 and ε_2

Exercise Continued

$$x^{(n+1)} = \frac{x^{(n)}}{2} + \frac{A}{2x^{(n)}} \quad until \left|x^{(n+1)} - x^{(n+1)}\right| < \varepsilon_1 + \varepsilon_2 \left|x^{(n+1)}\right|$$

- Define a bool variable, converged, that is true when $|x^{(n+1)} - x^{(n)}| < \varepsilon_1 + \varepsilon_2 |x^{(n+1)}|$ using fabs(x) for |x|
- `bool converged = fabs(xNew - xOld) < e1 + e2 * fabs(xNew);`
- What condition is true the solution is converged or iterations > maximum
- converged || iterations > maximum

Nested If Statements

- Can have one if block inside another
- Example: Find days in month
  - If the number of the month is 4, 6, 9, or 11 the answer is 30
  - If the number of the month is 2
    - If it is a leap year, the answer is 29
    - Otherwise the answer is 28
  - For all other month numbers (1, 3, 5, 7, 8, 10, and 12) the answer is 31

Days in Month

```
if ( month == 4 || month == 6
  || month == 9 || month == 11 )
{
    days = 30;
}
else if ( month == 2 )
{
    if (leapYear) // bool var
    {
        days = 29;
    } // continue on next chart
```


Days in Month Continued

```
    else
    {
        days = 28;
    }
} // ends else if (month==2)
else
{
    days = 31;
}
```


The `switch` Statement

- An alternative to the if-else-if
  - Tests for equality only
- Dangerous difference – once a case is selected code for that case ***and all subsequent cases*** is executed
- Start with switch (***<expression>***), where ***<expression>*** can be of type char or int
- Followed by "cases" with particular values of the expression
- Following example from Visual C++ Help

Example of switch

```
char c;
switch ( c )
      {
   case 'A':
      capa = capa + 1;
      break;
   case 'a':
      lettera = lettera + 1;
      break;
   default:
      nota = nota + 1;
      }
```

Use break to skip subsequent code sections

Use colon (:) after each case

Braces not required

default case is optional

Operation of switch

- Starts with `switch( <expression> )`
- Have a series of cases, srarting with the statement `case <constant> :`
- Statements in case whose ***<constant>*** matches the value of the ***<expression>*** ***and all subsequent cases*** are executed
  - Use a break statement to do only one case
- Braces not needed for each case, but overall `switch` statement has braces

Exercise with switch Statement

- Printing an error code
  - For code = 0 print “no error”
  - For code = 1 print “value too low”
  - For code = 2 print “value too high”
  - For other codes print “incorrect code”
- Write the program to print the correct message two ways
  - Use an if-else-if
  - Use a `switch` statement

if-else-if Solution

```
if ( code == 0 )
    cout << “No error”;
else if (code == 1 )
    cout << “Value too low”;
else if (code == 2 )
    cout << “Value too high”;
else
    cout << “Incorrect code”;
```


switch Statement Solution

```
switch ( code )  {
    case 0:
        cout << “No error”;
        break;
    case 1:
        cout << “Value too low”;
        break;
// continued on next chart
```


switch Statement Solution II

```
    case 2:
        cout << “Value too high”;
        break;
    default:
        cout << “Incorrect code”;
}
```

Alternative Structures

- Recall previous example of finding days in a month depending on month of year and condition of a leap year or not
- We used a nested if statement to code this
- An alternative is to use a combined condition
- Look at previous code first then consider alternative

Days in Month

```
if ( month == 4 || month == 6
  || month == 9 || month == 11 )
{
    days = 30;
}
else if ( month == 2 )
{
    if (leapYear) // bool var
    {
        days = 29;
    } // continue on next chart
```


Days in Month Continued

```
    else
    {
        days = 28;
    }
} // ends else if (month==2)
else
{
    days = 31;
}
```


Alternative Days in Month

```
if ( month == 4 || month ==  6 ||
     month == 9 || month == 11 )
    days = 30;
else if ( month == 2 && leapYear )
    days = 29;
else if ( month == 2 )
    days = 28;
else
    days = 31;
```

At this else-if we know that if month equals two it is not a leap year. Why?

The Dean's List

- Under a proposed change a student makes the dean's list if the student is a undergraduate and
  - The student completes 12 units in a with a grade point average of 3.5 ***or***
  - The student completes at least 6 units with a grade point average of 3.7
- Select variables and write code that sets the bool variable deansList to true if the student makes the list (false if not)

The Dean's List Code

- Use the following variables
  - status – a string variable equal to "grad" for graduate students and "undgrd" for undergraduaute students
  - units – a double variable equal to the units taken
  - gpa – a double variable equal to the grade point average
  - deansList – a bool variable that is true or false is a student is or is not on the list

Dean's List Problem Summary

- Set bool variable **`deansList`** to true if either condition below holds (false if not)
  - The student completes 12 units in a with a grade point average of 3.5 ***or***
  - The student completes at least 6 units with a grade point average of 3.7
- Variables to be used
  - type double **`units`** for units
  - type double **`gpa`** for grade point average
  - type string for **`status`** = "grad" or "ugrd"

The Dean's List Code II

```
if (status == "undgrd" ) {
    if (units >= 12 && gpa >= 3.5 )
        deansList = true;
     else if ( units >= 6 &&
              gpa >= 3.7 )
        deansList = true;
     else
        deansList = false;
}
else
    deansList = false;
```


Deans List Code III

```
deansList = status == "undgrd"
  && (
     ( units >= 12 && gpa >= 3.5 )
  || ( units >=  6 && gpa >= 3.7 )
     );
```