Program Control Structures

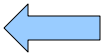
- C++ Control Structures
- Logical Operators
- Selection
- Repetition

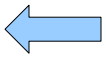
Essential Structures

- Simple Statements
  - e.g. cout << “Name: “ << Student;
- Compound Statements
  - {statement statement ...}
- Selection Statements
  - If, if else, switch
- Repetition Statements
  - while, for, do while

Simple Statements: Executed in Order

```
#include <iostream>
using namespace std;
int main(){
  char input;
 short val;
 cout << "Enter a keyboard character" << endl;
  cin >> input;
  val=input;
  cout << "The numerical value for " << input << " is " << val <<
endl; // A statement can extend to more than one line
  return 0;
}
```





Compound Statements

- A group of statements that can substitute for a simpe statement (see if, if-else, etc)
  - {

```
 cout << "Enter a keyboard character" << endl;
  cin >> input;
  val=input;
  cout << "The numerical value for " << input << " is " << val << endl;
// A statement can extend to more than one line
  return 0;
}
```

If Statement

- If (condition) statement
  - If the condition is true execute the statement; otherwise skip (do not execute) the statement.
  - If (value > 5) value=value-5;
  - If (error) {
    cout << “A major error has occurred.”;
    cout << “ Exiting Program” << endl;
    return 1;
    }

If-else Statement

- If (condition) statement$_1$
  else statement$_2$;
  - If the condition is true statement$_1$ is executed; otherwise statement$_2$ is executed.
  - ```
    if (coRate <= MAXCO)
            cout << "meets ";
        else
            cout << "exceeds ";
    ```
    ```

If-else with compound statements

- Determine if a number is between -7 and -3 or if it is between 1 and 5
- If number is negative
  - Check if it is between -7 and -3
    - If number is < -7 complain and quit
    - else
      - If number is <=-3 give thanks
      - else complain and quit
  - Otherwise check if it is between 1 and 5
    - If number is > 5 complain and quit
    - Else
      - If number is >=1 give thanks
      - else complain and quit

If-else with compound statements

- Determine if a number is between -7 and -3 or if it is between 1 and 5
- If number is negative
  - //Check if it is between -7 and -3
    - If (number is < -7) {complain and quit}
    - else
      - If (number is <=-3) give thanks
      - else { complain and quit}
- else //check if it is between 1 and 5
  - If (number is > 5) {complain and quit}
  - else
    - If (number is >= 1) give thanks
    - else {complain and quit}

If-else with compound statements

- Determine if a number is between -7 and -3 or if it is between 1 and 5
- If (number < 0){ //verify number between -7 and -3
  }
  else{ //the number is greater than 0, verify it is between 1 and 5
  }

If-else with compound statements

- Determine if a number is between -7 and -3 or if it is between 1 and 5
- ```
  If (number < 0){    //verify number between -7 and -3
      if (number < -7){     //number is at least -7.
                            //need to ckeck if it is less equal to -3

      }
      else{
      }
  }
  else{    //the number is greater than 0, verify it is between 1 and 5
  }
  ```
  ```

If-else with compound statements

- Determine if a number is between -7 and -3 or if it is between 1 and 5
-
```
If (number < 0){  //number <0,  verify number between -7 and -3
    if (number < -7){  //number <-7
        complain;
        exit;
    }
    else{
        if (number <= -3) { //number >=-7, verify <= -3
           give thanks
        }
        else {
          complain and quit
        }
    }
}
else{    //the number is greater than 0, verify it is between 1 and 5
}
```
- Color Annotated program

Covered Later

- Selection Statement switch
- Repetition Statement for
- Repetition Statement do-while

Repetition

- What is the number of dots in a triangle with n levels?

```
o
ooo
ooooo
ooooooo
```

- 1 + 3 + 5 + 7+ ... = (2*1-1) +(2*2-1)+...
- level=1; sum=0;
  - while (level <= n)
    - sum =sum + 2*level-1
    - level=level+1;

More Repetition

- Instead of bailing out when a user enters a incorrect number, repeatedly explain the mistake and ask again for a number
- ```
  bad_no=true;
  while (bad_no){
      prompt for number;
      get number;
      if (number correct)
          bad_no=false;
  }
  ```
  ```

Conditions

- <, <=, >, >=,==,!=
  - 4 == -5 - -9 is true
  - 4 != 2 is true
  - 'a' < 'b' is true (the ASCII code is arranged so that letter comparisons conform to alphabetical order)
  - '0' > '1' is false. (the ASCII code is arranged so that digit comparisons conform to numerical order)

Logical Operators

- &&, ||, ! ⟺ and, or, not
  - o

A	B	A && B	A || B	!A	!B
true	true	true	true	false	false
true	false	false	true	false	true
false	true	false	true	true	false
false	false	false	false	true	true

Using Logical Operation

- A number **num** is between -7 and -3
  - (**num** >= -7) && (**num** <= -3)
- A number **num** is between 1 and 5
  - (**num** >= 1) && (**num** <= 5)
- A number **num** is eithter between -7 and -3 or between 1 and 5
  - ((**num** >= -7) && (**num** <= -3)) || ((**num** >= 1) && (**num** <= 5))
- Better version of previous program.

Logical Operator Precedence

- There is an established precedence for logical operators. But using parentheses is saver.

TABLE 3.4 Precedence of Operations

Operation	Precedence
function calls explicit type conversions	highest (evaluated first)
! unary + unary –	
binary * / %	
binary + binary –	
< > <= >=	
== !=	
&&	
||	
=	lowest (evaluated last)

Exercises

- Section 3.2, #1
- Section 3.4, #1

Homework (due 9/28 before class)

- Programming Project, pg 105, #1
- Alter the program ifelse3.cpp so that it repeats the request to enter a number until the user gets it correct.
- Hand in: email to ruttan@cs.kent.edu with the subject:
  10061 Submission Homework 1