CISC/CMPE320

- Notices:
- Assignment 1 due this Friday at 7pm.
- Teamwork: Let me know who the "team leader" is and of any more membership changes. We are still trying to deal with any software and account problems that crop up.

Today

- Back to C++:
  - Boolean Expressions.
  - Conditionals.
  - Loops.
  - Demo of string and vector classes from the STL.

Boolean Expressions

- We have seen the boolean operators already. Here are a few notes:
- Something like

```
a < b < c
```

  will compile and run, but may not produce the desired result. Better to use:

```
a < b && b < c
```

- Remember that **&** and **|** are bitwise operators, not logical ones.

Boolean Expressions, Cont.

- The **&&** and **||** logical operators use "short circuit evaluation":
- For **&&** if the LHS is **`false`** then the RHS is not evaluated.
- For **||** if the LHS is **`true`** then the RHS is not evaluated.
- (Same as in Java.)

Boolean Expressions, Cont.

- Non zero integers are treated as being **`true`**, and zero is treated as being **`false`**. (*Ouch*!)
- So, you can use logical operators, **&& ||** and **!**, with integers.
- For example, the code:

```
int x = 10;
if (x)
```

- is the same as:

```
int x = 10;
if (x != 0)
```

Boolean Expressions, Cont.

- Also, this is legal syntax:

```
if (x = 10)
```

- The assignment operator returns the value being assigned, which in this case is a `true`! But suppose `x` is `12` and you really meant to type `==`…
- *Ouch, again*!

Boolean Expressions, Cont.

- See TestStuff.cpp.

- Applying **`!`** to a non-zero integer returns **`false`** or zero.
- An **`if`** statement will treat a pointer by itself as an integer – it will be **`true`** unless it is **`nullptr`**.
- You can also test assignment statements since the assignment operator returns the value being assigned.

Conditional Expressions

- C++ has if, if/else and switch.

- Syntax of if/else:

```
if (boolean_expression)
   true_part
else
   false_part
```

- Use **`{ }`** to enclose more than one statement.

`switch` Statement

- Same syntax as in Java:

```
switch (expression) {
   case val1:
      // statements if expression produces val1
      break;
   case val2:
      // statements if expression produces val2
      break;
   case val3:
      …
   default:
      // statements if none of the above is true
} // end switch
```

switch Statement, Cont.

- ***`expression`*** must be an integer.
- The **`break`** statements are needed to prevent the next case statement from taking place if the one above it is **`true`**.
- The **`default`** is only executed if none of the other **`case`** blocks have executed.

The Conditional Operator

```
int max = (n1 > n2) ? n1 : n2;
```

- Is the same as:

```
int max;
if (n1 > n2)
   max = n1
else
   max = n2
```

- Which is easier to read?

Four Loop Structures:

```
while (condition)
   statement_or_block

do
   statement_or_block
while(condition);

for(init; condition; expr)
   statement_or_block

for(type element : collection) // C++11 only
   statement_or_block
```

Goto

- C++ also has the infamous “goto” statement:

```
goto identifier :

….

identifier : statement
```

- Don’t use it!
- (Maybe use it to kick you completely out of a nested loop. **`break`** only kicks you out one level…)

Fall 2017 CISC/CMPE320 - Prof. McLeod 13

`break` and `continue`

- Work as they do in Java:
- **`break;`** spits you right out of the loop.
- **`continue;`** sends you back to the loop condition.

Fall 2017 CISC/CMPE320 - Prof. McLeod 14

string Class

- From the STL.
- Better than C-Strings!
- See StringDemo.cpp
- See CPlusPlus.com for more info and member functions.

Fall 2017 CISC/CMPE320 - Prof. McLeod 15

vector Class

- Demo of the vector (*a template container class*) class from the STL:
- VectorDemo.cpp
- Much better than arrays!
- Again, see CPlusPlus.com for more info and member functions.

Fall 2017 CISC/CMPE320 - Prof. McLeod 16

Other Container Classes

- The STL contains many other built-in container templates:
  - See: http://www.cplusplus.com/reference/stl/
  - array, vector, deque, forward_list, list, stack, queue, priority_queue, set, multiset, map, multimap, unordered_set, unordered_multiset, unordered_map, unordered_multimap, valarray, bitset.

Fall 2017 CISC/CMPE320 - Prof. McLeod 17

The STL in Visual Studio

- Very similar set of template classes.
- See:

https://docs.microsoft.com/en-us/cpp/standard-library/cpp-standard-library-reference

Fall 2017 CISC/CMPE320 - Prof. McLeod 18