Character Comparison

----> sort like from the dictionary "lexicographically" based
collating sequence ----> ASCII table

char a = 'a';
char tab = '\t';
char b = 'b';
// is a == b?
// is a < b?
// is a > b?

String Comparison

// dictionary sorted order ASCII table capitals sort before lowercase letters
string sA = "a"; ----> .at() string sB = "b";
// is sA == sB?
// is sA < sB?
// is sA > sB?

Set of string in sorted order
< to sort this set of strings
""
"A"
"AA"
"Aa"
"Aaa"
"Aaaaa"
"B"
"Bb"
"BbB"
"Bbbbbb"
"C"
"D"
"a"
"b"
"bB"
"bBb"
**Parameter Passing Schemes**

**CALLER**
The pile of code that invokes the function

**CALLEE**
The pile of code that gets called

<table>
<thead>
<tr>
<th>Pass-By-Value</th>
<th>Declared As: void foo(int i); void foo(int);</th>
<th>Called As: foo(12); foo('A'); foo(j); foo(j+1);</th>
<th>&quot;Very Safe&quot; Programming</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Declared As:</strong> void foo(int &amp; a);</td>
<td><strong>Called As:</strong> foo(i); foo(j);</td>
<td><strong>Means:</strong> &quot;unsafe&quot; can cause some &quot;side effects&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Pass-By-Const-Reference

<table>
<thead>
<tr>
<th>Declared As:</th>
<th>Called As:</th>
<th>Means:</th>
</tr>
</thead>
<tbody>
<tr>
<td>void foo( const int &amp; I, int j, int &amp; k );</td>
<td>int i(12); foo(i);</td>
<td>&quot;locked down reference&quot; &quot;strict&quot; What Gets Sent Is A Reference To The Value BUT IT IS LOCKED DOWN The Caller Cannot Change The Value Of The Caller's Variable Because It Is Read-Only It Would Be An Error To Do So... No Copy INBOUND - READONLY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Const Array Parameter</th>
<th>Declared As:</th>
<th>Called As:</th>
<th>Means:</th>
</tr>
</thead>
<tbody>
<tr>
<td>void foo( const int array[ ], int size );</td>
<td>int array[12]; int other[100]; foo( array, 12 ); foo( other, 100 );</td>
<td>The &quot;whole&quot; array gets passed to the function Because the function doesn't know how big it is, we need to send along a companion size parameter to control the loops in the function Kinda Like Pass-By-Const-Reference The Function cannot change</td>
<td></td>
</tr>
<tr>
<td>Array Parameter</td>
<td>Declared As:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>void foo( int array[], int size );</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Called As:</td>
<td>int array[12];</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>int other[100];</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>foo( array, 12 );</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>foo( other, 100 );</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Means:
The "whole" array gets passed to the function
Because the function doesn't know how big it is, we need to send along a companion size parameter to control the loops in the function
Kinda Like Pass-By-Reference
The Function can change any of the elements of the array

---

- Midterm Planned For Monday, July 23rd 10-Noon.
  You Can Bring 1 Page Of Notes 8.5 x 11. Both Sides. Typed Or Scribbled.
  Sample Problems Have Been Released In The CCLE For You To Ponder...
  Out Of 100 Points But Worth 25% Of Your Total Grade
  No Devices, No Phones, Closed Book
  Please Bring A Photo ID
  No Scantron Needed
- Expect To Write Small Amounts Of Code
- Covers:
  Variables and Datatypes
  Boolean Conditions
  `< ` `>` ` ` `<= ` ` `>= ` ` `== ` ` `!= ` ` `&& ` ` `|| ` ` `!
  Flow Of Control Statements
    if  if-else if-else if-else do-while for while switch
  Functions
    pass-by-value
    pass-by-reference
    overloading
  Not Very Much On: std::string
Midterm Review:
Covers up to functions and it does not include arrays
Project 2 kind of exam
C++ is very lowercase oriented
Errors: many thousands
C in the name Compile errors "spelling mistakes"
    typically flag the line number where it thinks the problem was making functions
    said you were make something and the compiler let you call it and you didn't make it....
    LNK2019 ----> linker had trouble creating your executable running the program when you try to rebuild it...
    declared: void today( );
    called it: main { today(); }
    created: void today( int I ) { }
    funny looking no source
    declared: void today( int I, int j, int k, double d=12.4, double e=1.356 );
    created: void today( int I, int j, int k, double d, double e ) {

Error and warning

Linker Error- lethal bad must fix

Syntax Warning - bee sting not good
    bat out all our warnings..

    int I = 1; //; syntax error
    int j = 12;
    int k = 12; /// declare avariable and never use it...

    cout << l << j << endl;

Logic Errors just because your code actually builds and runs shallow compliment testing important
    assert all these different possibilities

Double bugs = 1 / 2 * a + 3 / 4 * b ;
Int seven = 77 / 10;
Int remainderOfSeven = 77 % 10; /// leftover..

    // even: 2 4 6 8 10
    Int value;
    Bool isItEven = ( value % 2 == 0 );
    Bool isItOdd = ( value % 2 == 1 ); /// 1 3 5 7

If ( 2 < x < 3 ) /// compound boolean NOT RIGHT...
{
}

If ( 2 < x  &&  x < 3 ) { .... }
// short-circuit...

X = 5;
If (x < 0 && (x-5) > 10)  /// scary..  Short-circuit
{
  ...
}

If (x < 0 || (x-5) > 10)  /// scary..  Short-circuit
{
  ....
}

bool firstTime = true;
x = 5;
do
{foo( );
x = x + 1;
} while(x < 3 || firstTime)
} /// need to consider  do always runs while might never run....

If (x == 1)
{
double d = 12;
foo( ); bar(); foobar();
} else if (x == 2)
{
bar( ); foobar( );
}
Elseif( x == i )
{
wontbeableconvert();
}
else
do always runs while
{foo();
  break;
}

switch( x )
{
case 1:
double d = 12; /// cannot do this...
foo();
break;
case 2:
bar();
case i:

/// project 4
```c++
Int locateMinimum( const string array[], int n )
{
    for( int I = 0, j = 12, k = 13; I < n && I > 1 && j < 12; i++, j--, ++k)
    {
        if ( I == 12)
            break;   /// panic and flee
        if (k == 12)
            continue;   /// skips all the code below….   Loops back to the boolean ?

        /// goal : #1  "Make Progress"  toward eventually turning that boolean to false
        I = i+1;
    }
}

if (a < b)
    a = 12;
    b = 13;
    c = 14;
    cout << "a=" << a << " b=" << b << " c=" << c << endl;

Variables
    specific rule        [_A-Za-z][0-9_A-Za-z]*    good:     h123    bad 123h    *12%^%$
    lifecycle flow: declare initialize use it dies off
    has a type ------> C++ information space is required    int I; on the stack
    C++ is pretty mean: don't initialize it, garbagey value into all your variables
    int I = 12;
    int j( 5 );
    { scope....    Declared inside a scope
declar variables even outside a scope altogether

        /// bad
        int bad = 12;

        int main( )
        {
            int I = 12;
            int k = 12;
        }
```
Math Symbols:
+ - * /
+= -= *= /=
++ --
/ %

double d = 13 / 5; // place we’re putting has no effect....
double floatanswer = 13 / 5.0;
double anotherway = static_cast< double >( 13 ) / 5;
double alternate = double ( 13 ) / 5; /// C way of doing it.... Less safe...
int I = 13 / 5; // 2 no errors no warning
int remainder = 13 % 5; /// 3

int j += k++ - m; /// l-values (variable) and r-values (variable, expression, constant)

m = m - 1; /// order listed
j = j + (k - m);
k = k + 1;

int j += ++k - --m;

k = k + 1;
m = m - 1;
j = j + (k - m);

for (I = 0; I < 10; i++) {
}

for (I = 0; I < 10; ++i) /// no noticeable difference
{
}

Unravel all the contractions
++ + 1
-- - 1
+= _____ = _____ +

One half of a: 1/2 a 1 / 2 * a 1.0 / 2.0 * a

Number line: <--------|---------------|--------------->

2 3
\begin{align*}
(2 < x < 3) & \quad \& \quad \| \quad (2 < x) \& (x < 3) \\
(x == 4, 5) & \quad (x == 4) \| (x == 5) \\
0.5 * a + .75 * b
\end{align*}

\begin{verbatim}
string s = "assafdsdfsaddf";
string t = "";
char c = '4';
char bad = 'sdfddf';
char escape = '\n'; //\ endl
char tab = '\t'; //\ tab
\end{verbatim}

Process text in C++

If we use >> to read in a string get the first word terminates on whitespace
>> consecutively leaves behind the whitespace
defaults work perfect consecutively >> to read the data....
Getline whenever we process text
breaks on newline
Cin.ignore( 10000, '\n' ); // pushes past the newline
Mixing >> and calls to getline

Flow of control choices
convert from one to another

\begin{verbatim}
If (a == 12) --------> switch(a) {
    case 12:       // case fixed CONSTANT
        b = 100;
        case 11:
        case 10:
        Else if (a == 100)
            b = 100;
            // break; bugs
        Else if (a == 50)
            b = a;
            break;
        else
            b = 1;
            case 50:
                b = a;
                break;
            case b+1:
                break;
            default:
                b = 1;
                break;
}
\end{verbatim}

\begin{verbatim}
For (l = 0; l < j; i++) ======> l = 0; while(l < j) ======> l = 0; do
{ foo();
    { if (l < j) {
        foo();
    foo();
\end{verbatim}
While( l < j )
{
    l = l + 1;
}

< > <= >=