Computer Science II — CSci 1200 — Sections 1-4,6
Week 1, Class 2 — August 31, 2001
Functions and Other Aspects of C++
Overview

- C++ functions, including the purpose, declaration, definition, and parameter passing.
- Example of computing grade statistics.
- Other aspects of C++ that students find difficult, including expressions.
Functions — Purpose

- Modularity in writing programs
- Modularity in testing programs
- Re-use of code
A Trivial Example: Max and Min

Here are two simple examples to illustrate a number of aspects of functions:

```c
void Max( int x, int y, int & max_value )
{
    if ( x > y )
        max_value = x;
    else
        max_value = y;
}

int Min( int x, int y )
{
    return x < y ? x : y;
}
```

Here is code that might use these:

```c
int main()
{
    int z;
    Max( 5, 7, z );
    cout << z;
    cout << Min( 10, 4 );
}
```

Discussion points to consider here before looking at a more extended example:
• The structure of the function definition: header and body
• Return types
• Value parameters and reference parameters

We will hand-simulation the action of this program in class.
Example: Computing Grade Statistics

We will look at an extended example that is written to calculate several statistics about a list of grades, including

- minimum
- maximum
- average or mean
- standard deviation

The code is attached to this handout.
Computing Grade Statistics: Code Organization

- Function declarations specify the interface to the function while function definitions give the code of the function.

- The main program passes variables as arguments to the functions. These become function parameters.

- The \texttt{read\_values} function fills in the \texttt{scores} array and the number of scores. It has several error checking steps in its loop to be sure the program behaves reasonably if given bad input.

- The \texttt{square} function is a simple utility.

- The \texttt{compute\_stats} function does most of the work. We will consider it in detail. Note that \texttt{num\_scores} is now \textit{passed by value} and \texttt{scores} is treated as a constant array. The parameters that hold the statistics computed are \textit{passed by reference}. 
Return Types

- Functions that do not return a value have a void return type.

- Functions that do return a value must have the appropriate return type in the declaration and definition.
Parameter Passing in More Detail

- Arguments in the function call (in the main program) and parameters in the function definition are matched-up in order and must have consistent types.

- Use pass-by-value for variables that you don’t need to change.
  - The values in the arguments are copied to the parameter variables.

- Use pass-by-reference for variables that you do need to change.
  - The reference parameter name becomes an alias for the corresponding variable in the function call.
  - In class we will look at the effect of not using pass-by-reference for the statistical values computed.

- Arrays are implicitly passed by reference. (We will see how this works later in the course.)

- When we don’t want the contents of an array to change we use `const` in the function declaration.
Other Review Topics

Here are several other C++ problem areas we will go over in class:

• ‘=’ vs. ‘==’ (single vs. double equals). For example,

    cin >> a;
    if ( a = 5 ) // always true!
    {
        cout << "I flunked because a is " << a << "!\n";
    }

• Logic:
  – Meaning of && and ||
  – Short-circuiting.

• Assignments in the middle of expressions. The following code is legal, but convoluted:

    x = 15.6 + (a=b) * 13;

• Types, values and variables, again!

• Enumeration types make your code clearer. Here’s an example code fragment for storing and manipulations average temperatures:

    enum Month {JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER }; 
    const int MONTHS_IN_YEAR = 12; 
    int avg_temp[ MONTHS_IN_YEAR ];
for ( Month month_index = JANUARY; month_index <= DECEMBER;
    month_index = Month( month_index+1 ) )
    avg_temp[ month_index ] = 0;

avg_temp[ OCTOBER ] = 52;
Reminder

The lecture notes for this week’s classes have only reviewed part of the C++ background material you are expected to know. Use the lecture notes AND the review handout (from Tuesday) AND the textbooks to prepare for the pre-test.